

Article

VHRShips: An Extensive Benchmark Dataset for Scalable Deep Learning-Based Ship Detection Applications

Serdar Kızılkaya¹, Ugur Alganci²  and Elif Sertel^{2,*} 

¹ Satellite Communication and Remote Sensing Program, Informatics Institute, Istanbul Technical University, 34469 Istanbul, Turkey

² Geomatics Engineering Department, Civil Engineering Faculty, Istanbul Technical University, 34469 Istanbul, Turkey

* Correspondence: sertele@itu.edu.tr

Abstract: The classification of maritime boats and ship targets using optical satellite imagery is a challenging subject. This research introduces a unique and rich ship dataset named Very High-Resolution Ships (VHRShips) from Google Earth images, which includes diverse ship types, different ship sizes, several inshore locations, and different data acquisition conditions to improve the scalability of ship detection and mapping applications. In addition, we proposed a deep learning-based multi-stage approach for ship type classification from very high resolution satellite images to evaluate the performance of the VHRShips dataset. Our “Hierarchical Design (HieD)” approach is an end-to-end structure that allows the optimization of the Detection, Localization, Recognition, and Identification (DLRI) stages, independently. We focused on sixteen parent ship classes for the DLR stages, and specifically considered eight child classes of the navy parent class at the identification stage. We used the Xception network in the DRI stages and implemented YOLOv4 for the localization stage. Individual optimization of each stage resulted in F1 scores of 99.17%, 94.20%, 84.08%, and 82.13% for detection, recognition, localization, and identification, respectively. The end-to-end implementation of our proposed approach resulted in F1 scores of 99.17%, 93.43%, 74.00%, and 57.05% for the same order. In comparison, end-to-end YOLOv4 yielded F1-scores of 99.17%, 86.59%, 68.87%, and 56.28% for DLRI, respectively. We achieved higher performance with HieD than YOLOv4 for localization, recognition, and identification stages, indicating the usability of the VHRShips dataset in different detection and classification models. In addition, the proposed method and dataset can be used as a benchmark for further studies to apply deep learning on large-scale geodata to boost GeoAI applications in the maritime domain.



Citation: Kızılkaya, S.; Alganci, U.; Sertel, E. VHRShips: An Extensive Benchmark Dataset for Scalable Deep Learning-Based Ship Detection Applications. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 445. <https://doi.org/10.3390/ijgi11080445>

Academic Editors: Fabio Remondino, Joep Crompvoets and Norbert Haala

Received: 8 June 2022

Accepted: 5 August 2022

Published: 10 August 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: deep learning; optical satellite image; ship classification; end-to-end approach; dataset

1. Introduction

Target classification from optical satellite images is an essential task in remote sensing, which is implemented in various applications to identify geospatial objects such as airplanes, buildings, and ships [1–3]. Optical satellite images have been extensively used in the maritime domain owing to satellites’ capabilities to cover large areas very rapidly, provide synoptic views and high-resolution spatial details, and obtain multi-spectral information. These capabilities are used to produce valuable information for marine applications such as fishery [4], water clarity [5], shoreline extraction [6], and habitat mapping [7]. Optical satellite images are effectively used for patrolling the sea against illegal activities and monitoring maritime traffic [8]. Moreover, their broad coverage in near real-time with high spatial details makes optical satellites feasible for defense applications, such as automated detection of different targets [9,10].

Parallel to the progress in optical satellite technology and image processing algorithms, the outputs of the studies have evolved from detection through further stages [11,12].

Researchers have focused on the methods and algorithm development for monitoring the oceans since the first Landsat satellite launched in 1972. McDonnell and Lewis performed one of the preliminary studies on vessel detection using optical satellite imagery [13]. In early studies, defining a vessel from a satellite image was expressed as a classification. However, the classification of ships requires further stages like Detection, Localization, Recognition, and Identification (DLRI). Each stage has to be studied in detail according to the application requirements.

Previous research focused on different aspects of the ship detection and classification problem. Researchers applied different levels of preprocessing methodologies in previous studies, such as background and foreground segmentation (sea, lake, or river where the ship is located) using the Geographic Information System [14], Digital Elevation data [2], statistical methods [15], and deep learning methods [16]. In addition, there are studies on eliminating effects like clouds and haze [14] or minimizing the impacts of ocean waves [17]. While some studies aimed to detect ships in the open ocean [18] or inshore [2,19–21], some studies went further and classified the ships according to different levels [22,23].

The dataset's quality and extent (volume) are among the most critical factors in determining the success of various studies. Some researchers used open-source datasets (Table 1), and some generated custom datasets [19,22,24] for their implementations. Furthermore, some datasets include satellite images obtained from various sensors to differentiate the objects in different levels of detail. Two previous research studies used Sentinel-2 images with 10 m spatial resolution, which are open-source data but contain some limitations in high spatial detail levels [25,26]. On the other hand, some of the studies used very high resolution satellite images with a spatial resolution of 1 m or better [27,28]. Moreover, the spectral bands used in different applications vary depending on the sensor characteristics. While many studies concentrate on Red-Green-Blue bands, or their pan-sharpened versions, some other studies benefited from NIR [29], SWIR [30], and thermal bands [31].

Different investigations have been conducted on various ship sizes regarding detectable object sizes. According to Li et al., even though some studies define the minimum size of the ship to be detected as between 2 and 150 m, most studies don't specify any limits [10].

Aside from these considerations, the theoretical approaches for DLRI stages have evolved. McDonnell and Lewis employed fundamental statistical image processing methods in one of the first investigations in this area [13]. Today, numerous studies on different levels of the ship DLRI process are based on optical satellite images. In early studies, the outputs were limited to ship detection from the image without localization information due to the low image resolution and image processing power [11]. Parallel to the advances in optical satellite sensors, computational hardware, and software, classification of single ship images, localization of ships from the images, and multi-level classification like recognition and identification of ships have been performed using the threshold-based, salient-based, shape-and-texture-based, statistical, transform domain, anomaly detection, computer vision, or deep learning-based methods [12]. As is also the case in many other engineering disciplines, deep learning-based approaches are being increasingly used in the DLRI process of ships, especially with the success of Convolutional Neural Network (CNN) structures.

Paper Contributions

This study focused on the following requirements regarding dataset and methodology, which were not handled by the previous studies in detail.

- A database with a hierarchical level of classes which makes it suitable for all DLRI stages,
- A database that consists of a large variety of conditions in terms of geography, weather, and spatial arrangement,
- A method that follows all DLRI stages to present a proper remote sensing solution,

- A process that eliminates negative examples (images without ship), which are the largest percentage of images used in the majority of applications, before getting deeper stages,
- A method that forms a suitable base to adopt various deep learning networks and enables end-to-end and repeatable evaluation metrics which exemplify real scenarios.

This research introduces a unique and rich ship dataset named Very High-Resolution Ships (VHRShips). It proposes a unique deep learning-based Hierarchical Design (HieD) approach for multiscale and multipurpose ship detection and mapping purposes. The VHRShips dataset is specifically generated using Google Earth images obtained for different geographical regions and under various atmospheric conditions. Although we selected some images from offshore locations, most of the images in VHRShips contain complex inshore parts. Additionally, the dataset has an extensive range of ship types, including various maritime vessels and navy ship types. VHRShips enables the recognition of sixteen parent classes and a customized identification of navy ship classes for the specified eight categories matching the requirements of various maritime applications on different scales.

The proposed HieD is a multi-stage approach that follows a hierarchy of different stages. The end of each stage is tied to the beginning of the next stage, and all stages have a self-training and self-optimization process. This way, we get the flexibility to use the independent methods for each stage of DLRI, which avoids the loss function optimization for multi-stages. We also compared our proposed HieD approach with the commonly used YOLOv4 model to evaluate the performance of the hierarchical methodology. This research aims to contribute ship detection sub-domain of the maritime re-search field in these aspects: (1) providing a benchmark dataset consisting of large ge-odata, (2) introducing a novel deep learning approach to boost processing and analysis of object detection, and (3) automated extraction of 2D geo-information from large satellite image-based geospatial dataset.

2. Related Works

Recent research examined a variety of deep learning algorithms for different stages of DLRI. Some of these studies are summarized below.

Research by Zhang et al., was one of the initial ship detection studies conducted with deep learning techniques [27]. They built a cascade structure joining the feature extraction methods to CNN. The study claimed 99.1% precision for offshore ships and 95.9% for inshore ships. Tang et al., proposed a coarse ship location method by sea/land segmentation and ship/non-ship classification using deep learning based on wavelet coefficients [32]. They used 2000 SPOT-5 satellite images with 5 m resolution, and defined ships, ocean waves, clouds, coastline, and islands as object classes. The accuracy of the method was 97.58%. This study corresponds to the detection stage of our research. Zou and Shi developed the Singular Value Decomposition Network (SVDNet), which uses CNN structure and a SVD algorithm to detect images from optical satellite images [33]. The results are reasonably good, with a high recall rate. Liu et al., used a CNN-based algorithm to detect and classify ships among ten classes with a dataset generated from 1200 Google Earth images [23]. They achieved 99% detection-localization and 92% classification accuracy. Zhang et al., modified Faster RCNN for ship detection-localization [28]. They reached precision values between 92.95% and 97.64% according to the different confidence and intersection over union (IoU) values. Chen et al., used the Deep Residual Dense Network (DRDN) to detect and classify ships [34]. Their classification stage comprised five different non-ship (wave, island, landing stage, cloud, black background) and five different ships (big, medium, small, multiple, with wake) and achieved a 94.21% F1 score. Shi et al., used the BCCT200 dataset for their ship classification study [35]. Their architecture includes multi-features extraction, CNN, and softmax layers. In the last part of the method, they used the decision level fusion approach to vote on five different feature-based branches. The results of this approach achieved 92.5% average classification accuracy for the four classes of BCCT-200. Zhuang, Li, and Chen worked specifically on inshore ship detection. They worked with the structured sparse representation model (SSRM), Google Earth, HRSC

2016, and DOTA datasets, achieving 87.2% precision and 90.2% recall values [36]. Gao et al., emphasized the importance of negative examples in their work and proposed a method without sea–land segmentation [37]. Feng et al. handled the localization and classification separately [38]. They connected the location stage output to the classification input, the same as HieD. Zhang et al., proposed a multi-level ship classification dataset called FGSC-23 and a deep learning-based ship classification model: attribute-guided multi-level enhanced feature representation network (AMEFRN) [22]. The FGSC-23 dataset consists of 4080 images collected from Google Earth and the Gaofen-1 satellite. However, each image patch includes only one type of ship class, and the dataset is not designed for ship localization. Due to the existence of non-ship images, the dataset can be used for detection. The overall classification accuracy of AMEFRN in FGSC-23 is 93.58% in the grained level (L3). Our HieD approach and VHRShips dataset are feasible for classification, detection, and localization, unlike AMEFRN and FGSC-23, which are suitable for classification and detection. Although there have been many studies on DLRI [11,12,39], there are still plenty of gaps to be filled in developing a practical ship DLRI application.

3. Materials and Methods

3.1. Background for Ship Dataset

The dataset used strongly influences the accuracy and validity of DLRI studies. The number of total samples, the variety of the classes, the balance among the classes, the quality, the originality, the validity, the foreground-background separation format, and the accessibility of the datasets are the key specifications required for these studies [40].

This research is built explicitly for the DLRI process of ships from optical satellite images. The main features of the optical satellite images are the number of spectral bands, the spatial resolution, the incident angle of the sensor, and the radiometric resolution, which are provided with the original satellite image and metadata file. On the other hand, deep learning studies require the collection of a large and diverse dataset from satellite images for target recognition/identification; a process that is time-consuming and expensive. When using archive data, it is not always possible to obtain satellite images from the inshore and offshore regions with enough data to represent the ship types. Beyond being ship-specific, the requirement for an extensive dataset is valid for most geospatial applications where objects need to be automatically extracted from high-resolution and very high-resolution optical satellite images. However, such comprehensive datasets are not generally freely available. Nevertheless, some open-source datasets can be used in DLRI studies with some constraints. Detailed information about the most popular open-source ship datasets is provided in Table 1.

However, to process DLRI in an optical satellite image, the dataset must be suitable and include the required number of images for the training, validation, and testing of all DLRI stages defined in this research. Other requirements of an effective dataset are given below.

- High definition images collected by optical satellites,
- Images with spatial resolution information to ensample a real scenario,
- Image metafile of an existing ship with location and class information,
- Defined recognition and identification levels of each ship in the dataset,
- Images from the inshore and offshore sea,
- Images with rural and urban coasts,
- Images from different locations around the world,
- Some images with clouds and wave clutter to sample a noisy background.

When we focused on the above-indicated requirements, we concluded that the existing datasets were not eligible for this research.

Table 1. Open-source ship datasets for DLRI.

Dataset	Image Source	Application Purpose	Number of Classes /Number of Ship Classes	Description Assessment
VHR-10_dataset_coco [41]	Optical Satellite (Google Earth)	Object classification	10/1	Ships on the images are located yet not classified.
NWPU RESISC45 [42]	Optical Satellite	Scene classification	45/1	Ships on the images are located yet not classified.
DOTA [43]	Optical Satellite	Object classification	15/1	Ships on the images are located yet not classified.
HRSC2016 [23]	Optical Satellite (Google Earth)	Ship detection-localization & classification	19/19	Mostly in-shore images. 19 classes in Level 3. Experimental results only for Level 2 with 4 classes.
Airbus Sandbox Ship Detection ML [44]	Optical Satellite (SPOT)	Ship detection-localization	1/1	Ships on the images are located yet not classified.
xView [45]	Optical Satellite (WorldView-3)	Object classification	60/9	8 parent classes with 60 child classes (maritime vessel parent. class contains 9 different child classes).
HRRSD [46]	Optical Satellite (Google Earth and Baidu Map)	Object classification	13/1	Ships on the images are located yet not classified.
FGSD [47]	Optical Satellite (Google Earth)	Ship detection-localization & classification	44/43	Mostly in-shore images. There are two classification levels. Level 1 consists of submarine, merchant ship, aircraft carrier and warship classes.
BCCT200 [48]	Optical Satellite	Ship classification	4/4	A broad class definition with only gray-scale ship images from the RAPIER system.
MASATI v2 [49]	Optical Satellite (Microsoft Bing Maps)	Ship detection-localization	1/1	Classes: Ship, multiple ships, ship with no coast, ship(s) with coast, sea with no ship, and land with no sea.
FGSC-23 [22]	Optical Satellite (Google Earth and Gaofen-1 Satellite)	Ship classification	23/23	Three classification levels are defined which are ship-non ship (L1), coarse (L2) and grain (L3).

3.2. VHR Ship Dataset

Considering the limitations of existing ship datasets, we generated a new dataset called Very High-Resolution Ships (VHRShips) [50] by using Google Earth images. VHRShips contains 6312 images from 52 locations collected between 26 June 2005 and 1 February 2021.

The VHRShips dataset includes 1000 images without a ship and 5312 images with single or multiple ships.

While forming the VHRShips dataset, the zoom scale of all images was kept near-identical by using Google Earth Professional software (Google Inc., Mountain View, CA, USA, version 7.3.4.8642)'s "eye altitude level adjustment" feature. All images were collected at 500 m eye altitude level and in the high-definition spatial resolution format (720 × 1280 pixels). Through this effort, all images in VHRShips artificially have the same spatial resolution information, corresponding to 43 cm per pixel sampling. Some images with higher or lower spatial resolution were downsampled or upsampled, respectively.

3.3. Class Definitions

The definition of each DLRI stage used in this research is justified by considering the STANAG 3769 "minimum resolved object sizes and scales for imagery interpretation" as a reference [51]. After reinterpreting these definitions according to the purposes of this research and adding a new definition for localization, the DLRI stages can be described as follows:

- Detection: The discovery of the ship's existence (s) in the optical satellite image.
- Localization: Determination of the ship's precise location (s) in the given optical satellite image.
- Recognition: Defining the parent class of each image among the civilian and navy ship groups.
- Identification: Defining the precise child class of each navy ship.

The parent ship classes were initially determined by Lloyd's Register ship classification [52] and Jane's Fighting Ship classification [53]. We re-arranged the classes of VHRShips after analyzing the obtained satellite images and ship representations. According to the class definitions, the VHRShips dataset includes 11,179 ships in 24 parent and 11 child ship categories, which are given in Tables 2 and 3. The parent classes contain one navy and 23 civilian ship class definitions, and stand for the recognition level classification. On the other hand, the child classes detailed 11 navy ship classes of the navy parent class, representing the identification level classification.

VHRShips is a diverse and comprehensive dataset aiming to represent various real-life conditions. It includes complex backgrounds such as urban, civilian, navy shores, bridges, breakwaters, forests, canals, small and big islands/cliffs, and industrial zones. The dataset also comprises images with different ocean spectral properties, an advantage of worldwide image selection strategy. Some images contain atmospheric disturbances such as fog, haze, or clouds. Moreover, VHRShips includes some images of rough sea conditions and some images of ship wakes. These properties make it challenging to obtain successful results for all of the DLRI stages. Appendix A Table A1 demonstrates the geographic locations of the collected images used to create this dataset.

Table 2. VHRShips dataset specifications.

Property	Value/Description
Number of images	6312
Number of images with ship(s)	5312
Number of images without ship	1000
No. of ships	11,179
No. of classes	24 parent, 11 child
Image spectral channel	red-green-blue
Image spatial resolution	1280 × 720
Image source	Google Earth Professional
Bounding box (bBox) type	Horizontal-rectangle
bBox statistics	Length: 7–183–991
(min–mean–max)	Width: 6–47–428
	bBox area: 42–13,640–282,226

Table 3. VHRShips dataset class distribution.

Class Distribution-Number of Ships		
Parent classes (23 civilian classes and 1 navy class) and non-ship images		
Lpg: 33	drill: 33	floatingDock: 51
Ferry: 99	Roro: 102	Offshore: 122
Passenger: 143	Dredging: 194	bargePontoon: 714
dredgerReclamation: 268	Coaster: 344	Undefined: 417
smallPassenger: 419	smallBoat: 891	Tug: 846
Yacht: 1581	Fishing: 37	Container: 580
oilTanker: 594	Tanker: 777	generalCargo: 655
bulkCarrier: 677	oreCarrier: 771	Navy: 831
nonShip: 1000		
Child classes (navy):		
Other: 27	Aircraft: 35	Landing: 34
coastGuard: 40	Submarine: 50	Cruiser: 68
Frigate: 71	patrolForce: 101	Destroyer: 90
serviceCraft: 151	Auxiliary: 164	

After conducting initial experiments, we re-arranged the original dataset to minimize the confusion of some very similar ship classes. After this modification, this research employed the DLRI process with sixteen parent ship classes for the recognition-level classification, and eight child navy classes for the identification-level classification. This specified dataset is generated from the original dataset by performing a re-group process, which is detailed below:

- dredging and bargePontoon classes are grouped into bargePontoon (representing the steady platforms),
- smallPassenger, smallBoat, tug, yacht, and fishing classes are grouped into smallBoat (small size ships),
- oilTanker and tanker classes are grouped into tanker (common tanker group),
- generalCargo, bulkCarrier, and oreCarrier are grouped into generalCargo (cargo ships),
- cruiser, frigate, patrolForce, and destroyer are grouped into destroyer (combat navy ship group).

Some image examples from VHRShips, which are; with/without ships, clear/noisy, simple/complex, with/without land background, with calm/rough sea conditions, and with urban/rural background, are given in Appendix B Figures A1–A6.

3.4. Hardware and Software Configuration

This research does not include any computational performance analysis since it is research-oriented. Therefore, we used currently available mid-range hardware. The information related to hardware, software, and computing environments is provided in Table 4.

Table 4. The hardware and software used in the study.

Hardware Parameters	
Parameter	Value/Type
Processor	Intel Core i7-6700 CPU 3.4 GHz
RAM	32 GB
System	64 bit Windows 10
GPU Name	NVIDIA GeForce GTX 1060 3 GB
GPU Compute Capability	6.1
GPU Total Memory	3.22 GB (2.5 GB available for computing) 1152 CUDA cores 128 bit

Table 4. *Cont.*

Software Parameters	
Software Name	MATLAB 2021a Academic Version Google Colaboratory
Software Libraries	MATLAB Computer Vision Toolbox 10.0 MATLAB GPU Coder Support Package MATLAB Parallel Computing Toolbox 7.4 MATLAB Deep Learning Toolbox 14.2 MATLAB Embedded Coder 7.6 MATLAB Coder 5.2 MATLAB Image Processing Toolbox 11.3 CUDA 10.1 cuDNN 9.0

3.5. Methodology

The proposed HieD approach is a multi-stage learning structure used for the DLRI process of the ships in optical satellite images. The main difference between HieD and the conventional multi-stage structures is its independent but hierarchical stage formation. HieD provides extensive flexibility in terms of plugging any method to each stage. The only link between the stages is the contribution of success and failure of different stages to the next stage. Xception and YOLOv4 were chosen for this study, but the HieD method can easily be used with other deep learning networks. The schematic representation of HieD is given in Figure 1, and brief descriptions of the HieD stages are provided below.

3.5.1. Detection

In almost all optical satellite image-based applications, most of the images showing a total background are labeled as negative examples. Before dealing with the ships in the positive samples, eliminating the negative examples ensures suitable inputs for further stages of DLRI. At this point, HieD's detection stage accomplishes this task by classifying images as positives (with ship(s)) or negatives (without ship), without regard for the location of the ship(s). For this binary classification, well known deep learning classifier networks like Alexnet [54], ResNet 18/50/101 [55], DenseNet 201 [56], GoogLeNet [57], EfficientNet [58], and Xception [59] were implemented, and the highest overall accuracy was achieved with the Xception architecture.

Most images have spatial (x and y) plus channel dimensions, which are handled as a whole by the CNNs, generating a high computational load for the network. Chollet proposes applying spatial convolution to each channel separately [59]. Additionally, 1×1 convolution kernels are used to build the cross-channel correlations to implement the depth-wise separable convolution. Xception is a follow-up of Inception versions 1, 2, and 3 [60]. It performs the convolution kernels first and then applies channel-wise spatial convolution. There is no intermediate ReLU non-linearity layer in the Xception network. The parameters to tune Xception in HieD are given in Table 5. In brief, the binary classification is applied to the input images, and only the images labeled as "including ship(s)" pass through the next stage, localization.

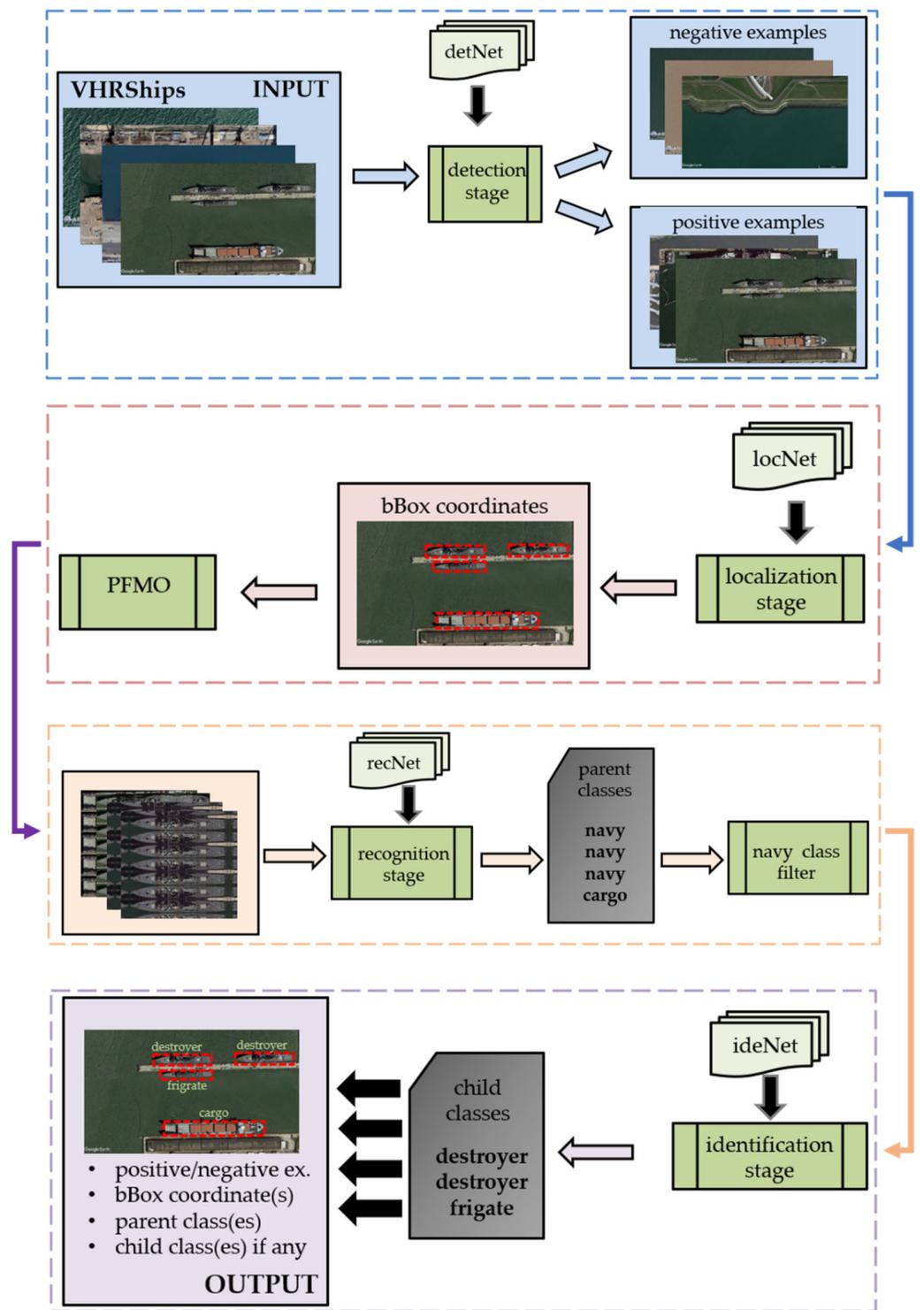


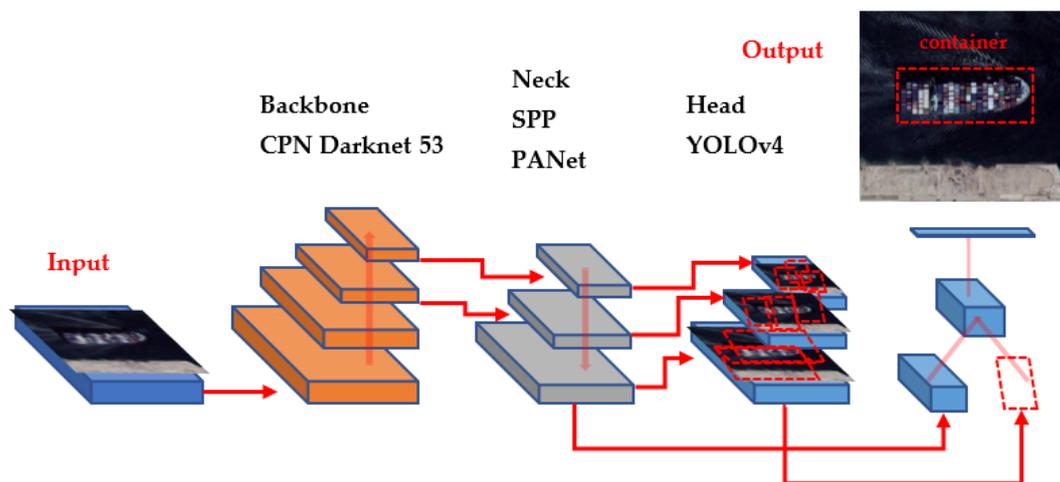
Figure 1. The flowchart of the proposed HieD approach.

Table 5. The parameters used in the detection stage.

Parameter	Value/Type
Network	Xception (MATLAB version 21.1.1) pre-trained with ImageNet database
Network input	416 × 416 × 3 (original images are resized)
Network output	ship/non-ship (binary classification)
Training optimizer	Stochastic Gradient Descent with Momentum (SGDM)
Loss function	Cross-entropy
Data augmentation	Random rotation [0, 360] & random X and Y reflection
Initial learning rate (LR)	0.001
LR drop frequency	Every 10 epochs
LR drop factor	0.1
Number of epoch	50
Mini-batch size	8
Detection threshold	0.2

3.5.2. Localization

Some image patches of VHRShips contain multiple or very small ships that cannot be classified without extracting them from the background. This fact prevents implementing the classification algorithm directly on the image as a whole. The general approach to localizing the objects in the images is to divide the image into sub-images and propose the regions that can contain ship(s). While the sliding window is the most straightforward and least computationally efficient method to divide the images into sub-images, the selective search, the regional proposal network, and the anchor-based search are newer and more efficient approaches. In this research, YOLOv4 is chosen for the localization of ships [61]. Indeed, the YOLO architecture is an end-to-end solution that covers localization, detection, and classification. YOLO optimizes the classes and the bounding boxes (bBox) simultaneously, unlike the proposed HieD. However, in the proposed approach, only the bBox localization feature of YOLOv4 is used (Figure 2).

**Figure 2.** YOLOv4 architecture used for localization.

For the localization stage, YOLOv4 provides a black-box procedure that gets the image labeled with ship existence and exports the bBox of the ship(s) in [x, y, width, length] format.

YOLOv4 is trained and tested with different parameters during the optimization of the localization step. At first, YOLOv4 is trained with one class corresponding to the ship as the foreground and the rest as the background to force the network to focus on only the localization regression. However, the best result is achieved with two classes: the navy and the civilian ships. Additionally, to suppress multiple detections of the same

ship, Non-Maximum Suppression (NMS) is applied [1]. The rest of the parameters to tune YOLOv4 for the localization stage are given in Table 6.

Table 6. The parameters used in the localization stage.

Parameter	Value/Type
Network	YOLO v4 pre-trained with Coco dataset
Backbone	Cross Stage Partial Network Darknet-53
Network input	$608 \times 608 \times 3$ (original images are resized)
Network output	[x, y, width, height] bBox with the score
Loss function	Complete Intersection Over Union (CIoU)
Data augmentation	Cutmix
Initial learning rate (LR)	0.001
LR decay factor	0.0005
Number of epochs	160
Batch size	64
Mini batch size	2
bBox score threshold	0.4
Ground truth overlap threshold	0.05
Non-maximum suppression threshold	0.75

3.5.3. Recognition

The ship bBox with various sizes was obtained as the output of the localization stage. However, the input for the classification nets should have a fixed size. The general approach to overcome this challenge was to resize input images according to the network requirements. The VHRShips dataset includes ship bBox ranging between 42 and 283,336 square pixels in size (Table 2). If the standard image resizing is applied, some ships will be exposed to extreme up-scaling, while others will be exposed to down-scaling. As previously mentioned, this research aims to perform DLRI stages in the optical satellite images, which are scaled. Thus, to avoid losing the scale information via down-scaling or up-scaling, it was decided to keep the extracted ships at their original sizes except for the patches that include ships bigger than the input sizes of the recognition and the identification stages. Firstly, we implemented a zero-padding process around the ships to fit all the extracted images into the input sizes of the classification stages. Afterwards, we determined that the zero-padded pixels, especially in very small objects, should avoid the weights of networks when adequately trained. Finally, the Pattern Forming via Mirror Overlaying (PFMO) approach was proposed to overcome this problem. The PFMO approach follows the steps indicated below.

- Analyzing the length-width ratios of the bBox and rotating it to ensure that the long side is horizontally aligned.
- Putting the rotated bBox in the center and overlaying the rotated bBox around the center bBox until the input size of the stage is filled.
- While overlaying the bBox, the reflection in the X and Y directions is applied to prevent the unintended gradient forming.

The recognition stage uses the Xception network, the same as the detection and identification stages. Even though the same network was used in the detection and the recognition stages, few of the tuning parameters were optimized for the recognition stage, which is briefly explained as:

- The input is gathered through PFMO with $416 \times 416 \times 3$ patch size.
- The output is the probability of 16 parent classes.
- No threshold is applied at the classification level output, the class with the highest score is selected.

Visualization of the PFMO process is provided in Figure 3.

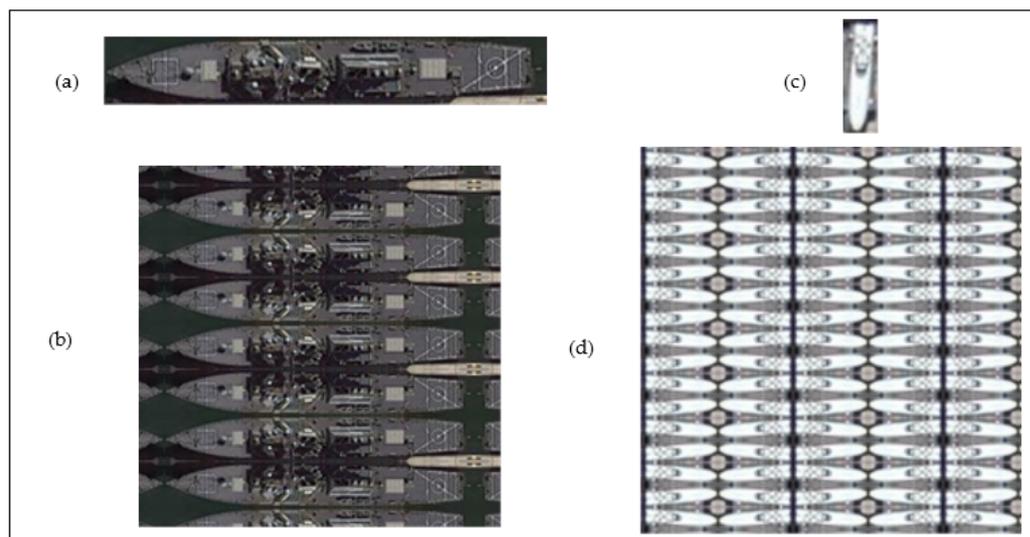


Figure 3. (a) Navy ship, (b) PFMO applied navy ship; (c) Yacht, (d) PFMO applied yacht.

3.5.4. Identification

This stage is an extension of the recognition stage that takes the inputs recognized as “navy ship” and identifies the child classes of “navy ship” as an output. This stage uses the same tuning parameters as the recognition stage for the Xception network training, except for using 8-child classes.

4. Results and Discussion

The results of the study are provided by three approaches. Firstly, the individual performances of the stages are provided. Secondly, the end-to-end evaluation is presented. Lastly, the performance comparison between HieD and one of the state-of-art networks, YOLOv4, is provided. We separated the dataset for training, validation, and testing. While 80% of the VHRShips dataset are kept for the training and validation (64% training and 16% validation), 20% of the dataset is used for testing. In summary, 4016 images are reserved for the training, 1021 images are reserved for validation, and 1275 images are reserved for testing. The images for the training, validation, and testing sets are selected considering the equal class distribution among the sets. There are 200 non-ship images in the testing set, and 1075 images with 2176 marked ships, 142 of which are navy ships.

4.1. Individual Stage Performances

The input of each stage is fed with entirely true samples while calculating the individual stage performances. For instance, only images featuring ship(s) are used in the localization stage. Only the navy class ships retrieved with the ground truth bBox are provided in the identification stage. The individual stage results are summarized in Table 7.

Table 7. The performance of the individual stage (values in percentage).

Stage	Accuracy	Recall	Precision	F1-Score
Detection	98.59	99.81	98.53	99.17
Localization	89.04	91.04	97.60	94.20
Recognition	84.31	72.57 ¹	78.33 ¹	75.34 ¹
		84.30 ²	83.85 ²	84.08 ²
Identification	80.99	83.67 ¹	86.63 ¹	85.12 ¹
		80.98 ²	83.31 ²	82.13 ²

¹ Arithmetic mean value; ² Class sample weighted mean value.

The results showed that the accuracy decreases according to the level of the stage; therefore, details increase. The accuracy of the detection phase is very high despite the complex backgrounds. More importantly, the recall value for the next stage shows promise for avoiding missing any ship(s) images. The confusion matrix of the detection stage is given in Table 8.

Table 8. The confusion matrix of the detection stage (values in percentage).

		Predicted Class	
		Non-Ship	Ship
True Class	Non-ship	92	8
	Ship	0.2	99.8

The localization stage also presents precise outputs, but the relatively lower recall value causes some of the ships to be missed, thus resulting in lower accuracy in the following stages.

The confusion matrix of the recognition stage is given in Table 9. The barge-pontoon, the container, the general cargo, the LPG, the navy, the roro, the small boat, and the tanker classes have close accuracy values to the overall accuracy of the recognition stage. On the other hand, the coaster, the drill, the ferry, the floating dock, the offshore, the passenger, and the “undefined” classes illustrate lower accuracy values than the overall value. When the less accurate classes are analyzed, the following outcomes are deducted:

- The coaster class consists of small water tankers, ore carriers, or cargo ships, which contain small ships that can be confused with small boats, as in this study. In particular, relatively low-resolution images do not include precise details and fail to detect compelling features of these small ships.
- Even though the number of ships in the drill ship class is minimal, the accuracy is close to the overall accuracy. Additionally, the drill class features are very similar to the barge-pontoon class. The drilling instruments in a drill ship are generally located on a barge-pontoon.
- The floating dock class is another class with limited sampling, illustrating a mixture with the barge-pontoon due to the similar shape form and the navy classes since floating docks carry navy ships in the test set.
- The offshore ship class has two characteristics: the navigating bridge on the bow side of the ship and a high bow freeboard. It seems that the number of samples in this class and the level of the differential features wasn't enough to perform over the overall value.
- Some big yachts in the small boat class can be confused with passenger ships. A decline in accuracy regarding the passenger class relative to the overall results from this variety can be noted.
- The “Undefined” class comprises ships with different features from the defined classes or having a pure resolution that makes them impossible to be classified. Most ships in the “undefined” class do not have common features to be learned by the network. Thus, the low accuracy in this class is an expected result.

Table 9. The confusion matrix of the recognition stage (values in percentage).

		Predicted Class														Recall		
		bargePontoon	Coaster	Container	dredgerRec	Drill	Ferry	floatingDock	generalCargo	Lpg	Navy	Offshore	Passenger	Roro	smallBoat		Tanker	Undefined
True Class	bargePontoon	81.7	0.6	1.1		0.6		3.9		5.0		0.6		4.4	1.7	0.6	82	
	Coaster	3.7	31.5				1.9		7.4		1.9			29.6	13.0	11.1	31	
	Container	0.9		78.4					20.7								78	
	dredgerRec	3.7			94.4				1.9								94	
	Drill	16.7				66.7						16.7					67	
	Ferry	11.8					47.1		5.9		5.9			17.6	5.9	5.9	47	
	floatingDock	25						50	8.3		16.7						50	
	generalCargo	1.7	0.7	1.0					90.4		0.5	0.2		1.2	2.4	1.9	90	
	Lpg									100							100	
	Navy	2.1	0.7				1.4	0.7	1.4		79.6			0.7	7.0	2.1	4.2	80
	Offshore	8.0							4.0		4.0	68.0	4.0		4.0	8.0	6.8	68
	Passenger									3.7	3.7		70.4		14.8	7.4	7.0	70
	Roro			5.0					5.0		5.0			85				85
	smallBoat	1.2	1.2						0.4		3.6	0.1	0.3		91.7	0.5	0.9	92
	Tanker	0.8	1.5				0.4		3.4		0.4		0.4		2.3	91.0	0.7	91
	Undefined	15.2	2.5	1.3			1.3		7.6		6.3	3.8	1.3		22.8	2.5	35.4	35
Precision	76	46	92	100	80	62	86	86	88	69	74	76	94	91	89	46		

The confusion matrix of the identification stage is given in Table 10. The aircraft, the destroyer, the landing, the service craft, and the submarine classes have close or higher accuracy values according to the overall accuracy of the identification stage. On the other hand, the auxiliary, the coast guard, and the “other” classes are under the overall value. When the less accurate classes are analyzed, the following outcomes are deduced:

- The auxiliary class is mainly confused with the destroyer class. Both classes have similar sizes, and some auxiliary classes have similar features such as helicopter decks, sensor masts, and a navigational bridge at the bow.
- The same confusion exists between the coast guard and the destroyer classes. The coast guard ships mainly have the same form as the destroyers (especially with the frigates or the patrol ships). They differ in color (white for the coast guard class and navy gray for the destroyer class).
- The “other” class case is the same as the “undefined” class in the recognition stage.

Table 10. The confusion matrix of the identification stage (values in percentage).

		Predicted Class								Recall
		Aircraft	Auxiliary	coastGuard	Destroyer	Landing	Other	serviceCraft	Submarine	
True Class	Aircraft	100								100
	Auxiliary	5	60		30	5				60
	coastGuard			70	30					70
	Destroyer	1.8	3.6	1.8	92.7					93
	Landing					100				100
	Other				33.3		66.7			67
	serviceCraft		10			10		80		80
	Submarine								100	100
Precision	67	89	88	75	75	100	100	100		

4.2. End-to-End Performance

The Hied DLRI for optical satellite image is formed when all stages are attached end to end. The conventional metrics are insufficient for analyzing the multiple stages necessary to evaluate the Hied approach. Therefore, falseNegative2 and falsePositive2

terms are defined, and their meanings are clarified in Table 11. While HieD offers several flexibilities in designing the stages, it introduces the error cycle due to the serial connection. We deduced the following outcomes when we evaluated the output metrics in Table 11.

- The detection stage has the same results as the individual because it is the first stage to stay out of the error cycle. Only two images are missing, and each of them have one ship. The results of the detection phase are very satisfactory, except for the false positives.
- The localization stage is also not affected by the error cycle. Sixteen false true images detected in the previous stage are eliminated in this stage with true negative labeling. Thus, the results are very close to the individual evaluation. The number of missed ships is a bit high, which lowers the recall values of the following stages. On the other hand, a low false positive value increases the precision value of the next stages.
- The recognition starts with 274 errors from the previous two stages. For this reason, the accuracy and recall values are lower than they are in the individual evaluation.
- The identification evaluation metrics are a bit higher than 50 percent. Even though this may seem low, the relatively low number of ships belonging to navy classes, the compulsory samples in the dataset, and the below-average recognition performance (falseNegative and falsePositive2) of the navy class generated these results.

Table 11. The performance of the end-to-end HieD approach.

Metric	Detection	Localization	Recognition	Identification
falseNegative	2 (No. of images labeled as without ship(s) even with ship(s))	226 (No. of ships which are missed)	228 (No. of ships which couldn't be detected & localized, so couldn't be recognized)	29 (No. of ships couldn't be recognized as Navy)
falseNegative2	NA (invalid metric for this stage)	2 (No. of ships couldn't be detected, so couldn't be localized)	NA (invalid metric for this stage)	7 (No. of ships which couldn't be localized, so couldn't be identified)
falsePositive	16 (No. of images labeled as with ship(s) even without ship (s))	46 (bBoxes labeled as ship even not)	405 (No. of ships wrongly recognized)	21 (No. of Navy ships wrongly identified)
falsePositive2	NA (invalid metric for this stage)	NA (invalid metric for this stage)	46 (No. of bBox labeled as ship even not, so incorrectly recognized)	50 (No. of ships recognized as Navy even not, so incorrectly identified)
truePositive	1073 (No. of images with ship(s)s)	1948 (No. of true localized ships)	1543 GTruth (No. of ships truly recognized)	85 (No. of ships truly identified)
trueNegative	184 (No. of images without ship(s))	16 (No. of images labeled as with ship(s) but none of ship localized)	NA (invalid metric for this stage)	NA (invalid metric for this stage)
Accuracy	98.59%	87.76%	69.44%	44.27%
Recall	99.81%	89.52%	70.91%	59.86%
Precision	98.53%	97.69%	77.38%	54.49%
F1-Score	99.17%	93.43%	74.00%	57.05%

4.3. Comparison with the State of Art Network

Beyond the individual or end-to-end performance metrics of the proposed database and method, the comparison with the well-known networks is crucial. At this point, the uniqueness of both VHRShips and HieD doesn't easily enable the adaptation of previous

databases to HieD or methods to VHRShips. The main bottlenecks are the end-to-end evaluation process, four levels for databases, and four stages for methods. On the other hand, to show the position of HieD among the well-known networks, YOLOv4 is adapted to the DLRI process and end-to-end evaluation process. The results of the comparison between HieD and YOLOv4 are given here. YOLOv4 is a one-stage detector network. It processes the image and provides objectiveness with bBox and the class of the target(s). Unlike YOLOv4, HieD presents two levels of classification. For this reason, YOLOv4 is trained with twenty-three classes (fifteen parent classes plus eight child navy classes) to make a complete comparison. The classification results are divided according to the groups used in the recognition and identification stages of HieD. While the objectiveness output of YOLOv4 corresponds to the detection stage, the localization stage is a common part of the two approaches. Fifteen civilian ship classes plus eight navy ship classes on behalf of the general navy class form the recognition stage. The results of eight navy ship classifications are accepted for the identification stage. Considering these pre-assumptions, YOLOv4 was trained with the same parameters, and the results are presented in Table 12.

Table 12. The performance metrics of HieD and YOLOv4.

Stage/Metric	Proposed HieD				YOLO v4			
	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score
Detection	98.59%	99.81%	98.53%	99.17%	98.59%	99.44%	98.89%	99.17%
Localization	87.76%	89.52%	97.69%	93.43%	76.35%	86.41%	86.77%	86.59%
Recognition	69.44%	70.91%	77.38%	74.00%	60.72%	68.72%	69.01%	68.87%
Identification	44.27%	59.86%	54.49%	57.05%	43.05%	45.77%	73.03%	56.28%

A visual representation of the comparison between the proposed HieD and YOLO v4 for inshore regions is provided in Figure 4, and provided in Figure 5 for offshore regions.

The detection performances of HieD and YOLOv4 in VHRShips are almost the same. For localization, HieD performs better than YOLOv4, even though it also uses the bBox localization feature of YOLOv4. The main reason behind this phenomenon is the number of classes used in the YOLOv4, which is twenty-three, while the YOLOv4 used in HieD deals only with two classes. In addition, YOLOv4 tries to tune classification for twenty-three classes while trying to reduce the bBox loss at the same time. This outcome is one of the reasons that HieD is proposed. The recognition performance of YOLOv4 is also less than HieD's. The limited number of samples for navy classes negatively affects the recognition level classification. Lastly, the identification performance of HieD is better except for the precision metric. The reason for that is the high number of false positive-2 in HieD, originating from the recognition stage.

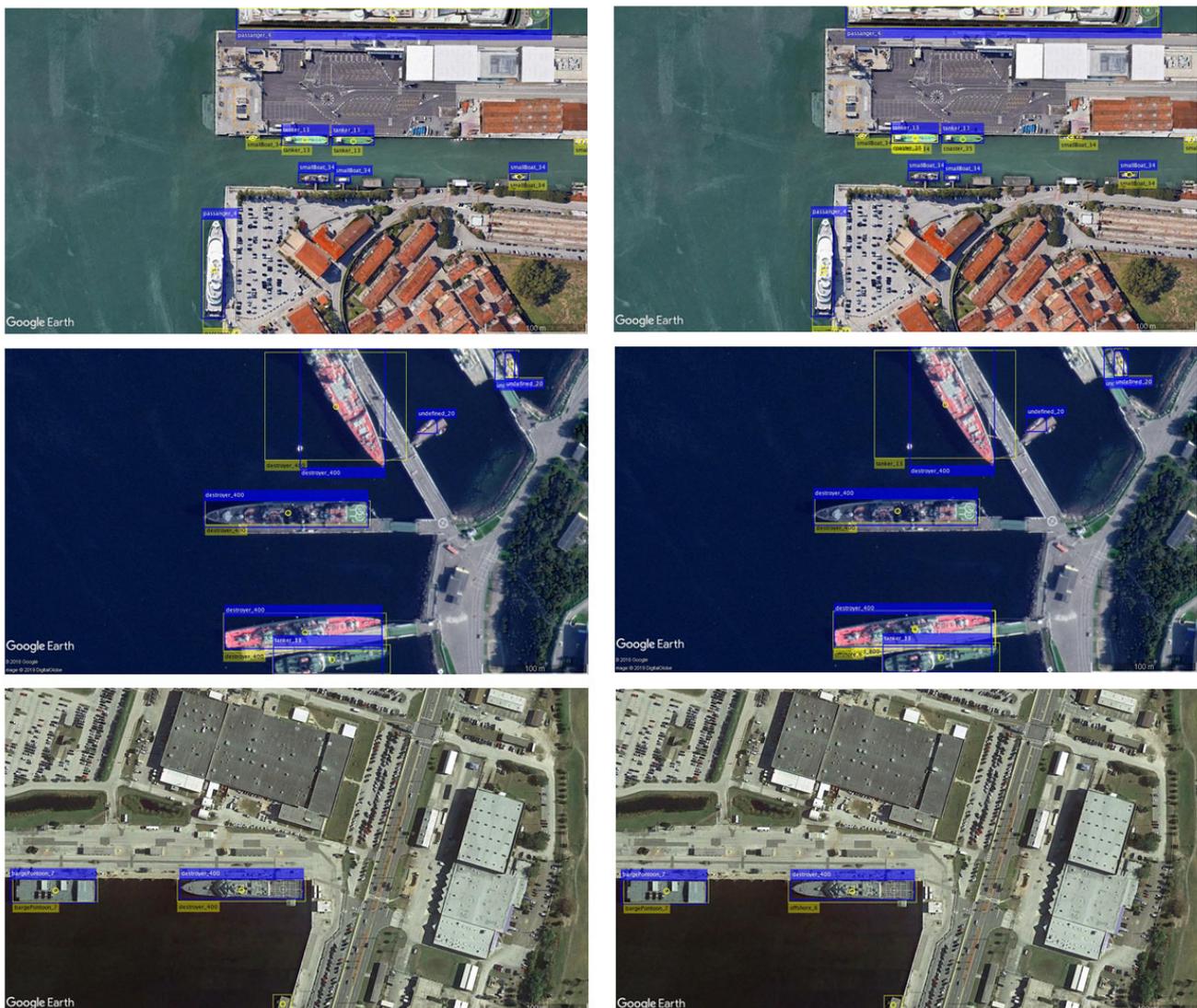


Figure 4. Comparative visuals of HieD (left column) and YOLOv4 (right column) for inshore regions (blue labels are the ground truth and yellow labels with a circle at the center of bBox are the results of HieD or YOLOv4).

4.4. Model Limitations and Future Work

Although the positive contributions of the introduced VHRShips dataset and proposed HieD approach are explained in the previous sections, our approach has some shortcomings that need to be considered. More research can be conducted on the below captions to improve the performance further.

- An additional sub-stage can be designed to detect and eliminate false positive samples for the localization, recognition, and identification stages, if further performance is needed. This design can also reduce the effects of the error cycle created by false positive outcomes.
- The distribution of unbalanced classes can be improved by implementing different over/under-sampling approaches, and the number of samples can be increased for the classes with limited data.
- Introducing more images with different classes will strengthen the dataset for the future.
- Different CNN networks can be integrated to increase the performance of the stages.

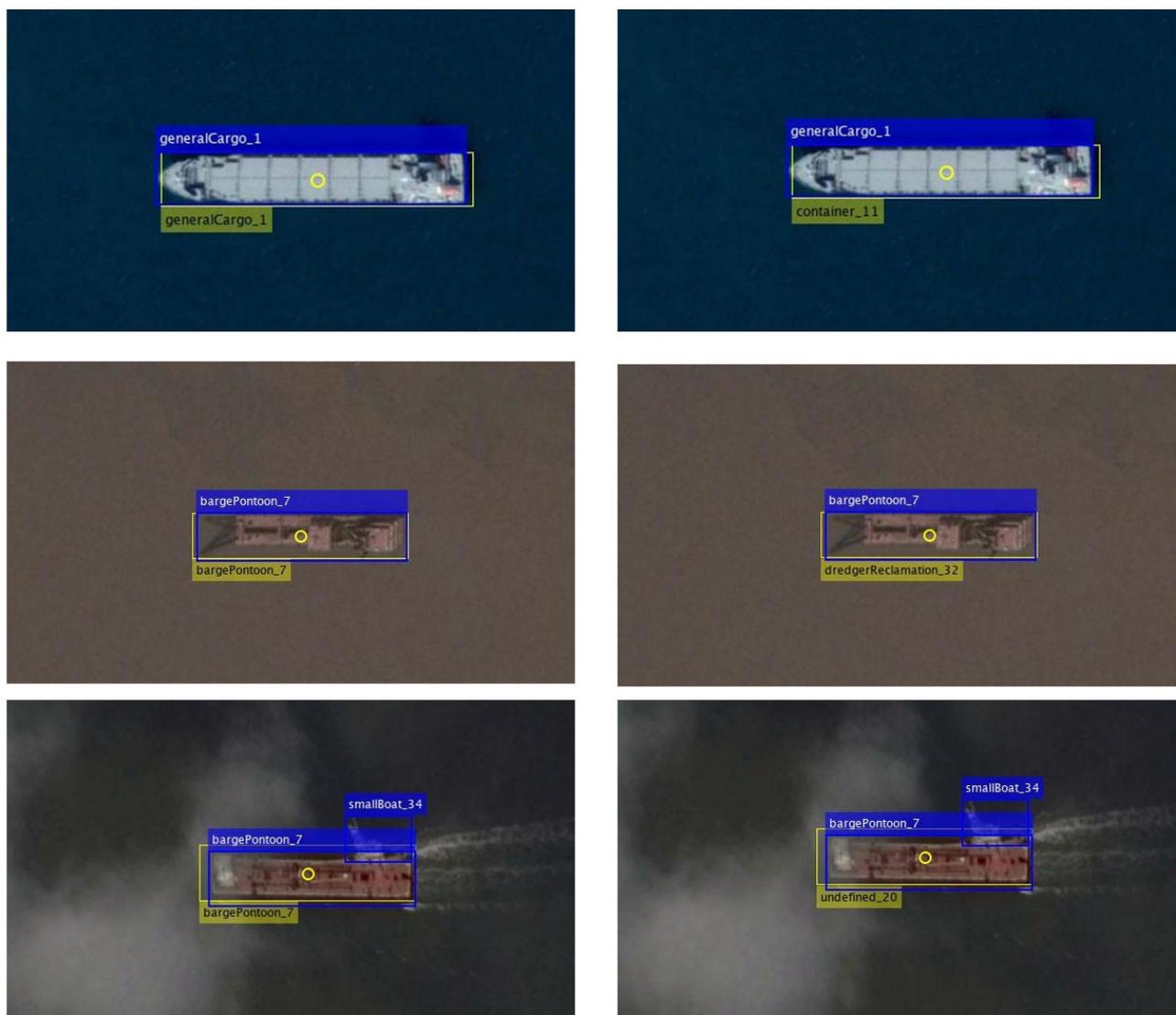


Figure 5. Comparative visuals of HieD (left column) and YOLOv4 (right column) for offshore regions (blue labels are the ground truth and yellow labels with a circle at the center of bBox are the results of HieD or YOLOv4).

5. Conclusions

This research introduces a scalable ship detection dataset—VHRShips—and proposes a deep learning-based HieD approach to perform an end-to-end DLRI process with optical satellite images. HieD is a dynamic and iterative approach formed with the classical CNN and YOLO networks, in which each stage can be independently reformed and tuned to maximize performance. This flexibility is not the only feature that HieD proposes; the PFMO method also helps to boost the performance of the recognition and identification stages. Additionally, the evaluation method used in the study is slightly different from the conventional methods in that we considered the effects of the error cycle and represented them in the outputs. Even though the stages of DLRI are connected with the strict error cycle phenomenon, the performance stays above a satisfactory level. Additionally, a comparison with the one-stage YOLOv4 method provides positive feedback. Another important outcome of this research is the VHRShips dataset, which was prepared from optical satellite images. We generated a complex dataset containing a large variety of ship classes with different ship sizes at various inshore and offshore background distributions to objectively represent real-life conditions. Moreover, images from this dataset reflect

different acquisition conditions, such as viewing geometry and seasonal variations, so that it can be used as a base dataset for further studies related to maritime mapping and ship detection. The outputs of HieD are very promising for further studies and could be used as a benchmark.

In conclusion, the VHRShips dataset and proposed approach could be used for the transfer learning and generalization evaluation of similar tasks. The VHRShips dataset, HieD method, and end-to-end evaluation metrics adequately represent real-world applications.

Author Contributions: Conceptualization, Ugur Alganci and Elif Sertel; methodology, Serdar Kızılkaya, Ugur Alganci and Elif Sertel; validation, Serdar Kızılkaya and Ugur Alganci; formal analysis, Serdar Kızılkaya; writing—original draft preparation, Serdar Kızılkaya, Ugur Alganci and Elif Sertel; supervision, Elif Sertel. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The codes and VHRShips test data to evaluate the HieD approach is available at: <https://github.com/radres333/VHRShips> (accessed on 7 June 2022).

Acknowledgments: We would like to thank anonymous reviewers for their supportive comments to improve our manuscript.

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

Appendix A

Table A1. Geographic locations of images collected to create VHRShips Dataset.

No.	Location	No.	Location
1	Port of Shanghai	22	Port of Hong Kong
2	Port of Tianjin	29	Port of Shenzhen *
3	Port of Guangzhou	30	Suez Canal
4	Port of Ningbo	31	The Panama Canal
5	Port of Rotterdam	32	Naval Station Norfolk
6	Port of Qingdao	33	Naval Station Mayport
7	Port of Dalian	34	Devonport, Plymouth
8	Port of Busan	35	Portsmouth
9	Port of Nassau	36	Port of Canaveral
10	Port of Barcelona	37	Port of Everglades
11	Port of Civitavecchia	38	Port of Cozumel
12	Port of The Balearic Islands	39	Port of New York and New Jersey
13	Port of Southampton	40	Okinawa Naval Base White Beach
14	Snezhnogorsk	41	Yokosuka Naval Base
15	Tartus, Syria	42	Sasebo Naval Base
16	Venice	43	Toulon
17	Taranto Navy Base	44	Sevastopol
18	Navy Augusta	45	San Francisco
19	Port of Los Angeles	46	Dardanelles *
20	Port of Haydarpasa	47	Bosphorus
21	Port of Izmir	48	Port of Mersin
23	England	49	Port of Antalya
24	Jordan Agaba	50	Hanau Germany
25	Kocaeli gulf	51	Port of Mersin
26	Norway	52	Pearl Harbour
27	Qatar	53	Qatar Port of Raundabout
28	San Diego	54	Port of Singapore

* These candidate locations aren't used in VHRShips.

Appendix B

The example images from VHRShips with different conditions are given in the following figures.

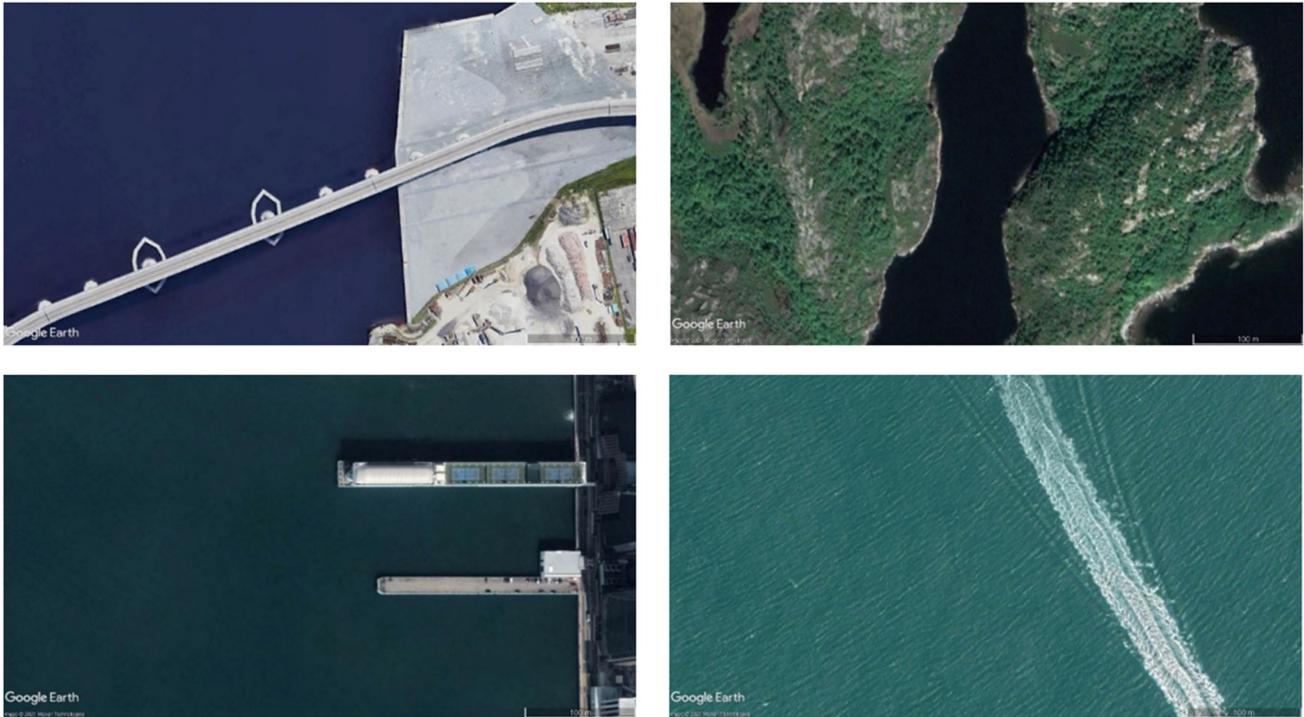


Figure A1. The example images from VHRShips without any ship.

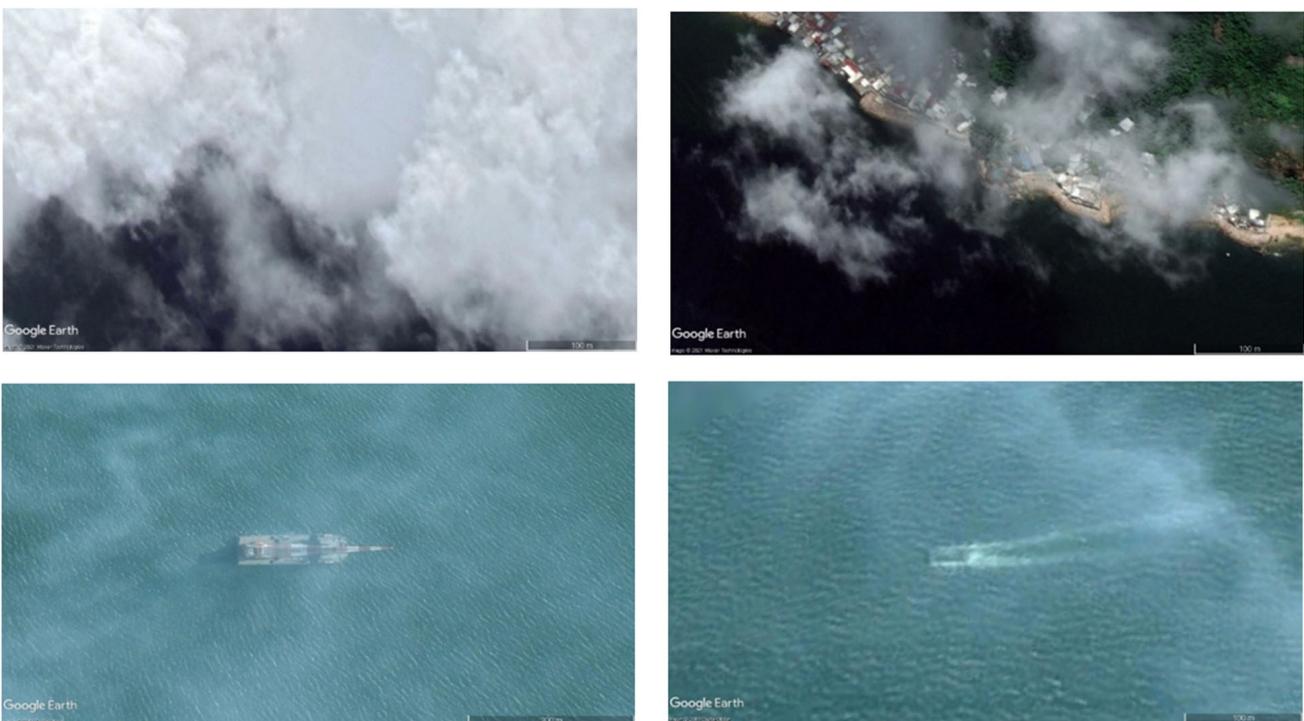


Figure A2. The example images from VHRShips with cloud and haze noise.

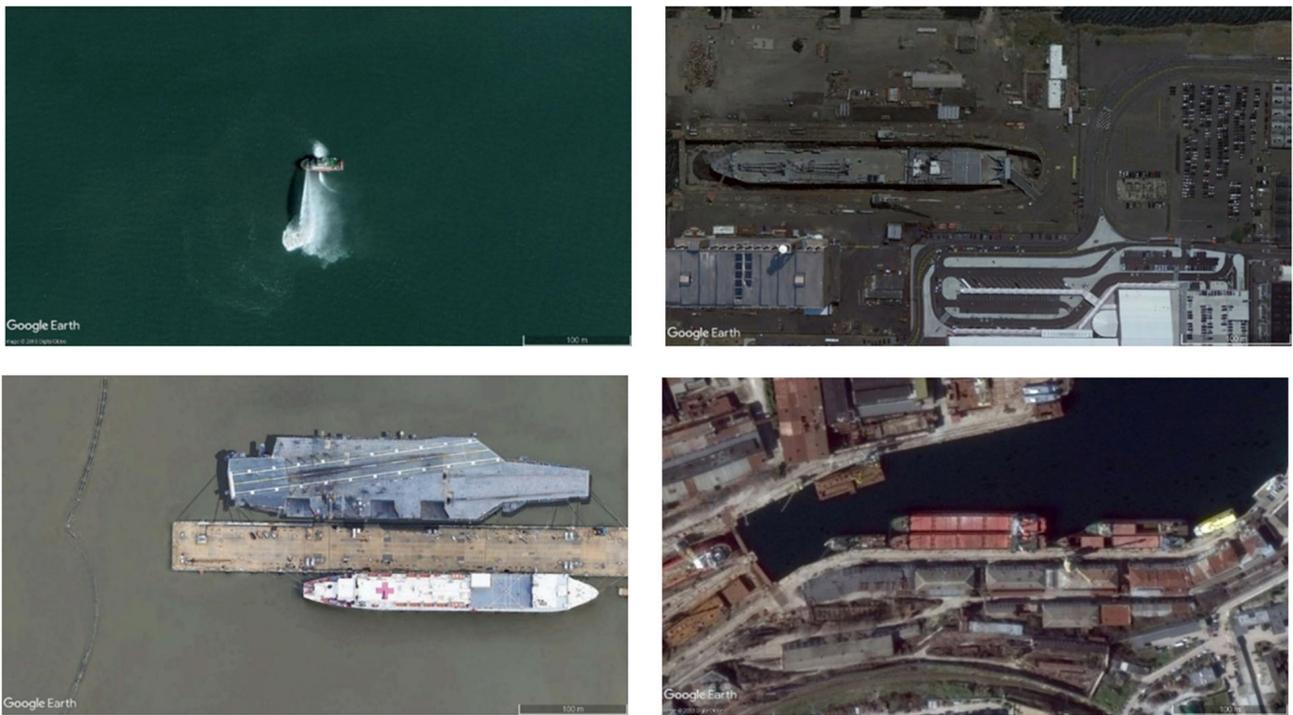


Figure A3. The example images from VHRShips with simple (left column) and complex (right column) background.

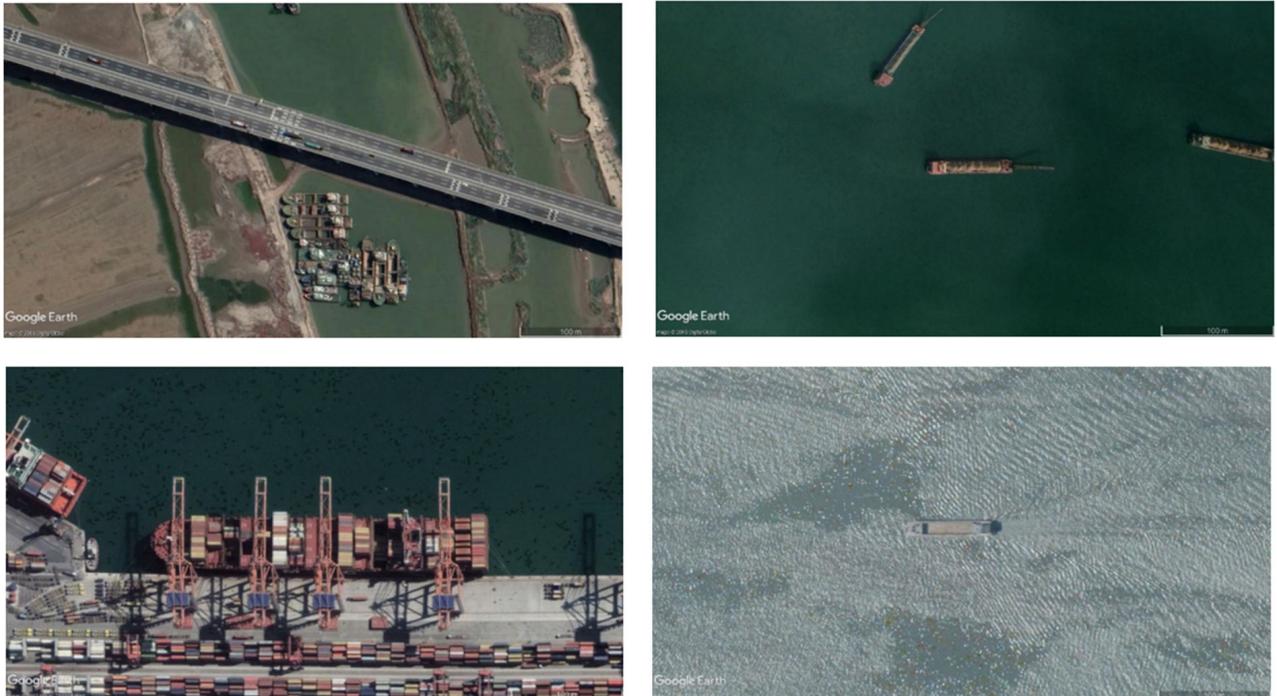


Figure A4. The example images from VHRShips with (left column) and without (right column) land background.

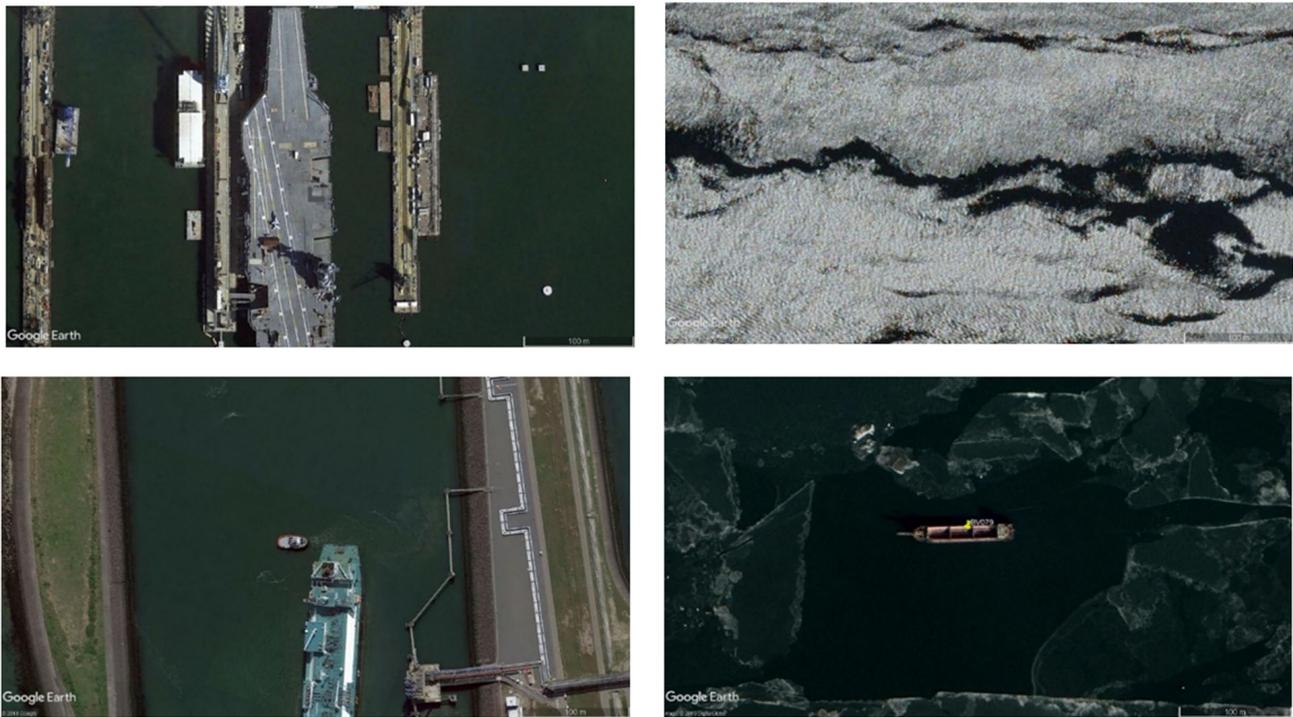


Figure A5. The example images from VHRShips with calm (left column) and rough (right column) sea conditions.



Figure A6. The example images from VHRShips with urban (left column) and rural (right column) background.

References

1. Alganci, U.; Soydas, M.; Sertel, E. Comparative Research on Deep Learning Approaches for Airplane Detection from Very High-Resolution Satellite Images. *Remote Sens.* **2020**, *12*, 458. [[CrossRef](#)]

2. Beşbinar, B.; Alatan, A.A. Inshore ship detection in high-resolution satellite images: Approximation of harbors using sea-land segmentation. In *Image and Signal Processing for Remote Sensing XXI. Int. Soc. Opt. Photonics* **2015**, *9643*, 687–698. [[CrossRef](#)]
3. Bakırman, T.; Komurcu, I.; Sertel, E. Comparative analysis of deep learning based building extraction methods with the new VHR Istanbul dataset. *Expert Syst. Appl.* **2022**, *202*, 117346. [[CrossRef](#)]
4. Kurekin, A.A.; Loveday, B.R.; Clements, O.; Quartly, G.D.; Miller, P.I.; Wiafe, G.; Adu Agyekum, K. Operational monitoring of illegal fishing in Ghana through exploitation of satellite earth observation and AIS data. *Remote Sens.* **2019**, *11*, 293. [[CrossRef](#)]
5. Page, B.P.; Olmanson, L.G.; Mishra, D.R. A harmonized image processing workflow using Sentinel-2/MSI and Landsat-8/OLI for mapping water clarity in optically variable lake systems. *Remote Sens. Environ.* **2019**, *231*, 111284. [[CrossRef](#)]
6. Zollini, S.; Alicandro, M.; Cuevas-González, M.; Baiocchi, V.; Dominici, D.; Buscema, P.M. Shoreline extraction based on an active connection matrix (ACM) image enhancement strategy. *J. Mar. Sci. Eng.* **2019**, *8*, 9. [[CrossRef](#)]
7. Mishra, D.; Narumalani, S.; Rundquist, D.; Lawson, M. Benthic habitat mapping in tropical marine environments using QuickBird multispectral data. *Photogramm. Eng. Remote Sens.* **2006**, *72*, 1037–1048. [[CrossRef](#)]
8. Corbane, C.; Najman, L.; Pecoul, E.; Demagistri, L.; Petit, M. A complete processing chain for ship detection using optical satellite imagery. *Int. J. Remote Sens.* **2010**, *31*, 5837–5854. [[CrossRef](#)]
9. Early, B.R.; Gartzke, E. Spying from space: Reconnaissance satellites and interstate disputes. *J. Confl. Resolut.* **2021**, *65*, 1551–1575. [[CrossRef](#)]
10. Norris, P. Developments in high resolution imaging satellites for the military. *Space Policy* **2011**, *27*, 44–47. [[CrossRef](#)]
11. Li, B.; Xie, X.; Wei, X.; Tang, W. Ship detection and classification from optical remote sensing images: A survey. *Chin. J. Aeronaut.* **2021**, *34*, 145–163. [[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. McDonnell, M.J.; Lewis, A.J. Ship detection from LANDSAT imagery. *Photogramm. Eng. Remote Sens.* **1978**, *44*, 297–301.
14. Jin, T.; Zhang, J. Ship detection from high-resolution imagery based on land masking and cloud filtering. In *Seventh International Conference on Graphic and Image Processing (ICGIP 2015)*; SPIE: Bellingham WA, USA, 2015; Volume 9817, p. 981716. [[CrossRef](#)]
15. You, X.; Li, W. A sea-land segmentation scheme based on statistical model of sea. In *Proceedings of the 2011 4th International Congress on Image and Signal Processing, Shanghai, China, 15–17 October 2011*; Volume 3, pp. 1155–1159. [[CrossRef](#)]
16. Li, R.; Liu, W.; Yang, L.; Sun, S.; Hu, W.; Zhang, F.; Li, W. DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3954–3962. [[CrossRef](#)]
17. Yang, G.; Li, B.; Ji, S.; Gao, F.; Xu, Q. Ship detection from optical satellite images based on sea surface analysis. *IEEE Geosci. Remote Sens.* **2013**, *11*, 641–645. [[CrossRef](#)]
18. Xie, X.; Xu, Q.; Hu, L. Fast ship detection from optical satellite images based on ship distribution probability analysis. In *Proceedings of the 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA), Guangzhou, China, 4–6 July 2016*; pp. 97–101. [[CrossRef](#)]
19. Li, S.; Zhou, Z.; Wang, B.; Wu, F. A novel inshore ship detection via ship head classification and body boundary determination. *IEEE Geosci. Remote Sens.* **2016**, *13*, 1920–1924. [[CrossRef](#)]
20. Li, W.; Fu, K.; Sun, H.; Sun, X.; Guo, Z.; Yan, M.; Zheng, X. Integrated localization and recognition for inshore ships in large scene remote sensing images. *IEEE Geosci. Remote Sens.* **2017**, *14*, 936–940. [[CrossRef](#)]
21. Li, X.; Li, Z.; Lv, S.; Cao, J.; Pan, M.; Ma, Q.; Yu, H. Ship detection of optical remote sensing image in multiple scenes. *Int. J. Remote Sens.* **2021**, 1–29. [[CrossRef](#)]
22. Zhang, X.; Lv, Y.; Yao, L.; Xiong, W.; Fu, C. A New Benchmark and an Attribute-Guided Multilevel Feature Representation Network for Fine-Grained Ship Classification in Optical Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1271–1285. [[CrossRef](#)]
23. Liu, Y.; Cui, H.Y.; Kuang, Z.; Li, G.Q. Ship detection and classification on optical remote sensing images using deep learning. *ITM Web Conf. EDP Sci.* **2017**, *12*, 05012. [[CrossRef](#)]
24. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *Proceedings of the International Conference on Pattern Recognition Applications and Method, Porto, Portugal, 24–26 February 2017*; SciTePress: Setúbal, Portugal, 2017; Volume 2, pp. 324–331. [[CrossRef](#)]
25. Heiselberg, P.; Heiselberg, H. Ship-Iceberg discrimination in Sentinel-2 multispectral imagery by supervised classification. *Remote Sens.* **2017**, *9*, 1156. [[CrossRef](#)]
26. Kanjir, U. Detecting migrant vessels in the Mediterranean Sea: Using Sentinel-2 images to aid humanitarian actions. *Acta Astronaut.* **2019**, *155*, 45–50. [[CrossRef](#)]
27. Zhang, R.; Yao, J.; Zhang, K.; Feng, C.; Zhang, J. S-CNN-based ship detection from high-resolution remote sensing images. *Int. Arch. Photogramm.* **2016**, *41*, 423–430. [[CrossRef](#)]
28. Zhang, S.; Wu, R.; Xu, K.; Wang, J.; Sun, W. R-CNN-based ship detection from high resolution remote sensing imagery. *Remote Sens.* **2019**, *11*, 631. [[CrossRef](#)]
29. Topputo, F.; Massari, M.; Lombardi, R.; Gianinetto, M.; Marchesi, A.; Aiello, M.; Banda, F. Space shepherd: Search and rescue of illegal immigrants in the mediterranean sea through satellite imagery. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015*; pp. 4852–4855. [[CrossRef](#)]

30. Xing, Q.; Meng, R.; Lou, M.; Bing, L.; Liu, X. Remote sensing of ships and offshore oil platforms and mapping the marine oil spill risk source in the Bohai Sea. *Aquat. Pract.* **2015**, *3*, 127–132. [[CrossRef](#)]
31. Liu, Y.; Yao, L.; Xiong, W.; Zhou, Z. Fusion detection of ship targets in low resolution multi-spectral images. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 6545–6548. [[CrossRef](#)]
32. Tang, J.; Deng, C.; Huang, G.B.; Zhao, B. Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1174–1185. [[CrossRef](#)]
33. Zou, Z.; Shi, Z. Ship detection in space borne optical image with SVD networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5832–5845. [[CrossRef](#)]
34. Chen, L.; Shi, W.; Fan, C.; Zou, L.; Deng, D. A novel coarse-to-fine method of ship detection in optical remote sensing images based on a deep residual dense network. *Remote Sens.* **2020**, *12*, 3115. [[CrossRef](#)]
35. Shi, Q.; Li, W.; Tao, R.; Sun, X.; Gao, L. Ship classification based on multi feature ensemble with convolutional neural network. *Remote Sens.* **2019**, *11*, 419. [[CrossRef](#)]
36. Zhuang, Y.; Li, L.; Chen, H. Small sample set inshore ship detection from VHR optical remote sensing images based on structured sparse representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2145–2160. [[CrossRef](#)]
37. Gao, L.; He, Y.; Sun, X.; Jia, X.; Zhang, B. Incorporating Negative Sample Training for Ship Detection Based on Deep Learning. *Sensors* **2019**, *19*, 684. [[CrossRef](#)] [[PubMed](#)]
38. Feng, Y.; Diao, W.; Sun, X.; Yan, M.; Gao, X. Towards Automated Ship Detection and Category Recognition from High-Resolution Aerial Images. *Remote Sens.* **2019**, *11*, 1901. [[CrossRef](#)]
39. Sasikala, J. Ship detection and recognition for offshore and inshore applications: A survey. *Int. J. Intell. Unman. Syst.* **2019**, *7*, 177–188. [[CrossRef](#)]
40. Ekim, B.; Sertel, E. Deep neural network ensembles for remote sensing land cover and land use classification. *Int. J. Digit. Earth* **2021**, *14*, 1868–1881. [[CrossRef](#)]
41. Cheng, G.; Han, J.; Zhou, P.; Guo, L. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote Sens.* **2014**, *98*, 119–132. [[CrossRef](#)]
42. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [[CrossRef](#)]
43. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
44. Kaggle. Airbus Ship Detection Challenge. Available online: <https://www.kaggle.com/c/airbus-ship-detection/data> (accessed on 6 October 2021).
45. Lam, D.; Kuzma, R.; McGee, K.; Dooley, S.; Laielli, M.; Klaric, M.; McCord, B. xvview: Objects in context in overhead imagery. *arXiv* **2018**, arXiv:1802.07856.
46. Zhang, Y.; Yuan, Y.; Feng, Y.; Lu, X. Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5535–5548. [[CrossRef](#)]
47. Chen, K.; Wu, M.; Liu, J.; Zhang, C. FGSD: A dataset for fine-grained ship detection in high resolution satellite images. *arXiv* **2020**, arXiv:2003.06832.
48. Rainey, K.; Parameswaran, S.; Harguess, J.; Stastny, J. Vessel classification in overhead satellite imagery using learned dictionaries. In Applications of Digital Image Processing XXXV. *Int. Soc. Opt. Photonics* **2012**, *8499*, 84992E. [[CrossRef](#)]
49. Gallego, A.J.; Pertusa, A.; Gil, P. Automatic ship classification from optical aerial images with convolutional neural networks. *Remote Sens.* **2018**, *10*, 511. [[CrossRef](#)]
50. Dataset, the Codes and VHRShips Test Data to Evaluate HieD Approach. 2022. Available online: <https://github.com/radres333/VHRShips> (accessed on 7 June 2022).
51. STANAG 3769. In *Minimum Resolved Object Sizes and Scales for Imagery Interpretation*, AIR STD 80/15, 2nd ed.; Air Standards: Washington, DC, USA, 1970; pp. 20330–25058.
52. Register, L. *Rules and Regulations for the Classification of Ships*; Lloyd’s Register: London, UK, 2018.
53. Saunders, S. (Ed.) *Jane’s Fighting Ships, 2003–2004*; Jane’s Information Group: Coulsdon, Surrey, UK; Alexandria, VA, USA, 2002.
54. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
56. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708. [[CrossRef](#)]
57. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
58. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.

-
59. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258. [\[CrossRef\]](#)
 60. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
 61. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.