Spring 2023

Assessing High Dynamic Range Imagery Performance for Object Detection in Maritime Environments

Erasmo Landaeta
Embry-Riddle Aeronautical University, landaete@my.erau.edu

Assessing High Dynamic Range Imagery Performance for Object Detection in Maritime Environments

by

Erasmo Alberto Landaeta Jimenez

A Thesis Submitted to the College of Engineering Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Embry-Riddle Aeronautical University
Daytona Beach, Florida
April 2023

Assessing High Dynamic Range Imagery Performance for Object Detection in Maritime Environments

by

Erasmo Alberto Landaeta Jimenez

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Eric Coyle, Professor of mechanical engineering, Daytona Beach Campus, and Thesis Committee Members Dr. Patrick Currier, Professor and Chair of mechanical engineering, Daytona Beach Campus, and Dr. Jianhua Liu, Associate Professor of electrical and computer engineering, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Thesis Review Committee:

Eric Coyle, Ph.D.
Committee Chair

Patrick Currier, Ph.D.
Committee Member

Jianhua Liu, Ph.D.
Committee Member

Jean-Michel Dhainaut, Ph.D.
Graduate Program Coordinator,
Mechanical Engineering

Patrick Currier, Ph.D.
Department Chair,
Mechanical Engineering

Jim Gregory, Ph.D.
Dean, College of Engineering

Christopher Grant, Ph.D.
Associate Vice President of Academics

Date

Acknowledgements

Here I would like to thank those who made my journey as a graduate student and working on this thesis possible, I feel very fortunate to have the support of this group of people and to have had a chance to interact with them throughout this process. I would like to thank my advisor Dr. Eric Coyle, he has been a mentor through most of my college career and research. Likewise, my committee members Dr. Patrick Currier and Dr. Jianhua Liu for their help in this thesis.

I want to recognize the RobotX team at Embry-Riddle Aeronautical University. With this team I got to apply some of my research techniques and gain confidence in what I was doing. I would like to give special thanks to Dr. Charles Reinholtz, Dutch Holland, Matthew Helms, and David Thompson for introducing me to the team and its work. I would also like to thank Adam Lachguar for his help annotating images.

To my family and my fiancé, I would like to thank you for supporting me all these years and encouraging me in the pursuit of my goals.

To the Department of the Navy, Naval Surface Warfare Center Carderock, and Naval Underwater Warfare Center Keyport, I want to express my gratitude for their support through the Naval Engineering Education Consortium (NEEC). I am grateful for their help and the opportunities they've given me.

My work was supported in part by NEEC grants N00174-19-1-0018 and N00174-22-1-0012, through NSWC Carderock and NUWC Keyport respectively. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research.

Abstract

Researcher: Erasmo A. Landaeta Jimenez

Title: Assessing High Dynamic Range Imagery Performance for Object Detection in Maritime Environments

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Mechanical Engineering

Year: 2023

The field of autonomous robotics has benefited from the implementation of convolutional neural networks in vision-based situational awareness. These strategies help identify surface obstacles and nearby vessels. This study proposes the introduction of high dynamic range cameras on autonomous surface vessels because these cameras capture images at different levels of exposure revealing more detail than fixed exposure cameras. To see if this introduction will be beneficial for autonomous vessels this research will create a dataset of labeled high dynamic range images and single exposure images, then train object detection networks with these datasets to compare the performance of these networks. Faster-RCNN, SSD, and YOLOv5 were used to compare. Results determined Faster-RCNN and YOLOv5 networks trained on fixed exposure images outperformed their HDR counterparts while SSDs performed better when using HDR images. Better fixed exposure network performance is likely attributed to better feature extraction for fixed exposure images. Despite performance metrics, HDR images prove more beneficial in cases of extreme light exposure since features are not lost.

Table of Contents

List of Tables

Page

Table

List of Figures

Page

Figure

Chapter I

Introduction

Carrying over 80% of international trade and 70% of the trade weight of the United States, the surface maritime domain proves to be a crucial component for the global community (statistics, n.d.). In recent years maritime trade has shown its impact on the world economy with the events of the Ever Given being stuck in the Suez Canal, where the accident caused a traffic jam that would cost the world economy $9 billion per day and strained supply chains around the world (refloated, n.d.). In the midst of the war in Ukraine, agreements have been made to allow the navigation of ships for grain exports, with these grains being an important food staple for many countries in the eastern hemisphere and needing the maritime domain for transport (council, n.d.). The movement of goods and food staples in the maritime environment are some of the reasons that make the maritime domain important and why it is crucial for the global community to be aware of events happening in this domain.

The introduction of autonomous vessels in our waterways helps remove human operators from dull and dangerous jobs, making the increasing number of autonomous vessels a welcomed inevitability (Thompson D. J., 2017). Both the military and commercial/research groups are increasingly fielding autonomous vessels in our waterways. Already, there are autonomous surface vessels (ASVs) fulfilling different roles. For example, in Vancouver WA ASVs from David Evans and Associates Marine Services are surveying the Western Galveston Bay for the NOAA (Machines, n.d.). Another example is the Saildrone which has been used by various agencies to collect data in places like the coasts of Hawaii all the way to the Mediterranean Sea. In these missions the Saildrones have mapped the sea floor, collected climate data, and even collected fisheries survey data (drone, n.d.). The United States Navy even employs, Sea Hunters, 132 ft trimaran ASVs capable of autonomous operations in open-ocean environments (recognition, 2016) . With maritime trade growing and data collection being needed by more agencies, we can only expect the presence of ASVs to grow. With more ASVs operating in our waterways the robotics community must ensure these systems operate in a manner that is safe for the systems and those around them.

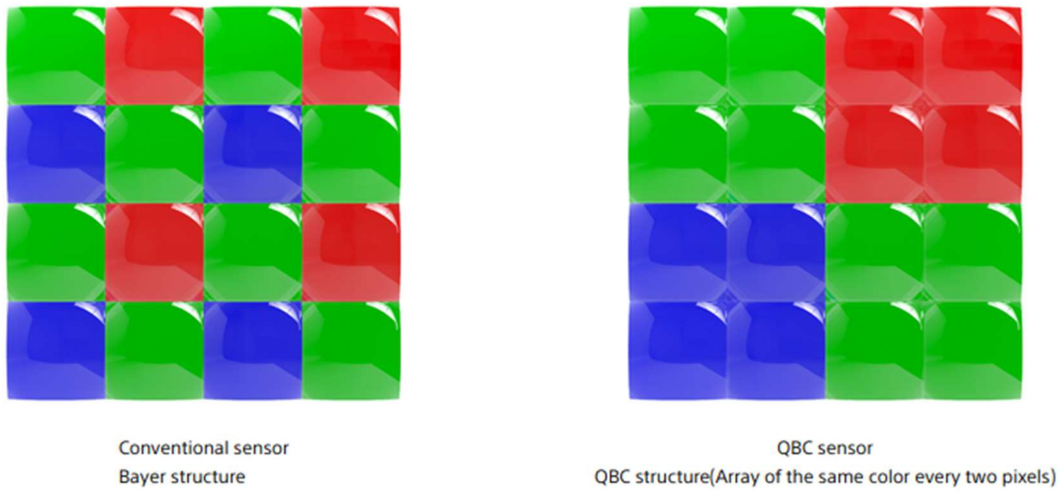
There is always an inherent risk when operating an autonomous system and the risk can be greater in a maritime setting, where surroundings can be unpredictable. The maritime environment is vast, ASVs can find themselves in channels, open water, marinas, littoral zones and more. In all these settings boat traffic, sea states, weather conditions, and wake are all unpredictable factors that can hinder an ASV's operational capabilities. An ASV's best defense against these unpredictable conditions is to be aware of its surroundings, which is the concept of situational awareness. Situational awareness is " a dynamic process of perceiving and comprehending the events in one's environment" leading to predictions of the ways the environment may change and mission performance (Nofi & Analyses, 2000). Situational awareness aids in determining system behavior when avoiding objects and it helps in creating maps used for path planning and visualization (Cadena, et al., 2016). One way researchers approach situational awareness is through sensor fusion. In a study by Haghbayan et al. (2018), the researchers combined radar, LiDAR, thermal and RGB cameras to detect and classify objects with bounding regions in Finnish waterways (Haghbayan, et al., 2018). Likewise, David Thompson (2017) fused GPS/INS, LiDAR, and camera data for detection and classification of maritime objects for the Maritime RobotX Challenge (Thompson D. J., 2017).

However, sensor fusion is not the only situational awareness approach available to researchers, sometimes situational awareness is done through a single type of sensor: like cameras or radar. In a research study by Kuwata et. al (2014), a stereo camera array provided situational awareness by sensing the position and velocity of boats in the nearby vicinity, the camera information was used to help implement the International Regulations for Preventing Collisions at Sea (COLREGS) (Kuwata, Wolf, Zarzhitsky, & Huntsberger, 2014). Another reason ASVs need good situational awareness is the need to follow COLREGS. COLREGS are the rules that help vessels avoid each other, the type of maneuver that must be applied will depend on the location and direction of travel of one vessel relative to another. Using imagery for situational awareness opens the door for many strategies to be used. With imagery, researchers have the option for different tools: computer vision methods, neural networks methods, and camera type selection (stereo, thermal, monocular).

Statement of the problem

This paper focuses on the vision aspect of situational awareness. Computer vision and machine learning process camera data into useful perception information. Computer vision has benefited from the implementation of Convolutional Neural Networks (CNNs), which create the ability to find and classify objects in images. These type of networks were used in (Haghbayan, et al., 2018), (Keunhwan, Kim, & Kim, 2021), (Kowlaski, et al., 2021), (Bovcon, Muhovic, Pers, & Kristan, 2019), (Zhang, Ge, Lin, Zhang, & Sun, 2022). Two widely used methods for vision perception in mobile robotics are object detection networks and semantic segmentation networks. Object detection networks are used to detect and classify objects by taking in an image input and returning regions or bounding boxes with the detected class of the object. On the other hand, semantic segmentation networks provide pixel wise detection and assign a class to each pixel in an image. These computer vision strategies will be further explained in the paper's methods section. Situational awareness approaches using imagery include stereo, monocular, and thermal camera data (Kuwata, Wolf, Zarzhitsky, & Huntsberger, 2014) (Park, Yonghoon, Yoo, & Kim, 2015) (Kowlaski, et al., 2021) ; a sensor net yet seen in the maritime domain is the high dynamic range camera (HDR). HDR cameras are able to process the entire range of visible light conditions from 10^{-4} to 10^{8} cd/m^{2} , resulting in less information loss compared to images taken at a fixed exposure (Mukherjee, Bessa, Melo-Pinto, & Chalmers, 2021). In HDR images areas of the image that "are too dark or light to allow discernment of detail or color, have been removed" (Cox & Booth, 2008). For example, light areas that were washed out in white pixels in a fixed exposure image show more detail and color in the HDR image, similarly areas that are dark due to shadows are no longer just black pixels. HDR cameras take images at different light exposures and 'stitch' them to make the HDR image.

HDR images use different pixel structures to SDR images. HDR images use quad Bayer structures (QBS) which applies the same color filter to adjacent pixel clusters. This method helps prevent information loss and reduce noise (Group, 2023). Figure 1 shows the difference between the SDR and HDR pixel patterns.



Conventional sensor
Bayer structure

QBC sensor
QBC structure(Array of the same color every two pixels)

Figure 1 QBC pixels. Shows the difference between SDR pixels (right) and HDR pixels (left). The left pixel is a representation of the QBC pixel structure, the structure used in HDR images and one of the reasons less data is lost in HDR images **(Group, 2023)**.

Synthetic and native HDR images

HDR images can be synthetic or native. Synthetic HDR images are made by applying augmentations to SDR images and through the application of expansion operators (Mukherjee, Bessa, Melo-Pinto, & Chalmers, 2021). On the other hand, native HDR images come from hardware and cameras. These HDR sensors use image signal processors which apply tone mapping to convert the scene into an 8-bit representation whilst optimizing the HDR range (Thompson D. J., 2023). Synthetic HDR images would be involved in a post analysis process and not in real time as native HDR images.

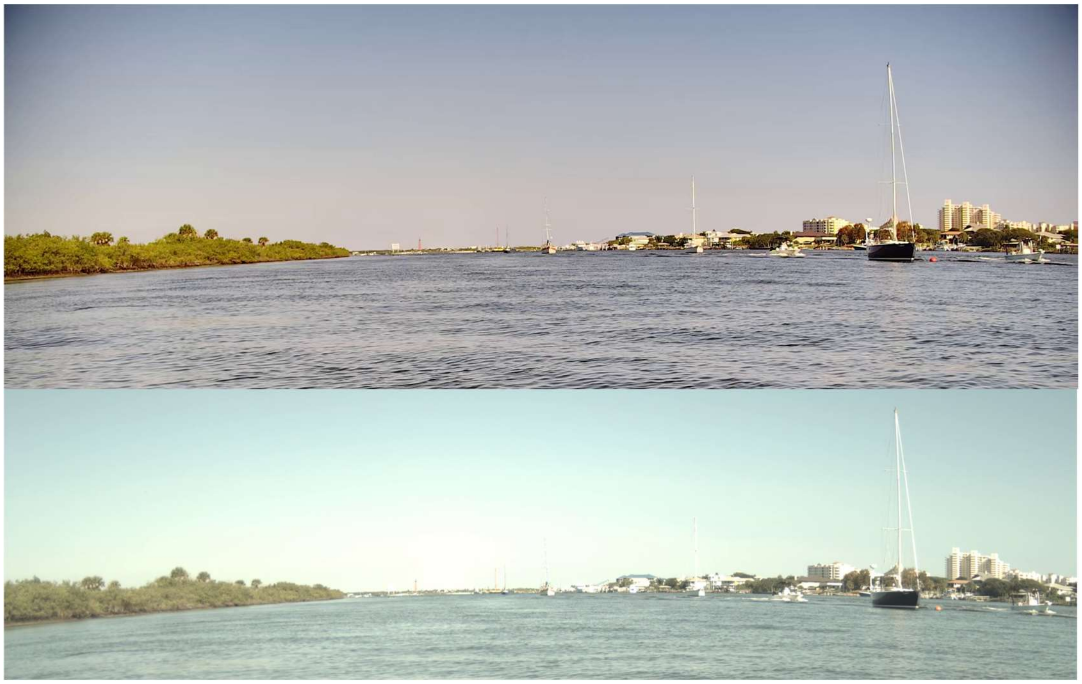


Figure 2 Fixed exposure vs HDR image: This image compares an HDR image (top) and a fixed exposure image (bottom). The images have been cropped to fit on the page.

Figure 2 shows the contrast between the fixed exposure image (top) and the HDR image (bottom). The difference is quite noticeable in fixed exposure image which has saturated white clouds and white pixels seem to blend together in the image, and the channel marker blends in the image. Since HDR images mitigate the effects of shadows and intense lighting by the nature of how they are made, there is a possibility that these images are better suited as inputs for object detection networks than fixed exposure images, in the maritime environment. The retention of detail in the presence of intense lighting or shadows in HDR images might help detection networks extract the features needed for classification. (Mukherjee, Bessa, Melo-Pinto, & Chalmers, 2021) and (Mukherjee, Melo, Filipe, Chalmers, & Bessa, 2020) show there are advantages when training neural networks with HDR images.

Purpose Statement

For the reasons noted above, this paper proposes using HDR images as the inputs of object detection networks in the maritime environment instead of fixed-exposure imagers. Since HDR images can detect a wider range of color and mitigate the loss of information in the presence of too much light or too much shade, it is theorized that HDR cameras may be better suited than fixed exposure imagers for the dynamic settings of the maritime domain. This paper compares object detection networks trained on HDR and SDR images to determine which network is better suited for the maritime environment. Unlike (Mukherjee, Bessa, Melo-Pinto, & Chalmers, 2021) and (Mukherjee, Melo, Filipe, Chalmers, & Bessa, 2020) the HDR images collected from this study will have originated from HDR cameras rather than software-based HDR imagery, and the HDR image footage will have a counterpart of SDR images with almost identical footage to compare detections networks trained on the two sets of images. To do this comparison object detection networks and transfer learning will be used to compare networks on trained on HDR and SDR datasets.

List of Acronyms

ASV	**Autonomous surface vessel**
CNN	**Convolutional neural network**
RGB	**Red, green, blue**
HDR	**High dynamic range**
SDR	**Standard dynamic range**
RCNN	**Regional Convolutional Neural Net**
SSD	**Single Shot Detector**

Chapter II

Review of the Relevant Literature

Scope of paper

The literature review will cover various vision-based situational awareness methods. Strategies covered include sensor fusion, where vision-based strategies are a component of a larger perception strategy, and vision-based only methods, where only camera sensors are used. It is common practice, in vision-based situational awareness, to apply machine learning strategies, computer vision strategies, and CNNs on camera images to extract the location and class of objects. These approaches are common in literature because they can be used in real time. Although these different methods are covered, this paper focuses on the application of vision-based strategies using CNNs; spatial sensors, infrared cameras, clustering, or horizon finding methods are outside the scope of this paper. However, it is important to know these concepts to understand how vision-based situational awareness has reached this point and how it is applied.

The domain where the strategies are being applied is another focus of this paper, the domain in question is the maritime domain. The strategies used in the ground, aerial, or underwater domain are outside of the scope of this paper, but there will be mentions of studies that use HDR imagers in the ground domain because until this point HDR imagers have not been widely used in the maritime domain as a part of a study.

Vision-based and machine-learning maritime perception

Methods that do not use CNNs in maritime perception to detect ships and other objects commonly rely on finding the horizon in images, performing background subtraction, performing edge detection, or use machine learning techniques on the images. In (Bouma, et al., 2008) to detect ships in a harbor the study used edge extraction filters to get boat features and locate the boats in infrared (IR) images. IR cameras were used to reduce noise. Years later, (Park, Yonghoon, Yoo, & Kim, 2015) paired horizon detection strategies with clustering algorithms like DBSAN to detect nearby boats in monocular camera images to apply COLREGS. They used the horizon to help their algorithm determine the distance of objects and clustering methods to distinguish the boat features from the background/ water surface, essentially locating the pixels that belonged to boats. The strategy of finding the horizon line in images still persists today, (Amed Hashamani & Umair, 2022) created a dataset focused on horizon line detection under different weather conditions to evaluate the performance of horizon finding methods. In (Kuwata, Wolf, Zarzhitsky, & Huntsberger, 2014) four stereo cameras were used to determine the velocity and direction of nearby boats to implement COLREGS using velocity objects. The stereo camera strategy consisted of generating range images from the stereo camera pairs, finding the water plane, and applying spatial and temporal filtering to identify and later classify objects like boats and other obstacles (Hunstberger, Aghazarian, Howard, & Trotz, 2011).

In (Prasad, Krishna Prasath, Rajan, Rachmawati, & Rajadbally, 2016) the team evaluates common computer vision techniques in the maritime environment. In the study, horizon detection, registration, background subtraction, and foreground object detection, were applied to the Singapore Maritime Dataset (SMD) to detect ships and other objects. The study found the strategies were not always useful, in detecting objects, due to effects from maritime weather, ship clustering of the horizon, and camera shaking. In the discussion the team urges the need for better vision-based algorithms for the maritime domain.

Convolutional Neural Networks

The introduction of CNNs in the maritime domain simplified the process needed to detect vessels. In the nondeep learning era the detection task requires multiple steps while with neural networks the algorithms and learning are all completed through the neural network and in the time of application the process is an image input then an output result (Zhan, Li, Ji, Li, & Pan, 2021). Improving CNN strategies in this domain is important because "CNN based methods significantly outperform conventional image processing techniques in detecting ship features under inherent noise in marine image data" (Keunhwan, Kim, & Kim, 2021). Object detection networks are common CNNs used in the maritime domain to detect objects. .In object detection networks a 'learner' tries to detect regions of an image containing objects. Objects are determined based on their features. Features are extracted from images using convolutional filters. The features are mathematically related to object types or class through activation functions. Ultimately, object detection networks find objects in images and output the regions where these objects are located, in the image, along with a label for the type of the object found. Another CNN strategy, but outside the scope of this paper, is semantic segmentation. Semantic segmentation networks do pixel wise detection and classification and can be used to find object instances or to parse the background finding the sky, ocean, or land. Segmentation networks are good for detecting areas of similar texture giving them the ability to identify these big background objects.

Since this study focuses on object detection networks it is important to understand some widely used object detection network architectures: Faster-RCNN, SSD,s and YOLO (Ren, Kaiming, Girshick, & Sun, 2017) (Liu, et al., 2016) (Dwyer, 2020).

Faster-RCNN

Faster-RCNN or Faster Regional Convolutional Network is an object detection network from the RCNN family. The RCNN family of detection networks started in 2014 with the introduction of RCNN in (Girshik, Donahue, Darrell, & Malik, 2014). The network consists of three modules, one of the modules uses selective search to propose regions of interest, the second module then uses convolutional neural networks to extract features from these regions, and the last module uses support vector machines to classify the objects inside each of the region proposals. After RCNN came Fast-RCNN in (Girshick, 2015) which increased training and deployment speed over RCNN. Fast-RCNN improved over RCNN by reducing the number of modules that had to be trained and using backpropagation over the network layers. To increase speed Fast-RCNN also applied the convolutional layers once to the entire image, instead applying convolutions to each region proposal, then regions of interests were used on the feature maps which were turned into feature vectors and then fed to classifiers. The improvements from Fast-RCNN made it 9 times faster for training than RCNN (Girshick, 2015).

The Faster-RCNN version combines a region proposal network (RPN) and the ‘Fast RCNN’ detector in a bundle where the two networks share convolutional layers during training. The RPN will find regions of interest in the feature map and then the Fast-RCNN uses those regions to find objects (Ren, Kaiming, Girshick, & Sun, 2017). Faster-RCNN is meant to be able to find objects at different scales thanks to its anchor box methods and their translation-Invariant anchors, meaning the network is able to keep track of the position of different objects throughout convolutions. Anchor boxes are pre-determined and define the scale and aspect ratio of bounding boxes that best match the ground truth boxes in the dataset (Ren, Kaiming, Girshick, & Sun, 2017).

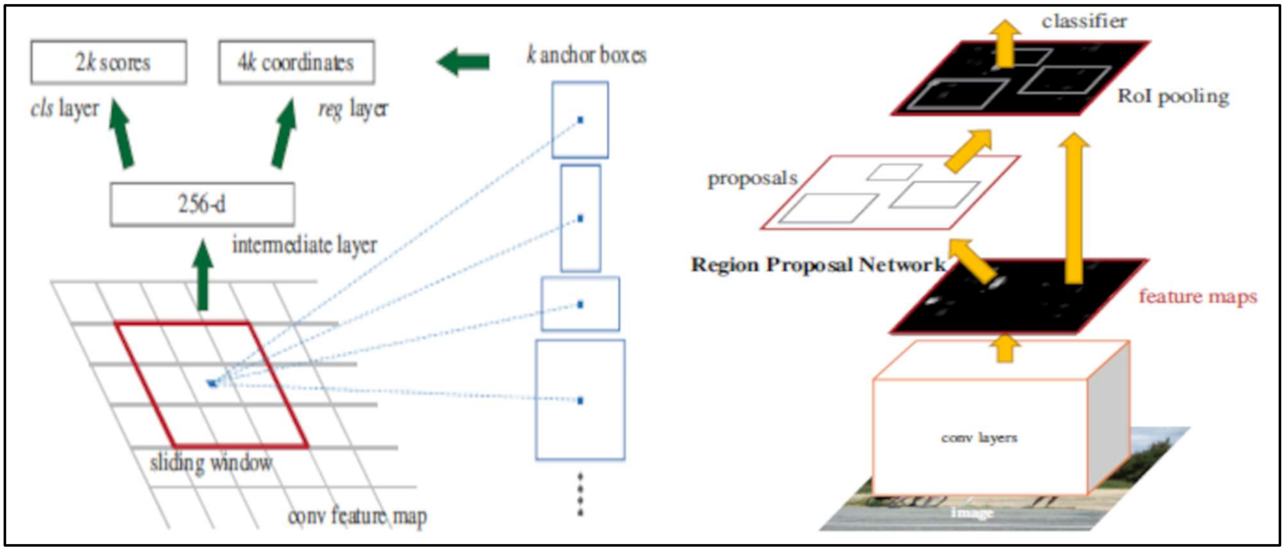


Figure 3 Faster-RCNN modules: the left side of the image shows the RPN anchor boxes, and the classification and regression scores given to each anchor box in the feature map. On the right it's the RCNN module, the figure shows how the RPN shares convolutions with classifier (Ren, Kaiming, Girshick, & Sun, 2017).

Figure 3 shows the usage of anchor boxes in Faster RCNN and how the RPN module and the classifications module work together. Figure 3 is also to show how important the anchor boxes are for proper detection with this network. Although this network can be more accurate than its counterparts it tends to be slower than them.

Single Shot Detectors

Compared to Faster RCNN, SSDs are faster and with comparable detection accuracy based on common object detection datasets, however, SSDs are not as good as RCNN models at detecting small objects (Liu, et al., 2016). SSDs function off a feedforward network method and produce a predetermined number of detection bounding boxes and scores each of the bounding boxes for the likelihood there is an object inside it. The bounding boxes are produced using pre-determined aspect ratios, the pre-determined aspect ratios are placed on feature maps at different scales (this is done so SSDs are able to detect different scaled objects). After the bounding boxes are scored for object confidence a non max suppression step is applied to reduce the number of repeated predictions for a single object.

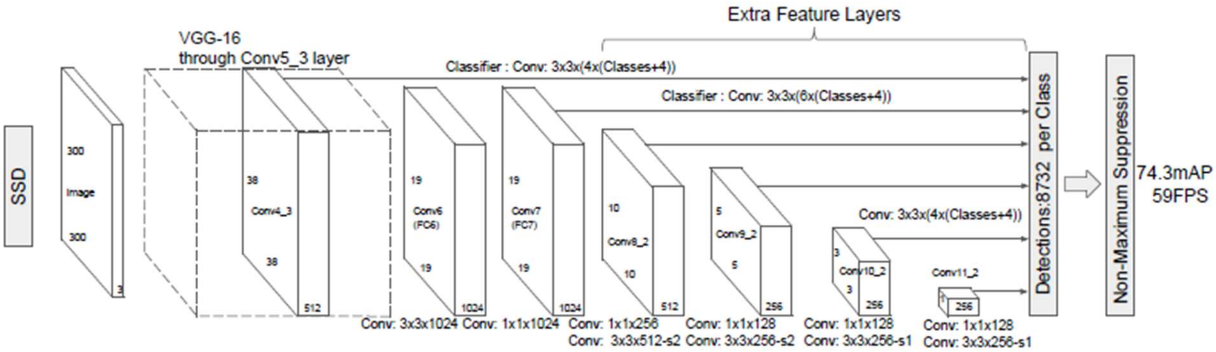


Figure 4 SSD architecture: shows handling of images from input to classification (Liu, et al., 2016)

Figure 4 shows the architecture of an SSD network. The input images go through the VGG16 feature extractor until a certain convolution layer, after that point is reached the convolution layer goes through different convolutions making feature maps at different scales. At the differently scaled feature maps the bounding boxes are applied to detect objects (Liu, et al., 2016).For this study the SSD network was paired with an inceptionV1 backbone. Only hyperparameter values were changed for this network, as discussed in the methods.

You Only Look Once (YOLO)v5

The YOLO object detection models are similar to the SSD models where there is a single module, and the architecture is quick to detect objects. YOLO detection networks started with (Redmon, Divvala, Girshick, & Farhadi, 2015), the goal of the model was to perform detections in a single stage and reduce inference time. The model divided images into grids, calculating the probability of the grid containing an object and then merging these grids. Later versions of YOLO improved the speed and accuracy of the algorithm with YOLOv2 introducing anchor boxes, YOLOv3 introducing feature pyramid networks and YOLOv4 introducing the 'bag of freebies' method to reduce the size of the network and increase speed and accuracy again (Maindola, 2021). The YOLOv5 model used in this study is an improvement on the YOLOv3 version.

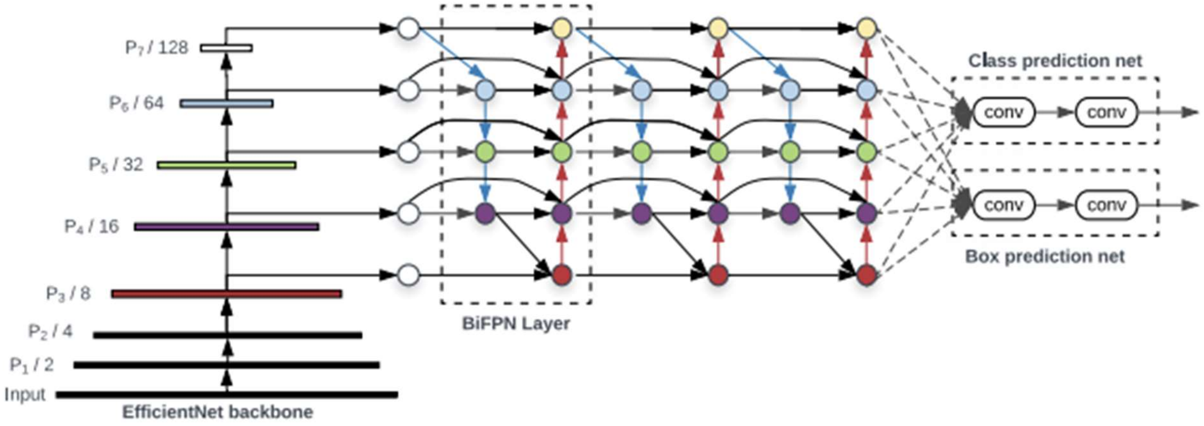


Figure 5 YOLOv5 Architecture. Shows the architecture of YOLOv5 (Dwyer, 2020). The layers of the YOLOv5 network and backbone are shown.

One of the biggest additions from the YOLOv5 model is the use of data augmentation methods like mosaic augmentation. Where 4 source images are combined into one whilst keeping object scale, the advantage is that it acts like a crop and combines classes that may not have been seen together in one scene (Dwyer, 2020). YOLOv5 also has autolearning anchor boxes, the anchor boxes are determined through the distribution of the annotations and k-means and genetic algorithms (Solawetz, 2020).

CNN applications in the maritime domain

This section will outline studies that focus on the application of CNNs in the maritime domain. Most of these studies use RGB images or IR images to sense the environment. In (Bovcon, Muhovic, Pers, & Kristan, 2019) semantic segmentation networks were used to classify the sky, water, and land, in order find navigable paths for ASVs to travel on. Some studies focused on benchmarking neural networks on the few available maritime domain datasets. One of these studies is by (Moosbauer, Konig, Jakel, & Teutsch, 2019), where they tested different versions of Faster-RCNN and Mask-RCNN whilst fine tuning them for the objects in Singapore Maritime Dataset (SMD) (Prasad, D., L, E, & C, 2016) , the best performance for the Faster-RCNN model was an f score of 0.854. In (Betti, Michelozzi, Bracci, & Masini, 2020) a YOLO detection network was trained using RGB images taken from internet search engines to create a dataset of 12 boat classes. In another comparison study by (Soloveiv, et al., 2020), the researchers created their own dataset using RGB and IR cameras for over 13 days in the Finnish archipelago and compared SSD networks and Faster-RCNN networks finding the Faster-RCNN with Resnet101 outperformed the other detection networks on their dataset. In another YOLO application, (Keunhwan, Kim, & Kim, 2021) combine RGB cameras with radar for a maritime situational awareness package. The researchers effectively trained a YOLOv3 network using a dataset composed of their own images along a coastal setting and ship images from PASCAL VOC and the Singapore Maritime Dataset (SMD)

(Prasad, D., L, E, & C, 2016). In (Kowlaski, et al., 2021) IR cameras were used to detect inflatable boats in Polish rivers. The team collected and annotated their own dataset to train YOLOv2, YOLOv3, and Faster-RCNN with different backbones. They found Faster-RCNN outperformed the YOLO networks based on 'classification rate,' the total number of correctly classified objects to the total number of correctly detected objects, a metric akin to precision. The study also showed YOLO networks were significantly faster than the Faster-RCNN networks. In (Zhang, Ge, Lin, Zhang, & Sun, 2022) the team modified a YOLOv4-tiny model to make it better suited for foggy maritime scenarios. The team introduced 3 modifications to the original YOLOv4-tiny architecture. The modifications try to filter out the effects of fog before feature extraction occurs, increase the field of view of the feature maps, and combine information in different dimensions of the channel space. The modified model was then tested on ocean images from COCO and TinyPerson, which were augmented to add the effect of fog. The team found their modified model had a 10-percentage point improvement over the original YOLOv4 network.

Literature review influence

The similar findings by (Soloveiv, et al., 2020) and (Kowlaski, et al., 2021) indicate Faster-RCNN outperforms YOLO and SSD models. These findings influenced the CNN architecture focus of this paper, these studies made Faster-RCNN with a resnet101 backbone the initial focus of the networks. Although the literature does show YOLO networks being used a lot, those networks were not trained using images that were collected solely by researchers, parts of their dataset were made from online images or taken from other datasets. Because the YOLO networks were not being trained with images collected by the researchers the focus of this study was initially centered around Faster-RCNN.

Challenges of object detection in images

To put it briefly, vision based situational awareness in a maritime domain can be considered to be an immature area of cross-disciplinary research with challenges it needs to address (Quiao, Liu, Lv, Li, & Zhang, 2021). One of the challenges that must be faced by the field is the quantity and quality of training data needed for vision-based perception to create robust neural networks. In general, there is a lack of labeled data available, especially compared to datasets available for ground vehicles containing cars and pedestrians. There are only a few maritime datasets that have full scene images and fully annotated images, making it difficult for research (Quiao, Liu, Lv, Li, & Zhang, 2021) (Environments). In this case there are no HDR maritime datasets known to the researcher.

The lack of maritime/ship datasets is not the only dataset challenge, these datasets must also capture different conditions that will be encountered by ships in the maritime environment. In real world application, the ships and their camera sensors will encounter objects under different conditions, these conditions are affected by lighting, scale, background, and viewpoint (Quiao, Liu, Lv, Li, & Zhang, 2021).

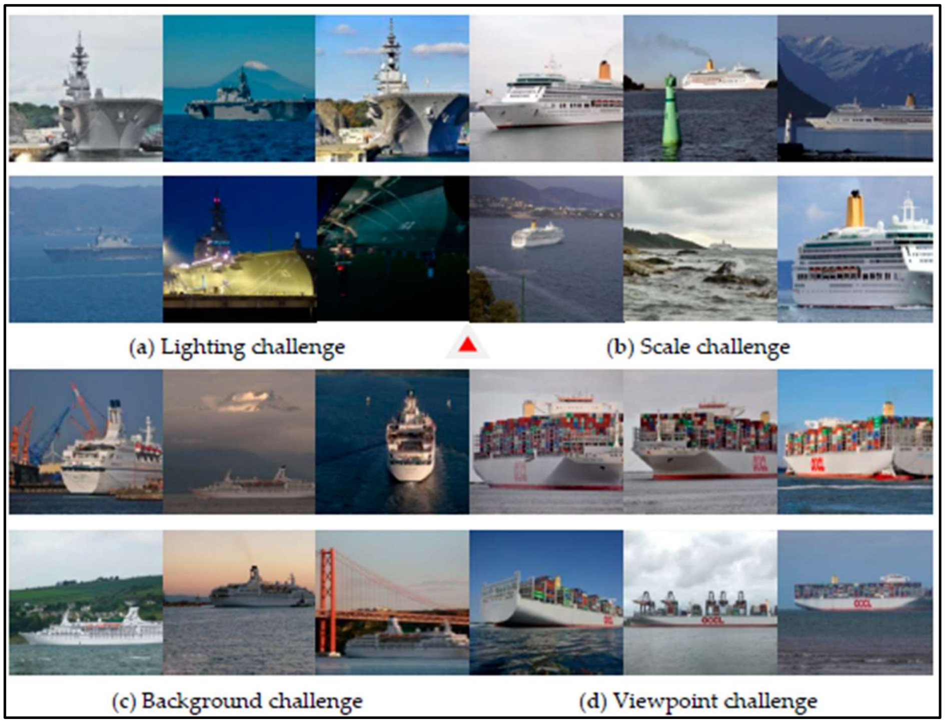


Figure 6 Scene challenges: shows the different conditions found in images taken in the maritime domain. A good detection network would need to work well with all these scenes, making detection in the maritime domain difficult (Quiao, Liu, Lv, Li, & Zhang, 2021)

Figure 6 shows the different situations that object detection networks must surpass in order to be effective in a maritime environment. Lighting challenges can be caused by weather conditions and even the time of day. Morning, noon, and dusk change the way the sun is reflected off objects and the water. Additionally, scale challenges are affected by the distance of objects to the camera. Another challenge has to do with background objects in the image. Background objects can confuse the network when running inference on images. Another challenge is the viewpoint or orientation of objects relative to the camera. For a neural network to perform well under all these situations the dataset must have all of these conditions present in it or augmentations can be done to simulate of these conditions.

Datasets in channels

There is greater difficulty when detections must occur in inland rivers or waterways than in an open water setting. In an inland river setting there are more objects of interest in the foreground and background as opposed to an open ocean setting with a clear scene and perhaps a few ships. With more background and foreground objects present the detector faces more difficulties in isolating the object bounds. Some of the foreground objects that can be found in inland rivers are mountains, rivers, trees, and onshore buildings (Quiao, Liu, Lv, Li, & Zhang, 2021). The dataset used in this study was collected in an inland river where there are plenty of background objects cluttering the background.

Studies in high dynamic range cameras

One of the few examples of HDR cameras being implemented in robotics is one where HDR cameras were used to detect traffic lights in an urban environment, where the researchers found the HDR cameras helped reduce false positive predictions due to reflections (Wang & Zhou, 2019). Some studies, outside the field of robotics, have compared the effect of CNNs trained on regular standard dynamic range (SDR) images and HDR images. In one of these studies, Mukherjee et al (2021) use datasets like PASCAL VOC and COCO to create synthetic HDR images to train an object detection network and compare it to a network trained with the original SDR images from the dataset. The researchers also used HDR images collected by cameras to test how an HDR trained network would perform in a real world application, but this part of their study did not have a network trained with a set of SDR images of the same scenes to compare against (Mukherjee, Bessa, Melo-Pinto, & Chalmers, 2021). In (Mukherjee, Bessa, Melo-

Pinto, & Chalmers, 2021) there were some advantages to using HDR images for training, the HDR trained network helped detect objects in extreme lighting conditions and outperformed the SDR trained network on the same images. In a similar study, with some of the same researchers, HDR images were mapped and transformed into SDR images to train object detection networks and make them better suited for difficult lighting conditions (Mukherjee, Melo, Filipe, Chalmers, & Bessa, 2020).

Chapter III

Methodology

The methods section will outline the tools used for image acquisition, image annotation, detection performance metrics and the training of neural networks.

Image acquisition

As explained, this study will use one datasets composed of native HDR images taken by an HDR camera and SDR images taken by a fixed exposure camera, the images from both cameras show the same scene and same horizontal field of view. The HDR and fixed exposure datasets used for this study were collected in a previous study exploring sensor fusion of visual and spatial data. The images were collected onboard a deck boat navigating through the Halifax River in the Daytona Beach area in Florida. The cameras used were the Flir Blackly S USB3 for the fixed exposure footage and the HDR camera was the Leopard Imaging LI-IMX390.

Table 1

Camera Information

HDR imager data	
Model	Leopard Imaging LI-IMX390
Resolution (pixel)	2880x1440
Field of view (degrees)	65

Fixed exposure imager data	
Model	Flir Blackfly S USB3 (IMX226)
Resolution	4000x3000 (12 megapixel)
Field of view	65

The table shows the resolution and field of view of each camera used in the study.

Table 1 shows some of the capabilities for the sensors used, it is important to notice the fields of view are equal and the resolutions are different. The footage collected by the cameras was logged continuously in two-minute segments. Each video file was saved with the date in epoch time for the time the segment started recording and then a sequence number for each video. These file names are used to identify corresponding HDR and 4k video segments. To produce images from the video footage, a parsing script was used to extract frames from the camera footage at 2 frames per second, extracted frames kept the name of the video used for extraction and then received a sequence number to identify the order in which the frames were pulled. The scheme looks like “hdr-(video sequence)-(frame sequence). This method was used to collect frames from select videos in the dataset. This amounted to 240 frames per video in almost all cases.

With regards to sensor mounting, both sensors are housed in the same compartment and are facing the same direction to collect images.

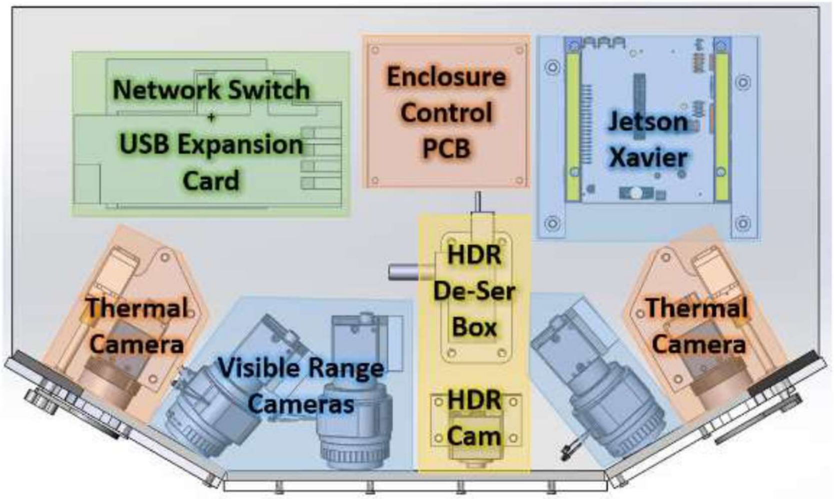


Figure 7 Camera enclosure: shows the layout of the camera enclosure, where the cameras are housed.

Figure 7 shows the layout inside the camera enclosure used to collect the camera footage. The two cameras used in this study are the HDR camera and the visible range camera left of the HDR camera. To clarify, there are three fixed exposure cameras set up at different angles but for this study only the center fixed exposure camera is being used, only the footage for the front-facing fixed exposure camera was annotated and used in the comparison, images from the two side cameras were not used in this study.

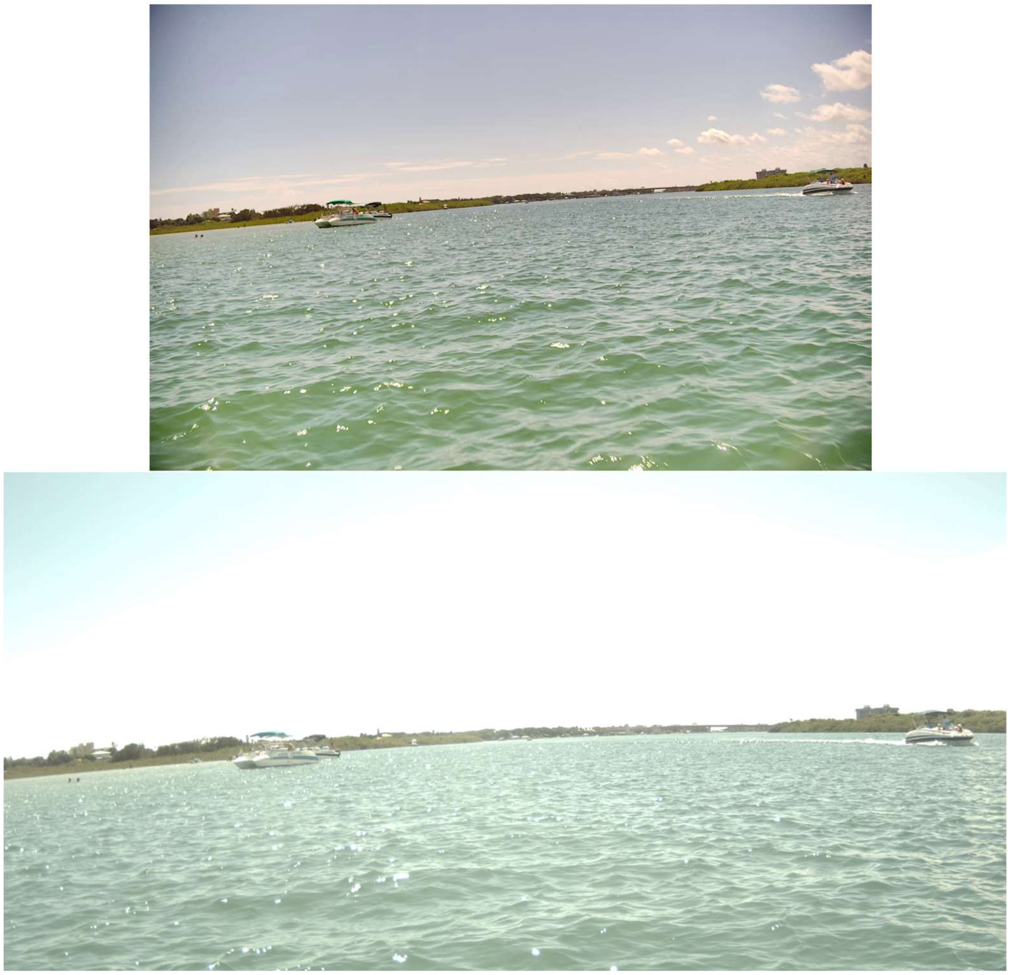


Figure 8 Image scene example. The HDR image is shown at the top and the fixed exposure image is show in the bottom. These images are not cropped but shrunk to fix in the page.

Figure 8 shows an example of data collected by the cameras. The figure shows the same scene as collected by the fixed exposure camera and the HDR camera. Figure 9 shows the locations where images were taken in (Thompson D. J., 2023). Most of the images came from the Daytona Beach and New Smyrna Beach areas. The areas covered for data collection are highlighted in orange. These images were taken from Google Earth.

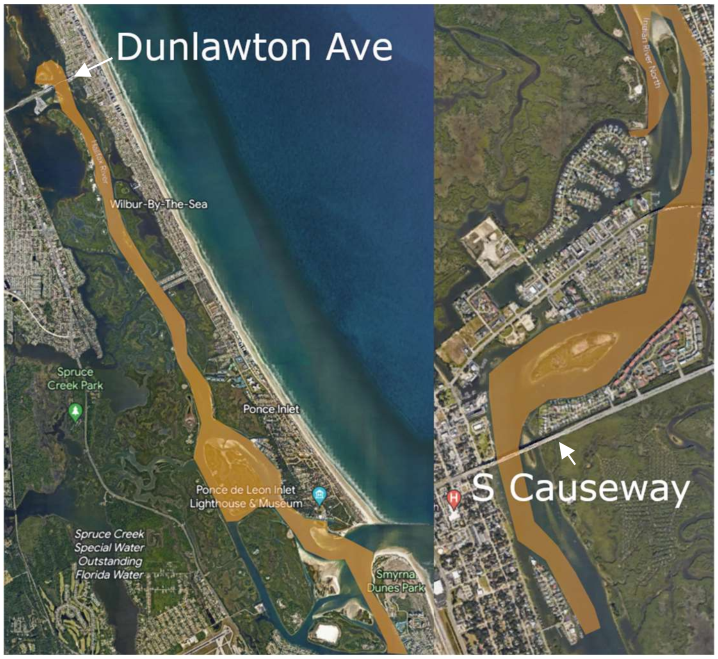

Dunlawton Ave
Wilbur-By-The-Sea
Spruce Creek Park
Ponce Inlet
Ponce de Leon Inlet Lighthouse & Museum
Spruce Creek Special Water Outstanding Florida Water
Smyrna Dunes Park
S Causeway

Figure 9 Data collection routes. Shows the maritime routes where data was collected for the study. The left image shows the route for data collected on October 16, 2021, and the right image shows the data collected on September 25, 2021. The names of prominent bridges are shown in the image.

Labelling tools

A helpful tool in this project was the machine learning platform ‘Supervisely’. With supervisely images can be stored and annotated in the cloud. This platform helps obtain statistics on the annotation data and allows different users to annotate simultaneously with any computer that can connect to the internet. Supervisely facilitated getting annotation help in this study. Supervisely was used to store frames extracted from the video footage. Once the images were in supervisely these were labelled according to the classes discussed below.

Annotation filter

A filter was applied to annotations in the dataset for objects that were smaller than 1000 pixels in area. This filter was applied after metrics revealed objects with areas less than 32^2 pixels were not being detected well. The filter was applied to both the fixed exposure images and the HDR images.

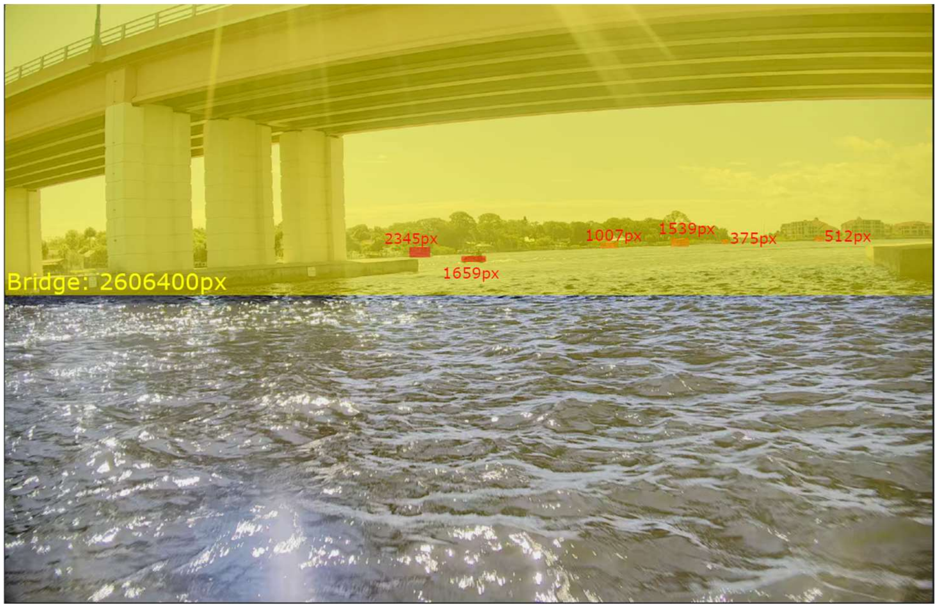


Figure 10 Annotation size example. Shows an annotation with the yellow bounding box showing bridge objects while the red bounding boxes show boat objects. The area of these bounding boxes are shown near the classes.

Figure 10 shows an example of an annotation and the size of the objects. The two boats in the image with a bounding box area of 375 and 512 pixels would be removed from the dataset with this filter.

Proposed classes

Since this study is concerned with the maritime domain, the classes selected represent objects vessels need to avoid or recognize for navigation. Water channels prove to be more difficult than an open water setting due to background objects that may be affecting features of interest. The classes selected in the study are in table 2.

Table 2

Annotation classes

Classes
Boat
Dock
Buoy/channel marker
Bridge

Shows the list of object classes considered for annotations in the dataset. The objects in the dataset were labelled using these classes.

All boats in the study were considered to be one class regardless of type: deck boats, sail boats, yachts, catamarans, pontoons, and jet skis all belong to the boat class. The decision to make all boats one class is to generalize boat instances and because identifying different boat classes would require a lot of instances of each class. The bridge class includes any bridges in the dataset. The dock class included piers, dock houses, and decks. Lastly, channel markers and buoys were combined into one class since there were not many instances of these objects individually. Some classes that were considered but left out of the study were ‘person’ and ‘aerial.’ There were not enough instances of the person and aerial objects for the detection networks to learn them.

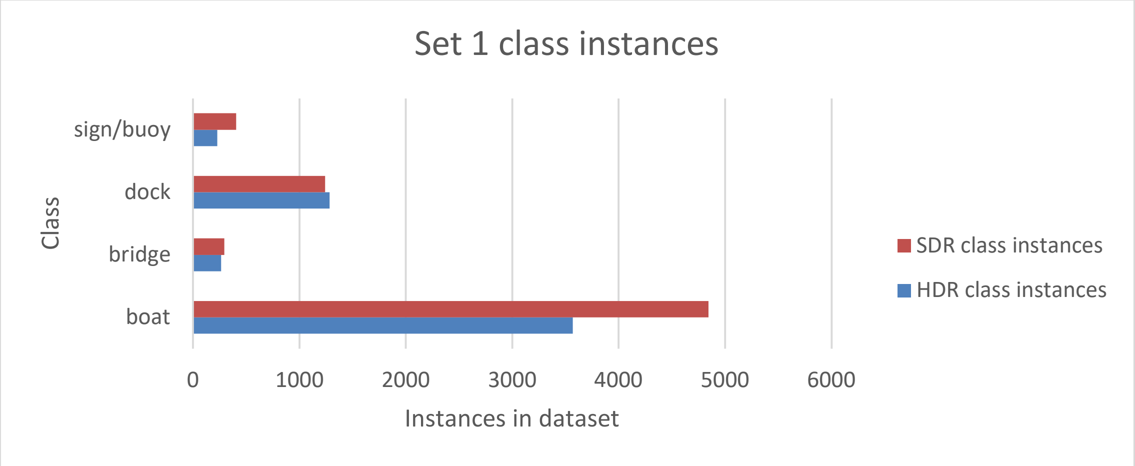


Figure 11 Class instances: shows class instances for HDR and fixed exposure images.

Figure 11 shows how classes were distributed in the fixed exposure and HDR set. It should be noted that most of the samples in the dataset are boats. There is a considerable difference in boat instances between the SDR set and the HDR set, this is due to a decision to filter out labelled objects with an area less than 1000 pixels from the HDR and SDR images. The decision to filter out objects with areas less than 1000 pixels was made after detection metrics from the HDR images showed that the regional CNNs were not detecting small objects well. The difference, in boat labels, is due to the 4000x3000 pixel resolution of SDR images and the 2880x1440 pixel resolution of HDR images. Because the SDR resolution is bigger the same objects were not affected by the filter since they appeared bigger in the SDR image. These objects were not removed from the SDR set because they were clearly visible and might hurt the training process if removed.

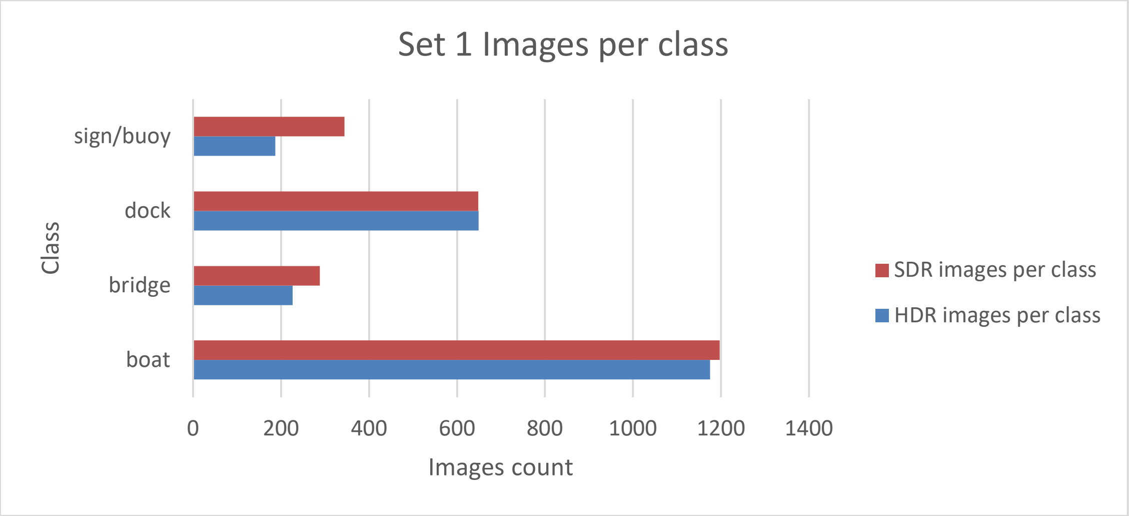


Figure 12 Images per class. Shows how many images have at least one instance for any class in the fixed exposure and HDR sets.

Figure 12 shows how many images there were per class in the dataset, so it shows how many images contain at least one instance of a specific class. It differs from figure 11 since figure 11 shows the total amount of objects that were of a certain class while figure 12 shows the number of images that contain at least one object of a specific class. Discrepancies here can also be attributed to the 1000-pixel area filter applied to the labels.

TensorFlow 2

To train the object detection networks the TensorFlow2 (TF2) object detection API was used. Object detection networks can be trained using pipeline configuration files through the TF2 object detection API. Training tools and hyperparameters for the networks can be selected through the protocol buffers in the pipeline configuration files.

Feeding the training images/labels to the training process was done through TF record files. This means the datasets were partitioned, 85% for training and the rest for validation, and later converted from xml and image files into the binary TF record file, meant to save space and improve efficiency.

Another reason to use TensorFlow is the use of transfer learning. Transfer learning depends on pre-trained models, pre-trained models are trained on large datasets with varying classes and have already generalized on different objects. Transfer learning takes advantage of the features and generalizations learned by the pre-trained models and trains these models to detect specific objects for new tasks. In transfer learning, the layers of the pre-trained model are frozen (meaning new training does not change them, retaining what the pre-trained model learned) and new layers trained on the new task are added. The new layers are trained to output the new classes of interest. The benefit from of this method is it helps avoid having to train the model from start, reducing the amount of time and images needed to get results (TensorFlow, 2022). Pre-trained models can be accessed from the TF2 Detection Model Zoo. The Model Zoo has different model architectures with the pre trained weight checkpoints as well as the configuration pipelines needed to edit the model for transfer learning. The disadvantage of using pre-trained models, in this study, is the datasets use SDR images and the feature extraction might not be ideal for HDR images. The effects of this are explained in the recommendations section.

After the models are trained, they can be exported into TF2's saved model format. At this point the model is a frozen model so that it can't be trained anymore but it is ready for inference on images.

Pre-trained networks

As mentioned in the previous section, the pre-trained models used in this study originated from TensorFlow model zoo in GitHub. All the models provided in the model zoo are trained on COCO17. The dataset contains 80 classes, some of these classes are cars, motorcycles, boats, airplanes, trucks, bus; some classes are animal related and even some appliance/ household objects (Simalango, 2018). Although this does not allow for selection, the COCO dataset has been described as having many small objects which should be an advantage when using this model for transfer learning (Kowlaski, et al., 2021).

Table 3

Pre-trained TensorFlow networks

Model name	backbone	Dataset trained on
Fasrter-rcnn-resnet101-v1-1024-1024	Resnet50	COCO17
ssd-mobilenet-v1-fpn-640-640	Mobilenet	COCO17

Shows the pre-trained models used, the backbone architecture, and the dataset the network was trained on.

Faster-RCNN with resnet101 was selected because studies in the literature claimed it outperformed models compared against it. The specific version of the SSD model was chosen because it was the most similar to the original paper, among the model zoo models.

COCO Metrics

The common objects in context (COCO) detection metrics were used to evaluate the performance of the trained models in this study. COCO was used because it provides metrics based on the size of objects as well as metrics based on recall. The metrics also include the mean average precision from PASCAL VOC, a common metric used to evaluate detection model performance. COCO detection metrics use 12 characteristics to analyze the performance of detection models (Context, n.d.). Due to different training methods for YOLOv5, only the COCO and PASCAL VOC metrics will be shown for the YOLOv5 models.

Before diving into the COCO performance metrics it is important to understand three important concepts the metrics are based on. The COCO detection metrics are based on intersection over union (IoU), precision, and recall.

Intersection over union - IoU

IoU is widely used in object detection models and segmentation models to measure how close the bounding box predicted by the detector is to the ground truths in the dataset. The formula for IoU is simply the area of overlap divided by the area of the union of the bounding box estimation and the ground truth.

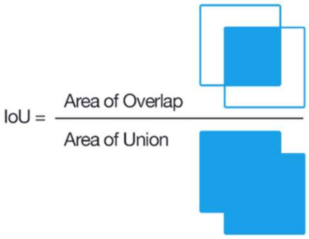


Figure 13 Intersection over union. Shows a visual for the concept of intersection over union (Hasty, Hasty, 2023)

Figure 13 shows a visual of the area of intersection and the area of union between two bounding boxes. The closer the IoU is to 1 the better the prediction or localization of the detector.

Precision and recall

On the other hand, precision can't evaluate the localization or localization accuracy of the bounding box. Instead, it evaluates the class prediction. Precision uses the class predictions of the detector and the classes annotated in the dataset. It uses the concepts of true positive (TP) predictions and false positive (FP) predictions from models. In the case of object detection, TP is when the model correctly predicts an object in the image, a FP prediction is when the model predicts an object that is not present or when the model misclassifies an object.

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

Equation 1 shows the equation for precision. In essence, the precision metric indicates the amount of correct class predictions made over the total amount of class predictions made by the model. Similarly, the recall metric measures how well the model can find ground truths or the objects of interest.

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

Equation 2 shows how recall is calculated. A false negative FN is when a ground truth goes unnoticed by the model. This metric essentially indicates how well a model can find samples in the dataset. These three concepts are what COCO builds on to measure object detection performance.

Table 4

COCO Detection metrics

Average Precision (AP)	
AP	AP at IoU = .5: .05 : .95
$AP^{IoU = .50}$	AP at IoU = .50 (used in PASCAL VOC)
$AP^{Iou = .75}$	AP at IoU = .75
AP Across Scales	
AP^{small}	AP small objects: area < 32^2
AP^{medium}	AP medium objects: 32^2 < area < 96^2
AP^{large}	AP large objects: area > 96^2
Average Recall (AR)	
$AR^{max = 1}$	AR given 1 detection per image
$AR^{max=10}$	AR given 10 detections per image
$AR^{max=100}$	AR given 100 detections per image
AR Across Scales	
AR^{small}	AR small objects: area < 32^2
AR^{medium}	AR medium objects: 32^2 < area < 96^2
AR^{large}	AR large objects: area > 96^2

List of metrics to evaluate object detection performance.

Table 4 shows the detection metrics that will be used in this study to evaluate models, it is meant to outline the factors considered by the performance evaluation metrics. This table should help better understand how the models are being evaluated. The "AP" metric is the mean average precision (mAP) averaged over all categories, it is also average over 10 IoU values, going from .5 to .95 in .05 percent steps. The $AP^{IoU = .50}$ and $AP^{Iou = .75}$ are not averaged over several IoU values, they just use that threshold. The COCO metrics include the average over different metrics because it rewards detectors that localize well (Context, n.d.).

Image size and batch size

An important hyperparameter for training networks, image size and batch size help determine the duration of training and can increase performance. Image size is the dimension raw images are shrunk to when inputted to the neural network for training. Larger image sizes retain more information, this helps in feature extraction. The batch size is the number of examples considered by the learner every step update of the weights. These hyperparameters were selected in a way that would prevent out of memory errors and increase model performance.

Table 5

Batch size and image size selection

Network	Image size (square)	Batch size
Faster-RCNN	800px	6
SSD	640px	6
YOLO	800px	6

Shows the batch size and image size used for each network in the study. The image size is squares so 800px means an 800 by 800px image input.

Anchor boxes and stride

Another hyperparameter, anchor boxes determine the bounding box shapes used by networks to find objects in images. The aspect ratios of anchor box shapes were determined using ground truth bounding box heights and widths. The heights and widths of all annotations in the dataset were used as data points and the K-means algorithm (Matworks, n.d.) was used to separate the data into clusters. The mean/center of these clusters are used to calculate the aspect ratios for the anchor boxes in the training, the algorithm was set to find seven clusters in the data. The new aspect ratios then replace the already existing aspect ratios in the model's configuration file.

Table 6

Aspect ratios

Image set	Aspect ratios
HDR	0.14, 0.36, 0.5, 1, 1.6, 3.0, 3.7
Fixed exposure	0.17, 0.42, 0.5, 1, 1.6, 2.0, 3.7

Shows different aspect ratios used making anchor boxes for training.

Table 6 shows the different aspect ratios used for training. It is important to not that these values must be recalculated when new data is added.

Width stride and height stride is another hyperparameter related to anchor boxes. These strides determine the pixel offset from center to center of the anchor boxes as they are distributed along feature maps (Huang J, 2017). Experimental tests found a width and height stride of 16 was best for model performance. As shown in figure 14.

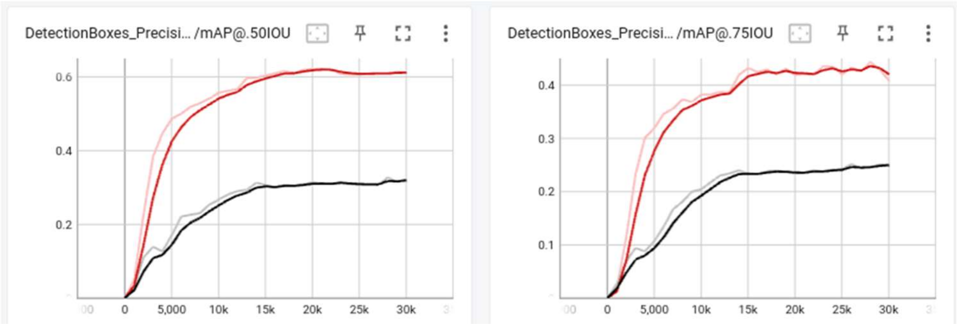


Figure 14 Stride effects. Shows the comparison of two models using different stride values. The red curve represents a model with width/height stride of 16 whilst the black curve shows a model with a width/height stride of 8. Other hyperparameters were equal.

Learning rate and optimizer

The learning rate for the training of the models was set on a schedule. In this scheme, the learner is given an initial learning rate and the learning rate changes after a predetermined amount of training steps are reached. Several learning rates can be implemented in one training session with the schedule method. The optimizer used for these models was the ADAM optimizer because it has become a standard in neural network training as it combines the momentum and RPM optimizers. For training an initial learning rate of .0003 was used and at 9000 num steps the learning rate was changed to .00002 until the training ended at 15,000 num steps.

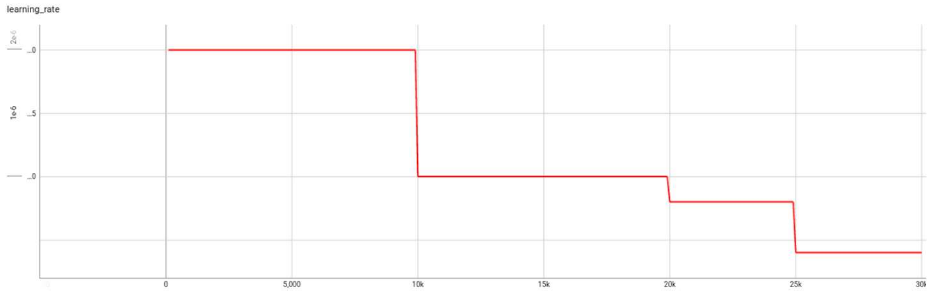


Figure 15 Learning Schedule. shows a learning schedule graph during training.

Data augmentation

Different data augmentation methods were used to train the 3 different networks to suite their architectures. For Faster-RCNN only horizontal flips were used in the image since the network performs well with just original images and horizontal flips (Liu, et al., 2016). For the SSD the default augmentations in the pipeline were used, these were horizontal flips and random crops. For YOLOv5 mosaic and crops were used. Mosaic crops make new images by combining images together, this helps combine classes that normally would not be together in the same image.

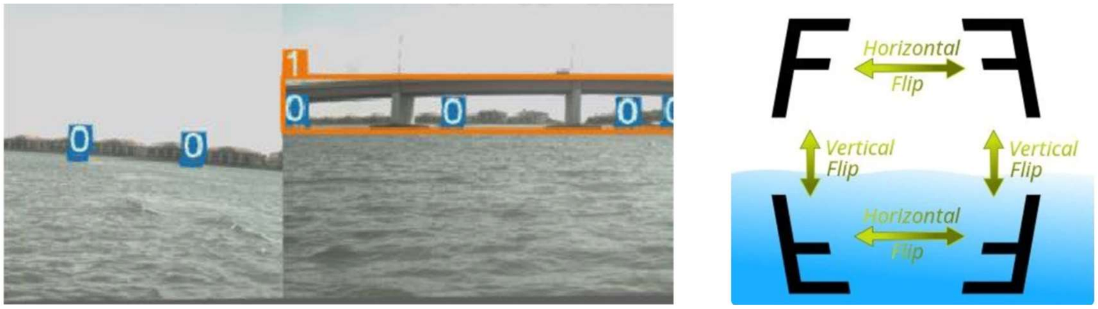


Figure 16 Augmentations. Shows a mosaic augmentation on the left and shows different flips on the right **(Hasty, Horizontal Flip, 2022)**.

Figure 16 shows mosaic augmentation and a depiction of flips. Vertical flips were not used for data augmentation because it would assume the vessel capsized.

Chapter IV

Results

After different training sessions, the best results from the selected detection networks are shown in table five.

Table 7

Results Table

	Faster-RCNN		**SSD**		**YOLOv5**	
Metric	HDR	SDR	HDR	SDR	HDR	SDR
mAP	.69	**.751**	.557	.578	.58	.64
mAP^{50}	.900	**.961**	.769	.811	.85	.93
mAP^{75}	.776	**.841**	.584	.601		
mAP ^{large}	**.843**	.833	.809	.782		
mAP ^{medium}	.547	**.604**	.334	.286		
mAP^{snall}	.227	**.407**	.315	.183		
AR^{large}	**.879**	.807	.848	.833		
AR^{medium}	.637	**.705**	.491	.496		
AR^{small}	.343	**.642**	.294	.233		

Shows the detection metrics achieved by each network and the image type used to train each network. Columns under 'HDR' represent networks trained using HDR images and columns under SDR represent networks trained using fixed exposure images. The metric column shows the metric criteria used for the comparison.

Faster-RCNN Comparison

Detection performance from table five shows the Faster-RCNN network trained on fixed exposure images outperforms its HDR counterpart in all the performance metrics except mAP^{large} and AR^{large}. The fixed exposure network has a clear advantage over its HDR counterpart in terms of metrics. The mAP metric is six percentage points higher for the fixed exposure trained neural network.

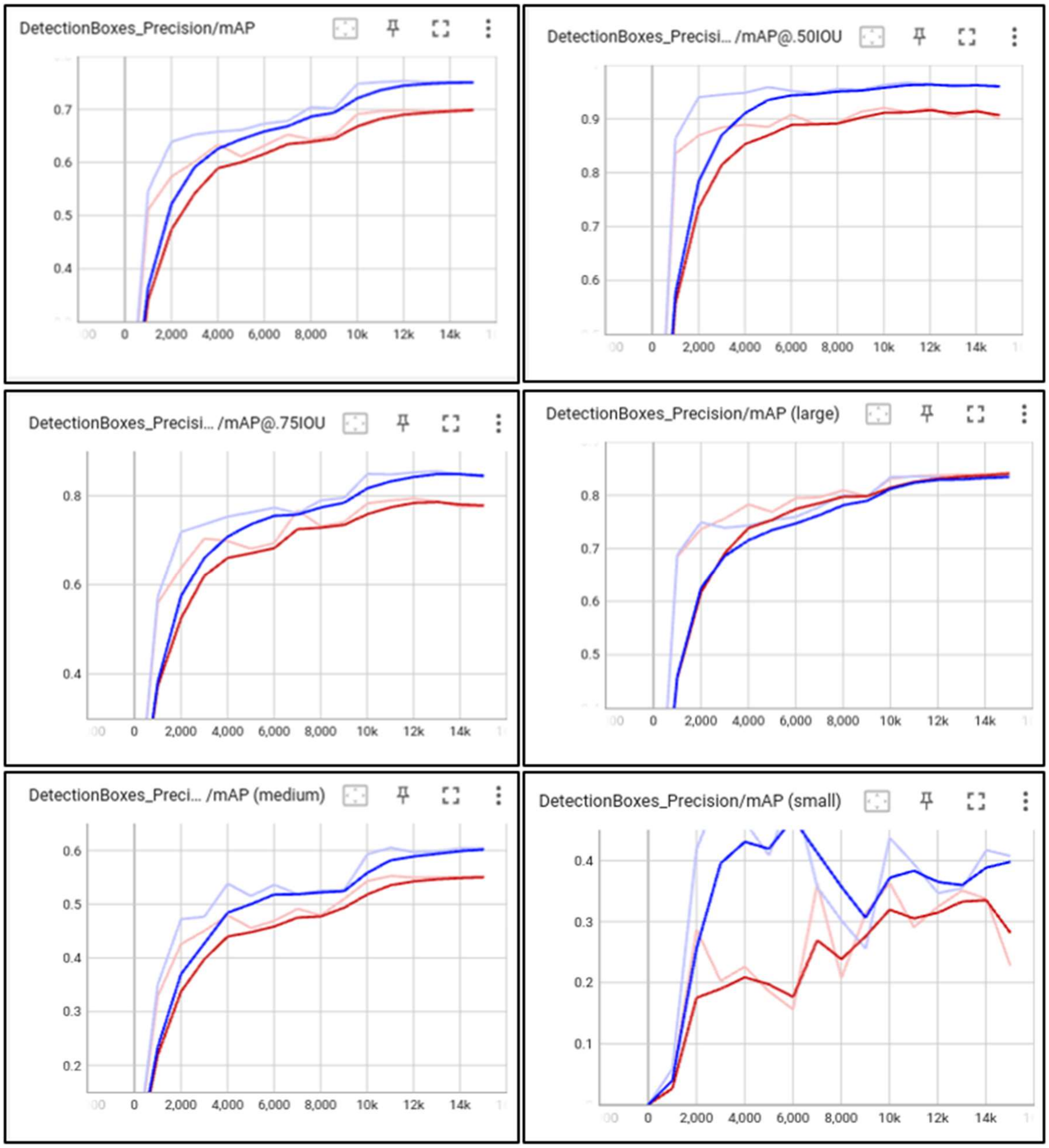


Figure 17 Faster-RCNN comparison. comparison of Faster-RCNN networks trained on HDR images (red) and fixed exposure images (blue)

Figure 17 shows the performance metrics of the HDR and fixed exposure neural networks. The performance is quite similar and in the case of detecting large objects the difference is almost negligible. However, the fixed exposure network largely outperforms the HDR network at detecting small objects. This training method took around 5 hours and 30 minutes to train for each network. The only difference in the training parameters were the anchor boxes and scales used.

These networks have converged, this is indicated by the plateau of the performance metrics. This means these networks have learned all they could from their input data.

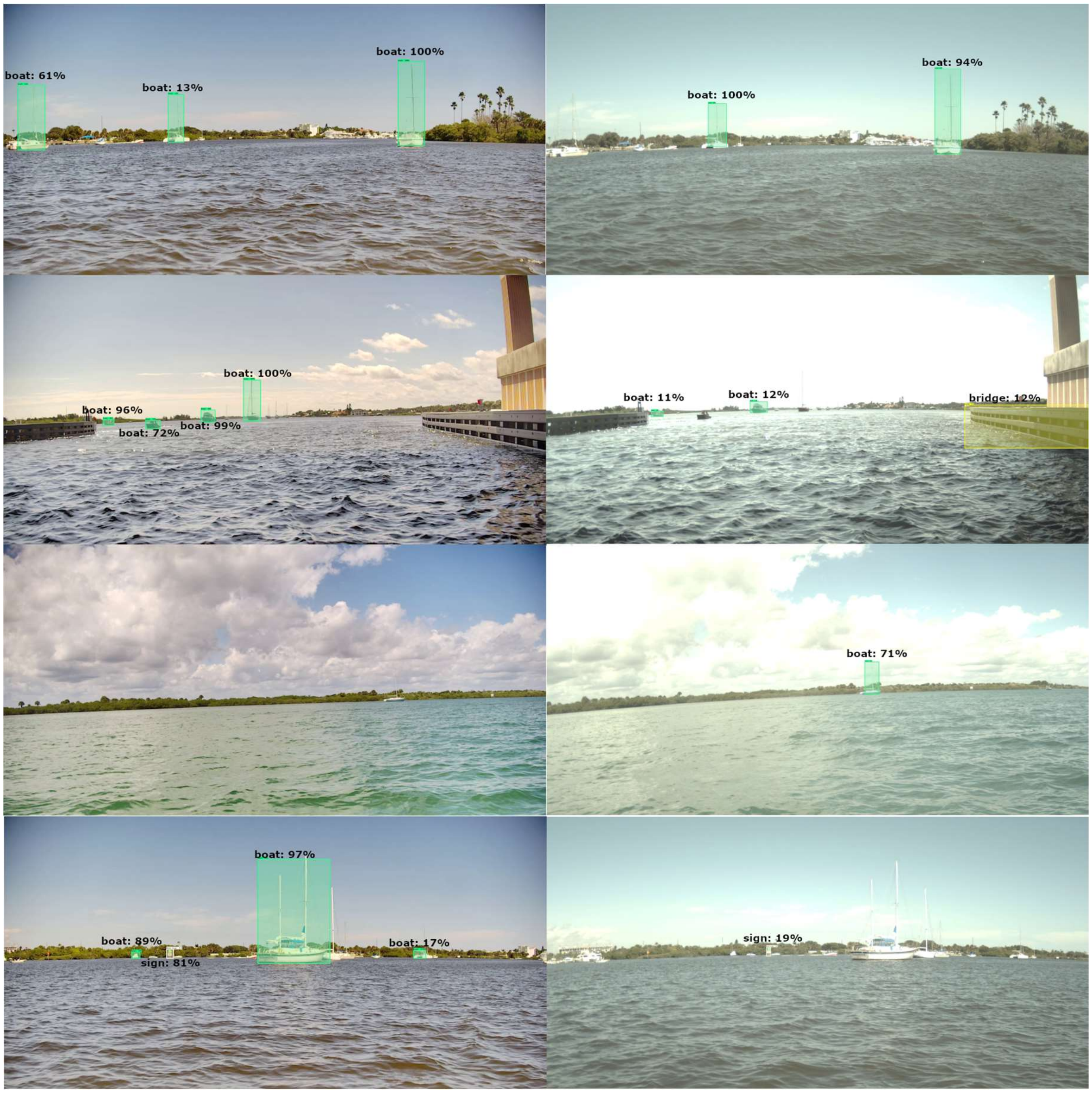


Figure 18 Inference comparison. HDR images (left) fixed exposure images (right) the black text on top of the bounding boxes show what class was detected and the confidence of the detection. The images have been cropped to fit to the page.

Figure 18 shows inference performed on test images. These images belong to video sets that were not included in the training, meaning there are no similar images in either the HDR or fixed exposure training sets. On row one a group of vessels containing catamarans and sail ships are detected well by both networks. In row two the image was taken under a bridge and there is a lot of lighting on the fixed exposure image, making it hard to distinguish some of the boats. The fixed exposure network fails to detect 2 vessels while the HDR network successfully detects all nearby vessels, the HDR performance is probably due to better contrast, compared to the fixed exposure image, between the vessels and the background of the image. In row 3 the HDR network fails to detect the sailing ship while the fixed exposure network tracks it well. Lastly, on row four the HDR network detects boats and marker channel signs while the fixed exposure network fails to detect any. Similar to row two, there seems to be worse contrast in the fixed exposure image than the HDR image. As can be seen in Figure 18, the accuracy of the HDR detections are higher than their fixed exposure counterparts which is common in images with high exposure.

SSD Comparison

The detection performance of the models while using an SSD architecture implies networks trained on fixed exposure images or HDR images are evenly matched. The HDR trained network outperformed the fixed exposure images in all the metrics considering size: $\mathrm{mAP}^{\mathrm{large}}$, $\mathrm{mAP}^{\mathrm{medium}}$,$\mathrm{mAP}^{\mathrm{small}}$ and the AR counterparts. However, the network trained on fixed exposure images outperformed the HDR trained network on mAP, mAP^{50}, and mAP^{75}. This means the fixed exposure network is classifying objects better than its HDR counterpart but not localizing them as well.

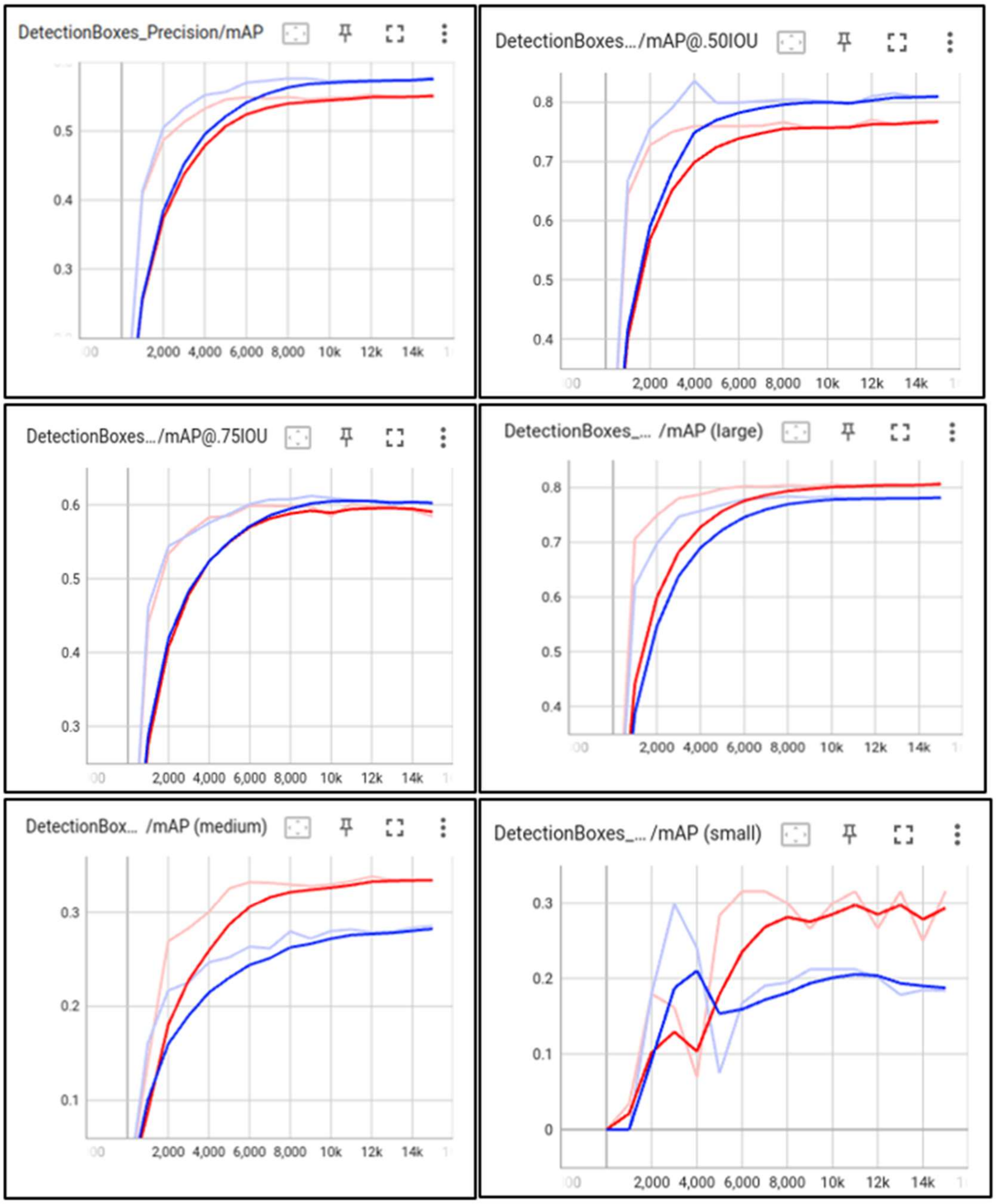


Figure 19 SSD comparison: shows the performance curves of SSD networks trained on HDR images (red) and fixed exposure images (blue) over 15k training steps.

Figure 19 shows the performance of the SSDs. The faded or lightly colored curves are the raw values while the solid curves are the smoothed values of the curves. The performance seems evenly matched in this presentation. The plateaus in the mAP metric indicate these networks have converged, learning all they could from this data.

YOLOv5 Comparison

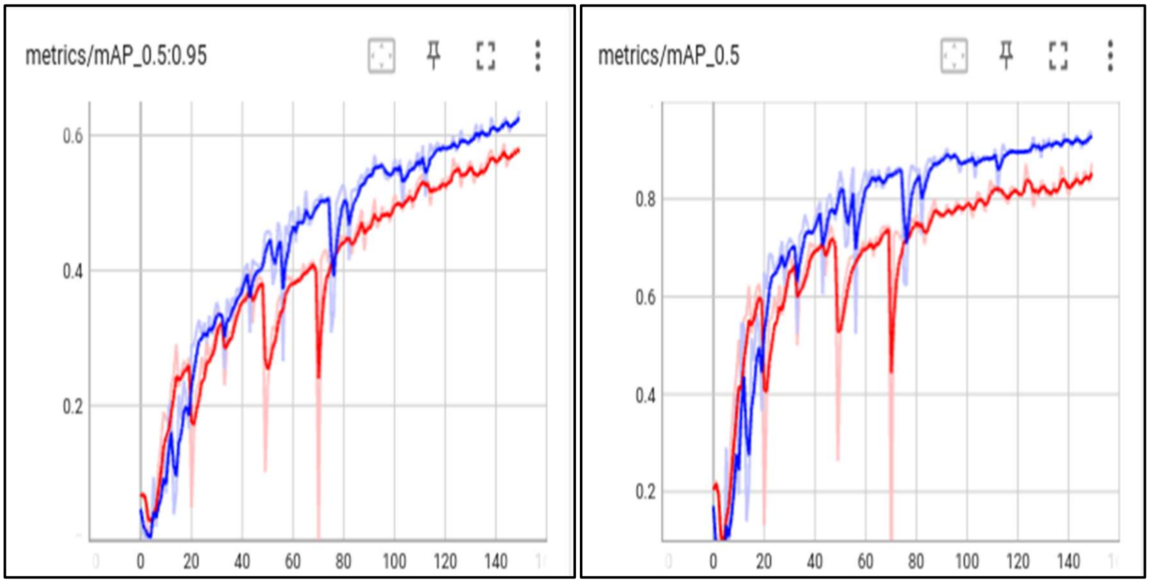


Figure 20 YOLOv5 Comparison. Shows the mAP of the YOLOv5 models on the left and the mAP at IoU 50 on the right. The red curve represents the HDR model, and the blue curve represents the fixed exposure images trained model.

Figure 20 shows the comparison between the YOLOv5 networks trained on HDR and fixed exposure images. In this case the fixed exposure image trained network outperforms its HDR counterpart. These graphs seem different compared to the graphs from Faster-RCNN and SSDs and that is because the training does not happen through TensorFlow so some of the values used to train the network end up being different.

It is likely the fixed exposure networks outperformed the HDR counterparts due to the use of transfer learning and that the pretrained models might need to extract and recognize features slightly different when using HDR inputs, this might have benefited the fixed exposure networks. Another reason for the foxed exposure networks to outperform the HDR counterparts is the implementation of the annotation filter. This gave the fixed exposure networks more instances to learn from.

Contrary to what was expected from the literature review, the SSD architecture outperformed the Faster-RCNN architecture at finding small objects in images according to AP^{small}, as shown in table five.

Chapter V

Discussion, Conclusions, and Recommendations

This section will share the conclusion, recommendations, and discussion resulting from this study.

Discussion

Detection metric results shows the object detection networks trained on fixed exposure images outperform HDR trained networks as can be seen in figures 19 and 20. This means the fixed exposure networks learned the data in the images collected by this study better than the HDR. What these results do not show is the performance of the models in all real-world scenarios. In the few comparisons that could be made in this study (the main dataset only has one full day of matching images for the two front facing cameras) it seems the cameras are evenly matched. What this study fails to capture is the different weather conditions that can be encountered in a maritime setting. More images with fog, rain or during dusk and morning etc. are needed to make a better comparison. Consider the HDR is meant to mitigate the effects of too much sun or too much shadow. Most of the images in this dataset were on a bright sunny day meaning that the camera sensors would pick up image information without difficulty, so the true value of using HDR imagery might not have been fully extracted in these results. Figure 21 shows one of the situations where the HDR is advantageous. In the scene of figure 21 the fixed exposure network and the HDR network both detect the same vessel located on the left side of the image and with the same confidence. However, the mast of the ship is not discernable in the fixed exposure image and so the network can only detect the hull of the ship. On the contrary, the mast is discernible in the HDR image, and the network captured the hull and the mast of the ship. In cases like this HDR imagery outperforms SDR imagery, more examples like this are needed for a better comparison.

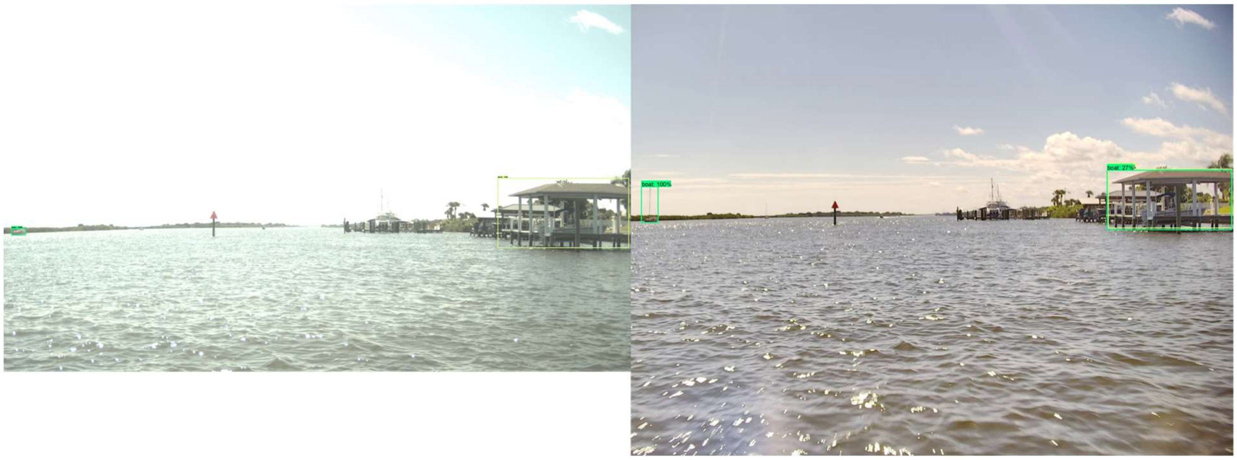


Figure 21 Extreme exposure inference. Shows model inference of a fixed exposure image (left) and HDR image (right). The objects of interest in this image are in the left corner of the image where a boat has been detected in both images. The confidence of the detection of that object is 100% for both networks.

Yet in terms of metrics, using fixed exposure images for object detection in a maritime environment is a good choice.

Another consideration is the possible issue of the classes chosen. The HDR imagery might suffer from fewer classes since the HDR images can carry more detail than the HDR counterparts. Adding more classes can and should be done but it requires more images and more instances per class to be applied properly.

Conclusions

The COCO detection metrics results lead to the conclusion that there is no significant advantage in training object detection networks using HDR imagery over fixed exposure imagery in a maritime environment. The detection metrics on table five show the neural networks trained on fixed exposure images outperform their HDR counterpart. Although the difference in performance is quite small for the SSD networks;

the difference in performance on the Faster-RCNN network is not negligible. Although the object detection networks trained on HDR images were outperformed by the fixed exposure image trained networks, this does not mean HDR sensors should be disregarded in the maritime environment. The HDR sensor still provides capable performance, and it is likely that more data can be extracted from HDR image detections that would be useful in a maritime environment, as shown in figures 18, 19, and 20. The gap in performance is not big enough for HDR imagery to be discarded from a sensor suite and as stated in the discussion section, in cases of extreme exposure HDR imagery seems to do better.

Recommendations

Some changes, and topics to explore on this work would be changing the model training method, increasing the GPUs available, data discrimination, and new object detection architectures.

Image filter

The image filter was applied incorrectly in this study. The image filter does not affect the fixed exposure annotations the same way it affects the HDR annotations creating a discrepancy in instance numbers per class. This provided the fixed exposure networks with more objects to look at. To fix this it might be good to crop the fixed exposure images to the same size as the HDR images.

TensorFlow object detection API as a tool

Although the TensorFlow 2 object detection API and its configuration pipeline tools are useful; these do enforce limits on the control over data and network training. The config pipeline method does not let the user easily ignore ground truth labels based on the size of the bounding box so the annotations themselves have to be changed and

TensorFlow records must be rewritten. Similarly, class weights must be set in the annotations before they are converted to TensorFlow record files and not all conversion files convert the class weights. Class weights were not applied due to this issue and lack of time, class weights could help the networks not overfit on classes like bridges or help learn classes with few instances. These data restriction result in time consuming solutions, using other versions of TensorFlow or pytorch instead of the pipeline configuration method should allow more control over the networks and better results. These different methods allow more control over training pipelines since the training is not dependent on the configuration file and instead done programmatically. This method allows for options without having to change the image annotations themselves, like class weights or discarding annotations based on size.

GPU constraints

With regards to the GPU, training for this study was mainly done on a single Nvidia Quadro RTX 5000. This restrained some of the hyperparameters like crop size and batch size. Training for the Faster-RCNN architecture had to happen with a batch size of 6 and an image size of 800x800 because anything larger than this would result in out of memory errors. It is recommended to use a GPU with more V-ram or use more than one GPU for the training, this would increase the amount of memory available for training and allow training with larger batch sizes and larger image input sizes. Increasing the image size would most likely increase the chances of detecting small objects, another benefit is that the mast of sail ships might still be visible in larger resized images, improving detection for these as well.

Image selection process

With hindsight, there should have been a more strict image selection process for the datasets used to train the networks. Originally there were intensions to annotate more videos than what was done but the process is very time consuming and different labels were used at the start of the projects and relabeling these resulted in wasted time. A better approach would be to select specific videos, not consecutive videos, and focus on obtaining frames with different types of vessels and at different angles. As discussed earlier, the HDR images might be more advantageous during extreme exposure conditions, a metric to define extreme exposure conditions should be utilized to find images to compare.

No anchor box architectures

Some object detection network architectures that should be explored are centernet and DETRs (detection transformers) because these do not need anchor boxes for the training. Since maritime objects occur at an ample range of scales, these anchor box less modes might detect maritime objects well.

Transfer learning

Although transfer learning is an invaluable tool that saves time and improves model performance, it might have hindered the results in this study. Since pre-trained models were used in this study, this means the models used features learned from fixed exposure images. The impact of using transfer learning is not known but future studies into this topic should fully train the detection networks using HDR images and then do the comparison. Although this would be a more lengthy and data intensive study, it would discard any doubts about transfer learning.

References

Amed Hashamani, M., & Umair, M. (2022). A Novel Visual-Range Sea Image Dataset for Sea Horizon Line Detection in Changing Maritime Scenes. *Journal of Marine Science and Engineering*, 1-15.

Betti, A., Michelozzi, B., Bracci, A., & Masini, A. (2020). Real-Time target detection in maritime scenarios based on YOLOv3 model. *9th International Symposium on Optronics in Defense & Security.* arXiv:2003.00800.

Bouma, H., Lange, d., J., D.-J., van den Broek, S. P., Kemp, R. A., & Schwering, P. B. (2008). Automatic Detection of Small Surface Tartgets with Electro-Optical Sensors in a Harbor Environment. *Proceedings of SPIE*, (pp. 711402-711408).

Bovcon, B., Muhovic, J., Pers, J., & Kristan, M. (2019). The MaSTr1325 dataset for training deep USV obstacle detection models. *International Conference on Intelligent robots and systems* . Macau China.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., . . . Leaonar J., J. (2016). Past, Present, and Futrue of Simulatanous Localization and Mapping: Towards the Robust-Perception Age. *IEEE transactions on Robotcis, 32*(6), 1309-1332.

Context, C. O. (n.d.). *COCO*. Retrieved from https://cocodataset.org/#detection-eval

council, e. (n.d.). *european union council council of the european union*. Retrieved from https://www.consilium.europa.eu/en/infographics/ukrainian-grain-exports-explained/

Cox, S. E., & Booth, T. (2008). Shawdow attenuation with high dynamic range images. *Environmental monirotring assessment , 158*, 231-241.

drone, S. (n.d.). Retrieved from https://www.saildrone.com/#:~:text=Saildrone%20is%20a%20US%20business,from%20any%20ocean%20on%20earth.

Dwyer, B. (2020, Juky 29). *roboflow*. Retrieved from https://blog.roboflow.com/advanced-augmentations/

Environments, A. B. (n.d.). Moosbauer, Sebastian; Konig, Daniel; Jakel, Jens; Teutsch, Michael.

Girshick, R. (2015). Fast R-CNN. *Cornell University arXiv*.

Girshik, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Cornell University Library, arXiv.org*.

Group, S. S. (2023). *Quad Bayer Coding*. Retrieved from Sony.com: https://www.sony-semicon.com/en/technology/mobile/quad-bayer-coding.html

Haghbayan, M.-H. F., Poikonen, J., Laurinen, M., Nevalainen, P., Plosila, J., & Heikonen, J. (2018). An efficent multi-sensor fusion approach for object detection in maritime environments. *International Conference on Intelligent Transportation Systems*, 2163-2170.

Hasty. (2022, december 21). *Horizontal Flip*. Retrieved from Hasty.ai: https://hasty.ai/docs/mp-wiki/augmentations/horizontal-flip

Hasty. (2023, February 7). *Hasty*. Retrieved from https://hasty.ai/docs/mp-wiki/metrics/iou-intersection-over-union#:~:text=To%20define%20the%20term%2C%20in,matches%20the%20ground%20truth%20data.

Huang J, R. V. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *CVPR*.

Hunstberger, T., Aghazarian, H., Howard, A., & Trotz, D. (2011). Stereo vision-based navigation ofr autonomous surface vessels . *Journal of field robotics*, 3-18.

Keunhwan, K., Kim, J., & Kim, J. (2021). Robust Data Association for Multi-Object Dtection in Maritime Environments Using Camera and Radar Meassurements. *IEEE Robotics and Automation, 6*(3), 5865 5873.

Kowlaski, M. L., Palka, N., Mlynczak, J., Karol, M., Czerwinriska, Elzbita, . . . Brawta, S. (2021). Detection of INflatable boats and people in thermal infrared with deep learning methods. *MDPI Sensors*.

Kuwata, Y., Wolf, M. T., Zarzhitsky, D., & Huntsberger, T. (2014). Safe maritime autonomous Navigation with COLREGS, using velocity objects. *IEEE Journal Oceanic Engineering, 39*(1), 110-119.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & C, B. A. (2016). SSD: Single Shot MultiBox Detector. *Computer Vision-ECCV 2016: 14th European Conference.* Amsterdam.

Machines, S. (n.d.). *sea machines*. Retrieved from https://sea-machines.com/hydro-international-case-study-more-data-with-less-effort-and-risk-2/

Maindola, G. (2021, 08 27). *A brief history of YOLO object detection model from YOLOv1 to YOLOv5*. Retrieved from MLK Making AI Simple: https://machinelearningknowledge.ai/a-brief-history-of-yolo-object-detection-models/#YOLOv1_8211_The_Beginning

Matworks. (n.d.). *kmeans*. Retrieved from Mathworks: https://www.mathworks.com/help/stats/kmeans.html

Moosbauer, S., Konig, D., Jakel, J., & Teutsch, M. (2019). A Benchmark for Deep Learning Based Object Detection in Maritime Environments. *IEEE/CVF Conference on computer vision and pattern recognition workshops*, (pp. 916-925).

Mukherjee, R., Bessa, M., Melo-Pinto, P., & Chalmers, A. (2021). Object detection under challenging lighting conditions using high dynamic range imagery. *IEEE Access, 9*, 77771 - 77783.

Mukherjee, R., Melo, M., Filipe, V., Chalmers, A., & Bessa, M. (2020). Backward compatible object detection using HDR image content. *IEEE Access, 8*, 142736-142746.

Nofi, A. A., & Analyses, C. f. (2000). Defininf and meassuring shared situational awareness. Alexandria, VA: CNA Corporation.

Park, J., Yonghoon, C., Yoo, B., & Kim, J. (2015). Autonomous collision avoidance for unmanned surface ships using onboard monocular vision. *Oceans 2015 - MTS/IEEE.* Washington.

Prasad, D. K., D., R., L, R., E, R., & C, Q. (2016). Video Processing from Electro-Optical Sensors for OBject Detection and Tracking in Maritime Environement: A Survey. *IEEE Transactions on Intelligent Transportation Systems* .

Prasad, D. K., Krishna Prasath, C., Rajan, D., Rachmawati, L., & Rajadbally, E. a. (2016). Challenges in vide-based object detection in maritime scenario usgin computer vision. *arXiv preprint arXiv:doi1608.01079*.

Quiao, D., Liu, G., Lv, T., Li, W., & Zhang, J. (2021). Marine Vision-Based Situational Awareness Using Discriminative Deep Learning: A Survey. *Journal of Marine Science and Engineering*, 1-18.

recognition, N. (2016, July 27). *Navy Recognition*. Retrieved March 11, 2023, from https://www.navyrecognition.com/index.php?option=com_content&view=article&id=4245

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. *Cornell University arXiv*.

refloated, O. s. (n.d.). *VOA*. Retrieved from https://www.voanews.com/a/firm-says-cargo-vessel-ran-aground-in-egypt-s-suez-canal/6910182.html#:~:text=The%20Panama%2Dflagged%20Ever%20Given,Egypt%2C%20March%2027%2C%202021.

Ren, S., Kaiming, H., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Regional Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*, 1137-1149.

Simalango, M. F. (2018, April 12). *What Object Categories / Labels Are in COCO Dataset?* Retrieved from Amikevile: https://tech.amikelive.com/node-718/what-object-categories-labels-are-in-coco-dataset/

Solawetz, J. (2020, June 29). *roboflow*. Retrieved from https://blog.roboflow.com/yolov5-improvements-and-evaluation/

Soloveiv, V., Farahnakian, F., Zelioli, L., Iancu, B., Lilius, J., & Heikkonen, J. (2020). Comparing CNN-Based Object Detectors on Two Novel Maritime Datasets. *IEEE International Conference on Multimedia and Expo Workshops (ICMEW).* London, UK: IEEE.

statistics, B. o. (n.d.). *on national maritime dan and every day, US economy relies on waterborne shipping*. Retrieved from https://www.bts.gov/data-spotlight/national-maritime-day-and-every-day-us-economy-relies-waterborne-shipping

TensorFlow. (2022, 12 15). *TensorFlow*. (Google) Retrieved from https://www.tensorflow.org/tutorials/images/transfer_learning

Thompson, D. J. (2017). Maritime Object Detection, Tracking, and Classification Using Lidar and VIsion-Based Sensor Fusion. *Embry-Riddle Aeronautical University Scholarly Commons*, 377.

Thompson, D. J. (2023). Neural Network Fusion of Multi-Modal Sensor Data for Autonomous Surface Vessels. *ERAU Scholarly Commons*.

Wang, J.-G., & Zhou, L.-B. (2019). Traffic Light Recognition With High Dynamic Range Imaging and Deep Learning. *IEEE Transactions on Intelligent Transportation Systems, 20*(4), 1341-1353.

Zhan, R., Li, S., Ji, G., Li, J., & Pan, M. (2021). Survey on Deep Learning Based Maritime Object Detection. *Journal of Avanced Transportation*, 1-18.

Zhang, Y., Ge, H., Lin, Q., Zhang, M., & Sun, Q. (2022). research of maritime object detection method in foggy environment based on improved src-yolo. *MDPI Sensors, 22*(20), 7786.