Joseph C. Afonne

Towards Sustainable Oceans: Deep Learning Models for Accurate COTS Detection in Underwater Images

School of Technology and Innovations
Master of Science in Technology
Master's program in Industrial Systems Analytics

Vaasa 2024

UNIVERSITY OF VAASA
School of technology and Innovations
Author: Joseph C. Afonne
Title of the Thesis: Towards Sustainable Oceans: Deep Learning Models for Accurate COTS Detection in Underwater Images
Degree: Master of Science and Technology
Program: Industrial Systems Analytics
Supervisor: Rayko Toshev
Year: 2024
Pages: 80

ABSTRACT :
Object detection is one of the main tasks in computer vision, which includes image classification and localization. The application of object detection is now widespread as it powers various applications such self-driving cars, robotics, biometrics, surveillance, satellite image analysis, and in healthcare, to mention just a few. Deep learning has taken computer vision to a different horizon. One of the areas that will benefit immensely from deep learning computer vision is the detection of killer starfish, the crown-of-thorns starfish (COTS). For decades, this killer starfish has dealt a big blow to the Great Barrier Reef in Australia, the world's largest system of reefs and in other places too. In addition to impacting negatively environmentally, it affects revenue generation from reef tourism. Hence, reef managers and authorities want to control the populations of crown-of-thorns starfish, which has been observed to be the culprits. Deep learning technique offers a real-time and robust detection of this creature more than earlier traditional methods that were used to detect these creatures.

This thesis work, which is part of a competition for a deep learning approach to detect COTS in real-time by building an object detector trained using underwater images. This offers a solution to control the outbreaks in the population of these animals. Deep learning methods of Artificial Intelligence (AI) have gained popularity today because of their speed and high accuracy in detection and have performed better than the earlier traditional methods. They can be used in real-time object detection, and they owe their speed to convolutional neural networks (CNN). The thesis gives a comprehensive literature review of the journey so far in the field of computer vision and how the deep learning methods can be applied to detect COTS. It also outlines the steps involved in the implementation of the model using the state-of-the-art computer vision algorithm known for its speed and accuracy – YOLOv8. The COTS detection model was trained using the custom dataset provided by the organizers of the competition, harnessing the powers of the deep learning methods such as transfer learning, data augmentation, and preprocessing of underwater images to achieve high accuracy.

Evaluation of the results obtained from the training showed a mean average precision of 0.803mAP at IoU of 0.5-0.95, acknowledging the detector model's versatility in making accurate detection at different confidence levels. This supports the hypotheses that when we use pretrained model, this enhances the performance of our model for better object detection tasks. Certainly, better detection accuracy is one way to detect killer starfish, the crown-of-thorns starfish (COTS) and help protect the oceans.

KEYWORDS: (Crown-of-thorns Starfish, Computer Vision, Object Detection, Deep Learning, Convolutional Neural Network (CNN), YOLOv8, Mean Average Precision).

Contents

1 Introduction 10

1.1 Statement of problem 10

1.2 Motivation for the study 12

1.3 Research questions and limitations of the study 13

1.4 Structure of the study 14

2 Literature Review 16

2.1 COTS biology and ecology 16

2.2 Challenges and limitations of COTS detection 19

2.3 COTS and marine species detection techniques 21

2.3.1 Natural predators and manual methods 21

2.3.2 Machine learning and image processing techniques 22

2.3.3 Shortcomings of traditional methods 29

2.4 Deep learning techniques 30

2.4.1 Convolutional neural networks (CNN) 32

2.4.2 State-of-the-art real-time detection techniques 34

2.4.3 Enhancing COTS detection in underwater environments 36

3 Methodology 40

3.1 The CSIRO dataset 41

3.1.1 Data collection method 41

3.1.2 Exploratory data analysis (EDA) of the dataset 42

3.1.3 Data annotation and normalization 43

3.1.4 Data splitting 47

3.1.5 Data preprocessing and augmentation 48

3.2 Deep learning franework 48

3.3 Evaluation metrics 50

3.4 Efficieny improvement methods 53

3.4.1 Loss function 54

3.4.2 Non-maximum suppression (NMS) 54

4 Experiments and results 56
4.1 Experimental setup 56
4.2 Model training 57
4.3 Results of the experiments 58
4.4 Model inference 63
5 Findings and analysis 66
5.1 Result and performance analysis 66
5.2 Discussions and limitations 69
6 Conclusions 71
References 73
Appendices 80
Appendix 1. Software and library requirements for YOLOv8 80

Figures

Figure 1. Coral reefs contribute to ecology (University of Southern California, 2023). 17

Figure 2. COTS feeding on corals while corals turn white (Clement, R. et al., 2005). 18

Figure 3. A road map of computer vision object detection (Zou, Z. et al., 2019). 23

Figure 4. Traditional computer vision workflow vs deep learning workflow (Mahoney, N. et al., 2019). 32

Figure 5. Basic architecture of CNN (Researchgate net, 2019). 33

Figure 6. The distribution of images in three folders**.** 38

Figure 7. Sample of train.csv data frame showing annotations and number per image. 39

Figure 8. COTS marked in red bounding boxes using coordinates for image 0-45.jpg. 46

Figure 9. COTS marked in red bounding boxes using coordinates for image 0-4538.jpg. 46

Figure 10**.** COTS marked in red bounding boxes using coordinates for image 1-99.jpg. 47

Figure 11. YOLOv8n (nano) architecture (Ju, R. & Cai, W., 2023. 49

Figure 12. Pictorial representation of IoU. 53

Figure 13. Training losses, training precisions and recalls for 200 epochs. 56

Figure 14. F1-Confidence graph. 59

Figure 15. Precision-Recall curve. 60

Figure 16. Sample images after training. 61

Figure 17. Validation sample showing model's prediction on the validation set. 62

Figure 18. validation sample showing the ground truth values of the validation set. 56

Figure 19. The ground truth test set. 64

Figure 20. The predictions of the model on test set. 64

Tables

Table 1. The confusion matrix. 52

Table 2. The comparison of the base model and other parameters. O+P = original plus processed data. 63

Equations

Equation 1. Equation for getting x_max. 45

Equation 2. Equation for getting y_max. 45

Equation 3. Equation for getting x_max. 45

Equation 4. Equation for getting y_max. 45

Equation 5. Equation for getting x_max. 45

Equation 6. Equation for getting y_max. 45

Equation 7. The formula for Precision. 52

Equation 8. The formula for Recall. 52

Equation 9. The formular for F1-Score. 53

Equation 10. The formula for IoU. 53

Abbreviations

GBRF	Great Barrier Reef Foundation
COTS	Crown-of-thorns Starfish
CNN	Convolutional Neural Networks
GPU	Graphical Processing Unit
CPU	Central Processing Unit
CSIRO	Commonwealth Scientific and Industrial Research Organization
UI	User Interface
YOLO	You Only Look Once
mAP	Mean Average Precision
AI	Artificial Intelligence
EDA	Exploratory Data Analysis
IoU	Interface Over Union
RCNN	Regional-Based Convolutional Neural Network
FRCNN	Fater Regional-Based Convolutional Neural Network
RPN	Regional Proposal Network
ROI	Region of Interest
SVM	Support Vector Machine
SSD	Single Shot Detector
NMS	Non-maximum Suppression
OWL-ViT	Object-Word Localization Vision Transformer
VJ	Viola-Jones
DPM	Deformable Part-based Model
SURF	Speeded-Up Robust Features
FAST	Features from Accelerated Segment Test
SIFT	Scale-Invariant Feature Transform
BRIEF	Binary Robust Independent Elementary Features
RFC	Random Forest Classifier

1 Introduction

Deep learning computer vision has been gaining popularity in recent years and has found application in many fields such as surveillance, medical imaging, robotics, self-driving cars, and a lot of other areas. One of such is in the protection of the oceans. The oceans are a home to many species of plants and animals and play a vital role in the sustenance of life on the planet earth. However, they face constant threat from various human activities and other natural factors. For instance, in the Great Barrier Reef, in Australia and in other parts of the world, the health of coral reefs, vital marine ecosystems that support diverse marine life, is in jeopardy because of the devastation by the crown-of-thorns starfish (COTS). Can deep learning methods help to reverse this negative trend by accurately detecting COTS in real-time? Doing so will result in furnishing the managers with the information they need to make an informed decision to save the oceans.

This research endeavors to focus on the critical need for fast and accurate COTS detection methods by harnessing the powers of deep learning and computer vision techniques for object detection. This thesis is part of a competition aimed at using deep learning methods in COTS detection as a sustainable and efficient way to protect the ocean and the ecology. By way of its contributions, this will provide yet another method of tools for marine biologists and conservationists for improved and accurate COTS detection.

1.1 Statement of problem

The Great Barrier Reef, located in Australia, is the world's largest coral reef ecosystem with about 3000 coral reefs, 600 continental islands, 300 coral cays and 150 inshore mangrove islands. It is a home to species of jellyfish, mollusks, worms, fish, sharks, whales, and dolphins (Foxwell-Norton, 2017). However, this ecological beauty is under attack. One of the threats to this interdependence and ecosystem is the devastation of the coral reefs by a type of starfish called the crown-of-thorns starfish. When there is an upsurge in the population of COTS, they feed on the corals, causing substantial damage to the

reef and the entire ecosystem, and as a result this has a knock-on effect on the environment.

Biologists and conservationists have long used various methods to detect this predator starfish, in the bid to control their population, but these have their limitations. In their quest for more reliable and accurate methods to detect COTS, they joined efforts with Google in a competition for deep learning solutions to detect in real-time the crown-of-thorns starfish. This exercise provides them with useful information needed to monitor their population growth and to control their population, such as getting them killed. Can deep learning methods help in any way to reverse this trend? Sure. Deep learning computer vision methods can more reliably and accurately help to detect the killer COTS, helping reef managers, biologists, and nature conservationists to save the oceans and other life-forms dependent on them.

There are various reasons why COTS detection is a challenging task. The varying size, color, and texture and camouflaging nature is sometimes difficult to detect. This is because some COTS are too small to be picked up by cameras. They can camouflage to blend in with their surrounding or environment and elude detection. In addition, the underwater environment can contribute to the difficulty in detection. As the depth increases, lighting decreases, which causes visibility problems. So, marine environment can be challenging due to water turbidity, varying lighting conditions, and distortions caused by waves or current. This makes it difficult to get high-resolution images needed for accurate detection. Then also the challenge of background variability. The reef's diverse and complex background can pose another challenge to COTS detection, distinguishing COTS from the surrounding environment and other underwater animals and objects. This work will consider the methods to improve the quality of the underwater images for improved detection given these challenging factors.

Interestingly, this thesis work offers a solution by using the state-of-the-art deep learning computer vision to detect the killer COTS. Deep learning techniques have shown

themselves to be more reliable, accurate and even outperformed current traditional methods of COTS detection. This research will look at the challenges standing in the way of accurate COTS detection of the traditional methods and how these can be overcome using computer vision deep learning methods. This research will consider the strengths of the deep learning methods with regards to improving the image quality of the challenging underwater environments the images were taken from. Improvement in image quality results in improved detection accuracy and reliability. This work will also justify the choice of YOLOv8 as a state-of-the-art deep learning object detection algorithm to perform the detection.

Since there is no benchmark to measure the performance of the detector in this field so far, this research will apply various advanced deep learning computer vision methods to improve the model and the quality of the underwater images to ensure improved detection. This will be done by fine-tuning the hyperparameters until better results are achieved. Better results are measured in terms of the value of IoU, precision and recall of the models. This tells us that the model can generalize well when deployed and used to perform COTS detection.

The COTS detector was trained using a secondary dataset, the CSIRO COTS detection dataset, made available by CSIRO and released for educational purposes. This is a dataset of underwater images, taken as video data and converted to images. This dataset was fully annotated by the publishers making it easier to be used for training and evaluation of the COTS detector model. The dataset contains data images taken under different underwater conditions to help capture the real-world conditions for the COTS detector model to generalize well.

1.2 Motivation for the study

The motivation for this thesis topic came from the courses - Applied Machine Learning and Artificial Intelligence: Concepts, challenges, and opportunities. These courses were like a steppingstone for me. They presented various machine learning and deep learning

techniques that could detect patterns from image data. The teacher presented concepts from computer vision, especially when he presented a situation where machine learning was used to read patterns from satellite images to make decisions. Our group project was to use TensorFlow deep learning methods to detect whether face masks were worn correctly or not. It was a hands-on project that stimulated my interest.

I started to think about other areas to apply what I had learned. I was contemplating, if image data could be read by machine learning methods, it would be good to do so in real-time using video data. The Great Barrier Reef COTS detection Google competition came as a stimulant. I felt I could learn a lot by working on this project (Kaggle, 2022). Besides, it involves real-time detections on video data using computer vision deep learning methods. After discussing with my Artificial Intelligence teacher, I saw where to go as far as this topic was concerned.

1.3 Research questions and limitations of the study

With the overall picture of this research work in mind, this work will address the following research questions:

- Can transfer learning from pretrained models effectively improve the generalization of COTS detection models across varied and underwater environments?
- Can data augmentation and preprocessing methods improve the visibility and distinguishability of COTS from the reef background in underwater images and compensate for the class imbalance?

These questions will be addressed while developing this research work; other pertinent questions will be answered too.

The study will focus on using deep learning techniques to detect the crown-of-thorns starfish in real-time. From the research onion model, this study follows the deductive approach of research. Based on the two research questions above, the study will test firstly, whether transfer learning from pretrained models effectively improves the generalization of COTS detection models across varied and underwater environments and

secondly, whether data augmentation and preprocessing methods improve the visibility of COTS from the reef background in underwater images and compensate for the class imbalance. The study will use secondary data, the underwater image dataset provided for the Google COTS detection competition. This dataset is made public for educational purposes. The dataset will be used to train the COTS detector to recognize and localize COTS from the images. It is cross-sectional data because it was collected once.

As part of the data limitations, there is class imbalance in the dataset images. The images in the three folders representing different environments of the reefs are not equal. So, this unequal distribution between COTS and non-COTS images can lead to biasness in the models and impact detection accuracy. It is a challenge to ensure that the COTS detector model generalizes when testing with the new and unseen data. Another limitation is seen during training of the models on different computer systems with different computational powers and resources. This same is true of deploying models on different reef monitoring devices. They perform differently on different platforms and devices, and this can affect the accuracy and speed of the detection. Notwithstanding these, this work will add to the available on COTS detection and will provide biologists, nature conservationists, and reef managers with the information needed to control the population of the predator COTS, a step towards sustainable oceans.

1.4 Structure of the study

This is the structure that this study will take. The following will be covered in the various chapters:

Chapter one presents the motivation for this thesis topic, the overview of the Great Barrier Reef and the ecological impact of the devastation by the crown-of-thorns starfish. The Google COTS detection competition is introduced, the dataset, the limitations, and the objective of the study for an improved COTS detection.

Chapter two will be on the review of the methods for existing literature for COTS detection. It will explore the limitations of the current methodologies and the methods of the

deep learning computer vision to overcome these. It will consider what makes deep learning methods so powerful for object detection and the various methods of deep learning to improve both the model and image quality for better COTS detection. This is the gap that will be explored.

Chapter three, methodology, will discuss more about the dataset for the competition, the characteristics, and the annotations on the data. Consideration of methods to improve the visibility of images for accurate detection such as data augmentation and dehazing. This chapter will explain about the data preprocessing steps to prepare the dataset for deep learning model for the study, YOLOv8. Discussion of methods for feature extraction by the deep learning algorithm for the implementation of COTS detector model to ensure accurate detection. There is also an introduction to the metrics for COTS detection model's evaluation.

Chapter four focuses on the deep learning model architecture. It discusses the actual implementation of the COTS detector model using a powerful CNN-based object detection one-stage algorithm, YOLO. YOLOv8 is the latest version of YOLO. It is a deep learning model that can be pretrained with large datasets such as COCO and ImageNet. It justifies the choice of this architecture. It will provide us with the model object and other methods to train, evaluate and test the model to measure its performance. It presents hyperparameter fine-tuning to improve COTS detection.

Chapter five will evaluate the results from the model implementation. Analyzing the experiment results will help us to see how accurate and reliable the model is.

Lastly, **chapter six** of the thesis will conclude with the final remarks for the thesis and what will be the likely way to further the project. This will outline what has been learned from the thesis and possible future research.

2 Literature Review

This chapter examines the related literature in marine life or species detection which COTS detection is a part of. Firstly, I present some information about the biology of COTS and its impact on the ecology. A clearer understanding of this creature and what accounts for the outbreak in population will help in the design of right solutions to detect it. Secondly, we will talk about the challenges and limitations of detecting COTS. There are challenges facing marine species detection generally. We explore what these challenges are and how they affect COTS detection too. Thirdly, we will explore the techniques that have been in use for the detection of both COTS and other marine life. These techniques include both manual and automated methods. Lastly, because of the shortcomings of the traditional computer vision methods for COTS detection, this research offers a better alternative in the state-of-the-art deep learning. The last section of this thesis will be spent on looking at the various methodologies of deep learning that will help to improve the quality of the images for deep learning algorithms for better detection of COTS.

2.1 COTS biology and ecology

As mentioned earlier, Australia prides in their Great Barrier Reef because it is a home to a variety of species of living things, plants, and animals. Nabeelah Pooloo et al., (2021) mentioned three areas that corals reefs are amazing; first, the small fish and other exotic organisms find their food and shelter there; second, the reef controls the levels of carbon dioxide in the ocean; and third, the reef protects the coastal areas from natural threats. These are indeed meaningful benefits to other forms of life on earth. Life on earth will be adversely impacted if these benefits are taken away. Figure 1 shows the beauty and the support for ecosystem in the reefs. Besides the beauty and support for other forms of life on earth, the multi-billion dollars generated as revenue from tourism for Australia, for example, each year add a strong impetus to why Australia should protect this heritage. They contribute to economic growth, development, and job creation. These are the backbones for a healthy economy. Unfortunately, this beauty is under attack by coral

bleaching caused by increase in temperature (global warming) and devastation by the crown-of-thorns starfish (COTS) caused by overpopulation of this type of starfish, the crown-of-thorns starfish.

Figure 1. Coral reefs contribute to ecology (University of Southern California, 2023).

They are reported as a major threat to the corals on the Great Barrier Reef and across the Indo-Pacific. The devastation comes with the population outbreak of these predators. The COTS can devastate the coral reef at an alarming rate. (Cameron S. Fletcher et al., 2021). The population outbreak of the COTS has occurred three times since 1960, and it is believed to come every 15-year interval. The present outbreak is termed the fourth population outbreak, and has been around since 2010 (Babcock et al., 2016). COTS are observed to populate very fast at 10^6 in 1-2 years and in such a situation can strip the reefs of 90% of living coral tissue (Pratchett et al., 2014). That is a large number. Figure 2 shows the crown-of-thorns starfish feeding on corals which eventually turn white. However, since COTS has always been a native of the Great Barrier Reef (not a new creature), the question is, what causes the population outbreak of these predators?

Several hypotheses have been put forward to explain the population outbreaks of COTS. These factors have been observed to occur in combination or simultaneously with other

factors. Babcock et al., (2016) is of the view that many biologists and theoretical ecologists agree that 'no one single factor' is responsible for the outbreaks of COTS. However, the identified factors include among others:

- Biological traits of COTS: These creatures have the natural capacity to reproduce very fast as stated earlier. This reproductive ability by the COTS is described by others as phenomenal. Given the right environmental conditions, COTS multiply rapidly and feed voraciously on the reef (Babcock et al., 2016).
- Run-off nutrients: The run-off nutrients into the Great Barrier Reef have enhanced the high survival of COTS larvae because these nutrients aid the growth of phytoplankton that the larvae of COTS feed on, leading to population outbreaks (Kroon, F. et al., 2021; F. Dayoub et al., 2015). When phytoplankton blooms, COTS multiplies, and more devastation to the reef occurs.
- Overfishing of predatory fish: Some fish in the Great Barrier Reef feed on the larvae and COTS, thereby helping to regulate their population to a minimum for them to have such a negative impact on the coral reef. But when these fish predators are removed by fishing and overfishing, COTS have been found to multiply astronomically, leading to population outbreaks (Kroon, F. et al., 2021).

Figure 2. COTS feeding on corals while corals turn white (Clement, R. et al., 2005).

2.2 Challenges and limitations of COTS detection

There are lots of challenges for accurate and efficient COTS detection. The current methodologies have shown certain gaps as limitations to COTS detection. This section looks at these factors. They include but not limited to:

- Species differentiation and camouflaging nature: Crown-of-thorns starfish have unique and difficult characteristics, such variability in size, appearance, and colors. In as much as bigger COTS can be detected with ease, smaller ones might be difficult. Besides, they are camouflaging in nature, as they can easily blend with the surrounding environment or that of the corals they feed on. So, this underscores one fact, that color segmentation for identification is not reliable. Instead, the texture of the thorns is a potential and reliable feature for COTS recognition and should be used as a feature for identification (Ryan Clement, et al., 2005). These unique characteristics of COTS present a challenge for efficient detection.
- Another main challenge to COTS detection is the problem of environmental variability. Underwater environments are complex and difficult such as varying light conditions, water turbidity, and image distortions often result in poor underwater images. Poor quality images in turn affect the quality of detection. The turbidity can be caused by suspended particles, which can include sediment, silt, clay, organic matter, plankton, and other microscopic materials. These contribute to light scattering in the water, and as a result affect the quality of the underwater images. Enhancing the picture quality can enhance the quality of the detection (Han, F. et al., 2020). Closely following this is the reef's diverse and complex background which makes it difficult to separate COTS and background environment.
- Data annotation and availability can exert pressure on COTS detectability. Annotated data is simply limited, and annotating images accurately is another. Object detection requires annotated data with accurate coordinates for bounding boxes and labeling for COTS. Even when such dataset is available, there is the problem that it is not enough and there is an imbalance in the distribution of COTS and the not COTS (Saleh, A. et al., 2022). This will likely create bias in the model leading to inaccurate detection. Many of the people doing COTS detection use CSIRO dataset which is made available

to the public for educational purposes. Although this is annotated using powerful hardware and software, the images are imbalanced, hence, do not represent the reefs and backgrounds equally (Saleh, A. et al., 2022). This can impact the model's accuracy.

- There is another challenge of limited generalizability. Since we do not have data of all the different areas of the reefs, the model trained with the data that are not representative enough of the environments, might not generalize well when used in real environment conditions.
- In addition to the difficult underwater conditions mentioned earlier, the presence of particles and other aquatic organisms can add noise and interference to the detection of COTS.
- Models behave differently on different devices when deployed in real-time. The differences in running on different devices and platforms can affect the accuracy of the detection.
- Blowers, S. et al., (2020) mentioned another challenge which could affect COTS detection. They called it biofouling on lens in installed cameras. This is because of gradual and continuous deposition of biomaterials on the lens of cameras mounted to get the video or images of COTS or other marine animals. This bio growth can accumulate and impact negatively on images and video. Later in this chapter we will look at various methods to enhance the image quality for better detection by the detector algorithm. This is one of the research questions whether preprocessing methods can improve the visibility and distinguishability of COTS images? In other words, does image preprocessing improve image quality and help us get better detections of COTS from the underwater images?

We have considered the factors that have likely contributed to the population explosion of the predator - crown-of-thorns starfish and the peculiarities of the COTS as complex creatures. In the next section we will discuss the different techniques that have been employed over the years for COTS detection. These methods have been used too for detection of other marine life.

2.3 COTS and marine species detection techniques

Over the decades, various methods have been proposed and deployed to detect these killer predators. They (COTS) pose no threats to the reef when their population is maintained to reasonable levels, because they have co-habited with other life forms in these reefs for millenniums, including the corals. There is limited literature for image processing for accurate COT detection. But there is enough literature on marine or underwater image improvement. Since this is related, this work reviews them and so garners some techniques that can be applied to COTS detection image enhancements. But let us consider the earliest methods for COTS detection.

2.3.1 Natural predators and manual methods

At the very beginning, the reef managers and handlers used various methods to detect and beat down the tide of population outbreaks of COTS. The earliest was the natural methods of using other aquatic animals such as predator snails and fish, and human divers. For example, some snail species have been used to reduce the population of the killer starfish, the crown-of-thorns starfish (Hall, M. et al., 2017). This employed the use of a giant marine snail called the giant triton. This can grow up to 0.5 m in length. This giant snail is known to hunt for the crown-of-thorns starfish (COTS) by scent alone. It can smell the presence of COTS. Even though the COTS has a defense mechanism of hundreds of sharp spines and a toxic coating, the triton beats against this defense system by catching the COTS and secreting a chemical that will kill it and eat it up eventually. The COTS runs away when it perceives the giant triton coming closer. This mechanism was also a control mechanism but was not that effective because these giant snails could only eat a few COTS per week. However, research is still on-going on how to use the smell of the giant snail, the triton, to drive away the predator starfish, the COTS, and possibly reduce their numbers (Hall, M. et al., 2017). Additionally, some predator fish have been used to detect COTS. This was also used to reduce their number to a manageable and non-disturbing level.

Next, there came another method that involved the physical removal of the crown-of-thorns starfish by human divers who were trained to look for them and collect them for removal from the water. They were either burned or buried afterwards. This method involved a lot of work and trial-and-error identifying of the COTS by these trained divers equipped with chemical toxins such as vinegar solution, bleach, copper sulphate. These went deep down in the waters in search of COTS and to inject them with poisonous chemicals or solutions and to leave them to die within few days (F. Dayoub et al., 2015). Like the previously mentioned methods, this was equally ineffective. It was based on the discretion or judgment of the human divers who might not be accurate in identifying COTS because of some reasons. There was the human error factor too based on trial and error. This process was slow and not efficient given the vast areas involved. In the words of F. Dayoub et al., this method is expensive because it uses human divers, and it introduces significant safety concerns limiting dive time and restricting work to daylight hours and calm sea conditions. Sometimes at this depth, human divers find it difficult to see clearly because of light not penetrating well.

2.3.2 Machine learning and image processing techniques

Computer vision techniques were employed to detect COTS as better alternatives to the earliest methods – the natural and human methods of detection. These methods were mainly based on machine learning techniques. They are referred to as traditional computer vision methods. This section discusses the various traditional computer vision approaches applied to COTS detection. The following figure, figure 3, sheds light on the period in the computer vision object detection that these methods dominated the computer vision space.

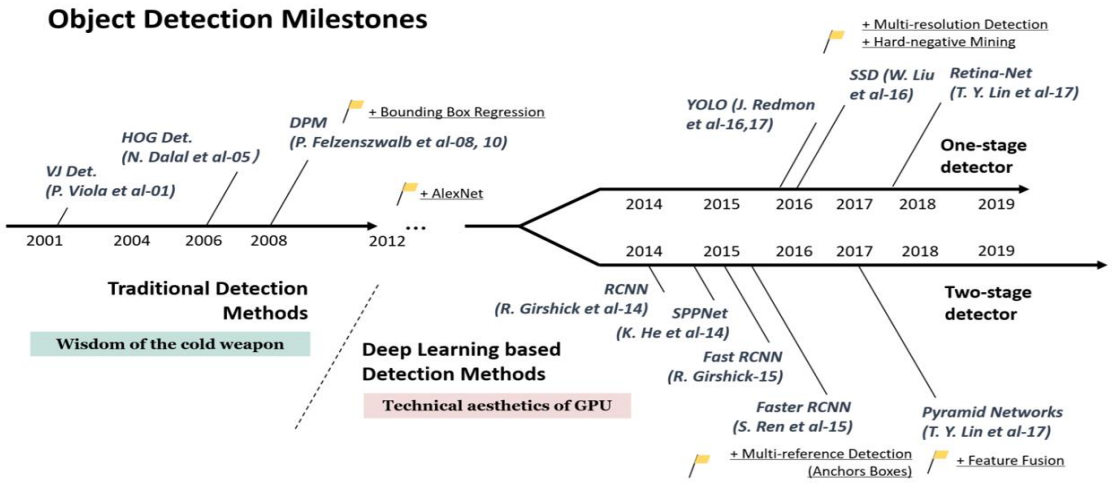

Figure 3. A road map of computer vision object detection (Zou, Z. et al., 2019).

Figure 3 shows that the computer vision techniques used prior to 2014 were mainly the traditional methods. These methods include Viola-Jones Detector, HOG Detector, SIFT, and part-based methods. These were actively used at the beginning of the evolution of computer vision. Efforts were made to use technologies that would increase the accuracy of the detection models and introduce high efficiency into the process. This would have a better impact on the detection of the crown-of-thorns starfish. Since computer vision technology was gaining traction because of its success in various areas of applications, researchers turned their attention to applying computer vision in COTS detection. Let us discuss these methods and what contributions they brought to the table as far as object detection is concerned, and by extension COTS detection.

Scale-Invariant Feature Transform (SIFT) as a feature detector was considered the best computer vision algorithm in the early 2000s (Blowers, S. et al., 2020). It solved the problem that results from scaling images. It overcame the problem of scaling images common with its predecessors, meaning that it works regardless of scale of the image. It does this by detecting the features (a feature transform) and computing the feature vector which describes the region surrounding the features. This is necessary so that the result is not based on the scale of the image. Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and shear (Joseph Howse et al., 2020). However, this algorithm is patented in the United States (Blowers, S. et al., 2020).

Viola and Jones algorithm or detection framework (VJ) is a powerful machine learning technique for object detection proposed by Paul Viola and Michael Jones in 2001 in their paper entitled Rapid Object Detection Using a Boosted Cascade of Simple Features (Asad, H. et al., 2020). Although primarily proposed for face detection problems, it since has been adapted in the detection of other object classes, including COTS. The algorithm extracts relevant features from images. These features make the decision whether an object is in the image or not by sliding a window through all the possible locations and scales in an image. One positive side of Viola-Jones algorithm is its detection speed, the reason why it was used in real-time detection (Blowers, S. et al., 2020). It can achieve detection speed because of three important techniques, namely **integral image**, **feature selection** and **detection cascades**.

Speeded Up Robust Features (SURF) is an alternative to SIFT because it is faster. It uses a different set of feature descriptors based on the sum of the Haar wavelet responses around a specific blob detection (a region of varying contrast or color), according to Blowers, S. et al., (2020). It owes its speed because of utilising the precomputed integral image in the detection. However, like SIFT, it is patented in the United States.

Histogram of Oriented Gradients (HOG) is a powerful feature extractor proposed in 2005 by Navneet Dalal and Bill Triggs which was published in their paper *Histograms of Oriented Gradients for Human Detection* (Zou, Z. et al., 2019; Rahmad, C. et al., 2020). It has the advantage of comparing color and contrast gradient slopes and angles over an image (Blowers, S. et al., 2020). These features are generated from histogram of gradients which are compared to expected histograms of known objects. Originally proposed for object and pedestrian detection, HOG represented an object as a single value vector as opposed to a set of feature vectors where each represents a region of the image, computed by sliding window detector over an image. The HOG descriptor is computed for each position, while the scale of image adjusted to get a HOG's feature (Rahmad, C. et al., 2020).

Local Binary Pattern (LBP) is an image feature detector, primarily introduced for face detection. It was later applied to COTS detection because of its performance on images. It was supposed to overcome the problem of illumination in images. It is obvious that illumination changes affect the image quality because they create incoherence in images and consequently, the performance of the face recognizer reduces (Asad H. et al., 2020). The LBP methods worked well in detecting textures in COTS. When it was used in COTS detection and counting, it achieved 65% and 48% accuracy respectively (Clement, R. et al., 2005). Dayoub, F. et al (2015) reported on a proposal that used LBP in detection and monitoring of COTS from underwater imagery. Since it used the template-based approach (texture), the LBP proved to be a powerful technique in the detection of COTS because of its textural surface. When describing the LBP methods, Asad, H. et al., (2020) highlighted why LBP was successful, pointing out that LBP calculates the LBP values for each pixel by considering the neighboring pixels. After which it converts those values into a histogram. Eventually, there will be a histogram for each image in the dataset. So, LBP and the histogram represent the feature vectors for an image. This provides local features-based robustness against illumination changes and other negative factors. Hence, LBP methods can extract the features from an image.

Gabor Filters search for patterns in an image by passing 2-Dimensional Gaussian filters and observing regions that have similar frequency responses to known patterns (Blowers, S. et al., 2020). The Gabor Filters methods were used for texture analysis and pattern recognition for object classification. These methods were used to recognize and classify marine animals such as fish and other aquatic life in videos.

Oriented FAST and Rotated BRIEF (ORB) is a method for feature extraction developed by OpenCV (Open Computer Vision) library as an open-source alternative for SIFT and SURF which were patented. It uses the FAST (Features from Accelerated Segment Test) algorithm to determine key points and then uses BRIEF (Binary Robust Independent Elementary Features) algorithm to create the feature descriptors (Blowers, S. et al., 2020). This method performs well in rotated objects to improve accuracy.

Integral Image is a technique for speeding up computations related to feature extraction in images (Asad, H. et al., 2020). In other words, Integral Image speeds up the computations related to convolution process. This is necessary because feature extraction was done by considering large rectangular regions of different scales, which then moved over the image. Of course, this generated a lot features, some relevant, some irrelevant. Another technique is also applied, which is called **feature selection** that uses AdaBoost. This process reduces the number of features to a reasonable level, discarding the irrelevant features while keeping the relevant ones. The last technique, **detection cascades**, reduces the computational overhead by spending less computations on background windows (Zou, Z. et al., 2019).

Random Forests Classifier (RFC). This is a supervised machine learning algorithm that is an ensemble of decision trees, created by bootstrap of samples of the training data (Dayoub, F. et al., 2015). In other words, at the time of training the data, random forests classifier constructs a multitude of decision trees and outputs the class selected by most trees. Random forests classifier, also known as decision forests, is used for making prediction for classification and regression problem. Additionally, it can be applied to computer vision problems, such as image classification, image labeling, action recognition and object detection. The output class of the classifier is the mean probabilities of the trees making up the forest. What is the positive side of this algorithm? Separate decision trees are trained separately on a random subset of the training set and participate in the final prediction by the aggregation of the individual decision trees. Another advantage of these methods is that as an ensemble of algorithms, it performs better than one single algorithm (Benjamin Johnston et al., 2019).

In 2015, Dayoub, F. et al. proposed an automated COTS detection and classification robotic system equipped with computer-vision that would be capable of COTS detection based on color and texture of COTS. Their proposal to use a robot was not new, but what was new was the computer-vision techniques used by these robots to detect and classify

objects. Their COTS' tracking system was a machine learning technique based on Random Forest Classifier (RFC) trained on images from underwater footage. The classifier was able to extract the color and texture of COTS from the training images. To track COTS using a moving camera, they embedded the RFC in a particle filter detector and tracker where the predicted class probability of the RFC was used as an observation probability to weight the particles. The robotic arm on which the camera was attached moved at different speeds and heights over real-size images of COTS to mimic reef environment (Dayoub, F. et al., 2015).

The objective was to attach a monitoring system to an Autonomous Underwater Vehicle (AUV) in the reef environment. The monitoring system on detecting COTS would send information to the robot's arm to inject poison into the COTS to kill it. This classifier showed high precision in detecting COTS. They go further to state the advantages of such a robotic system; it eliminates the costs and risks associated with using human divers; a robot could operate tirelessly day and night and even at a depth unimaginable and is immune to sea surface conditions. A robot could use a position system and localize itself no matter the underwater conditions and share useful information about COTS, depth, water temperature, light levels, and terrain complexity with other robots (Dayoub, F. et al., 2015). They suggested this could lead to efficient coverage of the reefs.

Support Vector Machine (SVM). This is a powerful supervised machine learning algorithm that can be used for regression and classification tasks. In his work on the application SVM machine learning in oceanography and earth science, Ahmed, H. (2020) listed various fine tasks accomplished using this machine learning method to analyze ocean data, to recognize patterns in oceanographic phenomena with high accuracy and efficiency. He stated that SVM in some cases performed better than other machine learning models in marine management. For example, SVM was applied to face and speech recognition, face detection, and image recognition tasks and turned out to be successful (Ahmed, H., 2020). A team of researchers used it to make successful predictions for sea-level rise in Brazil (Moura, M. et al., 2010). In their work, Ogunlana, S. et al., 2015 referred

to the application of SVM in marine species recognition and classification. The SVM identified fish based on shape feature, body length, anal fin length, caudal fin length, dorsal fin length, pelvic fin length, and pectoral fin length. Ahmed, H. (2020) mentioned also of the use of SVM in monitoring marine and coastal water quality with a high accuracy.

K-means clustering. This is an unsupervised machine learning algorithm that partitions a dataset into a predefined number of clusters. This algorithm has been used in tasks involving pattern recognition or grouping of similar data points, such in marine life. Data collected through various sources, such as underwater cameras, sensors, or satellite imagery is passed to this algorithm to extract patterns for the recognition and classification of different marine species, habitats, or behavior patterns. Some of the advantages of using K-means clustering are:

- The simplicity and ease of implementation because it is straightforward and computationally efficient to implement on a large dataset.
- It does not require labels for the training data but partitions the data based on similarities in features, as unsupervised learning does not need to be labeled.
- The clustering of similar data points together promotes dimensionality reduction of the features.
- It serves as feature enhancements of the original dataset. Other algorithms use k-means-enhanced features as their data preprocessing step for an improved performance.

Some of the machine learning methods for computer vision object detection and classification have been discussed in this section. Yet, there are still more not mentioned, such as **Linear Discriminant Analysis (LDA)** and **Principal Component Analysis (PCA)** (Moniruzzaman, M. et al. (2017). They are both dimensionality reduction techniques used to model the features for marine life training set. PCA is an unsupervised technique for feature extraction which captures the variance and reduces redundancy in the data, whereas LDA is a supervised feature extraction technique that emphasizes class separability. LDA is widely used in classification as it enhances the data for classification algorithms. Thomas, T., et al., (2021) **mentioned Light Gradient Boosting Machine (LGBM)**

machine learning algorithm being trained on images of corals to enable the detection, classification, or even performance of segmentation of the different species of coral. LGBM has gained popularity in machine learning for its speed, efficiency, and strong predictive capabilities. It handles large and high-dimensional feature datasets efficiently. In this section, we have considered the strengths of some image processing techniques used to improve the performance of image detectability of marine animals such as COTS.

2.3.3 Shortcomings of traditional methods

Traditional computer vision methods, we have looked at many of them, have several limitations that impact on object detection generally, and especially COTS and other marine species detection because of their peculiarities. This section will consider some of these shortcomings.

- **Limited robustness to variations**: We have discussed the challenges of underwater images such variations in lighting conditions, water turbidity and image quality. Traditional methods find it difficult to adjust to these variations and other problems such as occlusions, image distortions and viewpoints. This is a big problem for marine object detection (Wang, N. et al., 2022).
- **Manual feature engineering**: The traditional methods of computer vision use techniques that depend on handcrafted methods for feature extraction which is domain-based. These methods do not capture all the relevant information in the data. This is not easy to design and has the problem of generalization on a different domain dataset (Wang, N. et al., 2022).
- **Scalability**: Scalability issues arise when a large amount of training dataset is involved because of computational limitations which make them unscalable (Qin, H. et al., 2020).
- **Performance in complex environments**: In images with cluttered backgrounds and multiple objects, traditional methods struggle to detect and recognize objects.
- **Fixed architecture**: Their fixed architectures and rigid designs male them lack the ability to generalize on different domains.

To overcome these shortcomings or limitations of the traditional methods, computer vision experts have developed newer techniques. These are deep learning techniques based on Convolutional Neural Networks (CNN). The next section addresses this.

2.4 Deep learning techniques

Mahony, N. et al., (2019) defined deep learning as a subset of artificial intelligence that is based largely on Artificial Neural Networks (ANNs), a computing paradigm inspired by the functioning of the human brain. They continued by stating that like the human brain, it is composed of many computing cells or 'neurons' that perform a simple operation and interact with each other to make decisions. This definition gives a good picture of deep learning. Saleh, A. et al., (2022) in their work added that these several layers of neural network enable it to "learn" from huge quantities of data. The neural network learns by extracting higher-level features from input training data. Because this involves several layers, it is referred to as "deep" networks. The network can even go 'deeper'. The lower layers could detect the edges, whereas the higher layers could identify parts of an object (Saleh, A. et al., 2022). However, an important question to ask is, why do we need to use deep learning techniques as a better way to detect COTS?

Well, deep learning has pushed the limits of what was possible in the domain of digital image processing; in the recent years deep learning approaches have outperformed previous state-of-the-art machine learning techniques in many areas, including computer vision (Mahony, N. et al., 2019; Kaur, R. et al., 2022). It will be good to look at the various advantages for favoring using deep learning in the case of object detection, in our case – the detection of COTS. One of the major advantages of deep learning is that deep learning models can learn automatically from our raw input data without the need for manual feature engineering done by the computer vision engineer, as in the case of using traditional computer vision techniques (Mahony, N. et al., 2019; Zou, Z. et al., 2019). From figure 4a, it is obvious that traditional means require the manual feature extraction and selection of features. It is the computer vision expert or engineer that crafts or designs such techniques that will choose the important features. Domain-specific

knowledge is important and required here. But in deep learning, it is not required because the models take care of the feature extraction, the important features, from the raw data (figure 4b). This is particularly helpful when the number of classes to train the model increases, it becomes more and more difficult for the computer vision engineer to handcraft the feature extractors. Diversity in appearances, illumination conditions and backgrounds, all make it extremely difficult to manually design or handcraft a robust feature descriptor to perfectly describe all objects (Zhao, Z. et al., 2019).

Unlike the traditional computer vision techniques, deep learning achieves higher accuracy in performance. In other words, they are more accurate than the traditional methods in many tasks. This is because deep learning models can learn complex non-linear relationships between input and output variables, which allows them to make more accurate predictions (Mahony, N. et al., 2019; Muller et al., 2019; Saleh, A. et al., 2022). For example, a deep learning method for fish classification achieved a performance accuracy of 87%. It took 6 seconds to identify 115 images (Saleh, A. et al., 2022). Another advantage of using deep learning computer vision is that it can handle large amounts of data more efficiently than the earlier counterpart (the traditional computer vision techniques) (Pathak, A. et al., 2018). This is made possible because of parallel computing techniques (not available in the past) which allows deep learning models to learn complex patterns from large amounts of unstructured data – image, audio, and text data. The learning capacity of deep learning models improves when more data is made available, unlike the learning capacity of traditional models which is fixed, even with more data made available. This is good news in today's world because of the abundance of devices that generate lots of data. Personal phones are good examples.

Deep learning is also flexible because the models are trained rather than programmed. This has the advantage of making it easier for the models to learn some important features which would be impossible to achieve by programming. For instance, it is difficult to handcraft a technique to detect emotions in an image. It is flexible also in that models and frameworks can be re-trained using custom dataset for any use case, contrary to

traditional methods which are more domain specific. Deep learning techniques can be supervised, semi-supervised, or unsupervised. Also, deep learning architectures include convolutional neural networks (CNNs), recurrent neural networks (RNNs) and long short-term memory networks (LSTMs). These have been applied to fields such as computer vision, speech recognition, natural language processing, and medical image analysis. CNN has been a powerful engine behind the revolution we see in the deep learning computer vision. I will spend some time talking more about this architecture and development.

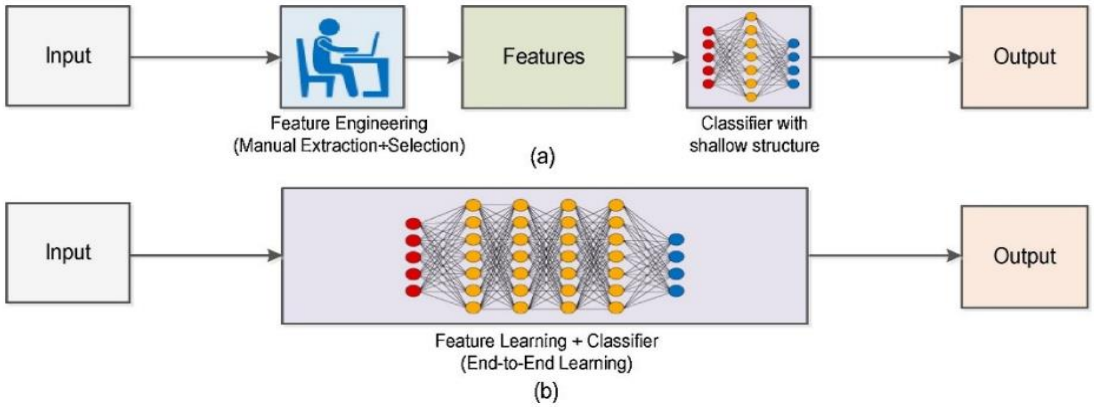

Figure 4. (a) Traditional Computer Vision workflow vs. (b) Deep Learning workflow (Mahony, N. et al., 2019).

2.4.1 Convolutional neural networks (CNN)

Convolutional Neural Networks (CNN, for short), also called ConvNets, are developments that are responsible for a big jump in the ability to recognize objects by deep learning methods. Figure 5 shows a vivid illustration of the basic architecture of convolutional neural network (CNN). There are two main parts to a CNN architecture, namely the feature extraction part and the classification part, also present in the traditional detectors. The feature extraction part has the input layer, the convolution layers, and the pooling layers, while the classification part includes the fully connected layers and the output layer. The input layer takes the input image in the form of pixels and passes it to the convolution layers for processing. Here, filters called kernels are used on a small region of the image to produce feature maps, highlighting specific patterns or features like

edges, textures, or shapes. of the images. As the layers get deeper, the network learns different levels of image features.

The network uses activation functions to add non-linearity to the network allowing it to learn complex relationships between features. The next layer, the pooling layer, is used to reduce redundancy in the feature maps. The fully connected layers process these features and connect them together in layers for the network to learn complex relationships between high-level features. The last layer, the output layer, makes the final prediction or classification based on the learned features using the necessary activation function depending on the task.

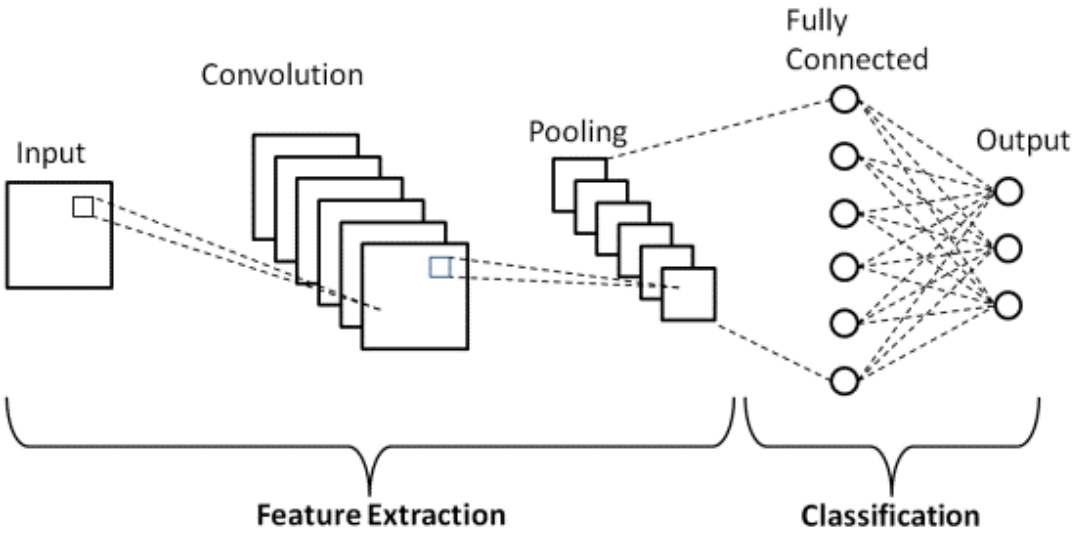

Figure 5. Basic architecture of CNN (Researchgate net, 2019).

What makes CNN powerful is its architecture which can vary in depth, number of layers, filter sizes, and configurations. This makes it possible for it to be used for different computer vision tasks such as object detection, image classification, and segmentation. By having different architectural configurations, the networks can learn patterns from images and perform various tasks. Popular CNN architectures include VGG, ResNet, AlexNet, and Inception. These have their different strengths and have been used as backbone

algorithms when implementing COTS detectors. Let us talk about the recent, state-of-the-art techniques of CNN that have championed real-time COTS detection?

2.4.2 State-of-the-art real-time detection techniques

Why is it necessary to consider real-time object detection using CNN methods? This is because the problem that this project work is set out to solve will involve the detection of COTS in real-time image (or video) data. An efficient real-time underwater target detection algorithm is very important. Does CNN provide algorithms that support real-time object detection? Yes. Over the years, many algorithms or techniques based on CNN have been developed and used, with the proceeding ones offering some improvements over the preceding methods and technologies. In the section convolutional neural networks methods, explanation was given on the CNN-based method detectors, two-stage and one stage detectors. They have contributed to real-time object detection because they promote speed and accuracy of detection.

Two-Stage Detectors

The earliest two-stage detector was **Region-Based Convolutional Neural Network (R-CNN)**. This detector was based on a two-stage technique. In stage 1, the selective search algorithm generates the region proposals from the input images. In stage 2, the region proposals generated are passed to the CNN network to extract the features from the input image and to pass these to a support vector machine (SVM) to perform classification and bounding box regression. This is a major drawback for R-CNN, making it to be slow in speed, though more accurate, 66.0% of Mean Average Precision (mAP) (Patel, S. et al., 2021). The **Fast R-CNN** was released to overcome this shortcoming of R-CNN and improve on the speed and accuracy. It does this by using the concepts of region of interest (ROI) that reduced the time consumption, unlike the R-CNN. It uses CNN to process the input image and then max pool the features for each region proposal. This reduces computation cost and time because it reduces the redundant region proposals that add computation cost and time. Again, instead of using the support vector machine (SVM) to classify each region proposal as in R-CNN, Fast R-CNN uses softmax layer that is trained

jointly with the CNN. This adds improvement in classification accuracy and removes the need for a separate SVM. Fast R-CNN achieved a mAP of 66.9% A stretch toward real-time detection.

Faster R-CNN was later introduced and used the concept of a region proposal network (RPN) to replace selective search algorithm. It integrates the RPN and the Fast R-CNN into a single network to handle the region proposals. In other words, it is composed of two modules. The first module is a deep CNN fully connected that generates the regions, and the second module is the Fast R-CNN that handles that processes the proposed regions. This improvement achieved a 69.9% mAP for the Faster R-CNN.

One-Stage Detectors

You Only Look Once (YOLO) falls under a group of detectors referred to as one-stage detectors widely used for real-time object detection tasks. It is known for its speed and accuracy in detecting and localizing multiple objects in an image or video. Unlike in the two-stage detectors where there are two stages to detection, YOLO is a one-stage network whereby the generation of the bounding boxes and class prediction are handled in one single evaluation or in a single pass. This accounts for its speed, which is faster than the two-stage detectors based on region proposals. It can process up to 45 frames per second (FPS). This is a good candidate for real-time object detection. Another advantage is that since it uses fully connected CNN, it learns features from input images and detect objects of various shapes and sizes without relying on predefined classes (Toan, N., 2022). In addition, it solves the object detection problem as a regression by handling the bounding boxes and class probabilities at the same time. This simplifies the network architecture and reduces the number of parameters and as a result improves the speed of the networks. This will be good for our COTS detector because we need a detector that can detect (identify and localize COTS) in real-time.

Single-Shot Detector (SSD) is one of the detectors categorized as one-stage. It is a state-of-the-art real-time object detection algorithm that provides better speeds compared to

Faster R-CNN. It takes only one single shot to generate regions of interests (ROIs) or region proposals, and at the same time use CNNs to classify the regions; unlike the two-stage detectors that handle these in two separate stages (Asad, H. et al., 2020). SSD uses the VGG-16 model pre-trained on ImageNet dataset as the base model (Patel, S. et al., 2021). Additionally, at the end of the base model are additional convolutional layers used for object detection.

Zero-Shot Object Detection

Zero-shot object detection has to with the ability of a model to detect objects that it has not been explicitly trained on. In the object detection methods thus discussed, the models are trained on specific classes of the dataset, and their performance is limited to recognizing those classes. However, in the case of zero-shot object detection, models extend their capability by recognizing classes in images based on free-text queries. The methods used in this category include OWL-ViT, which is an open-vocabulary object detector that can detect objects images based on free-texts; GLIP, adds word-level understanding to find objects by the semantics; and Segment Anything, to add masks to see the pixel-level location of the objects. These methods help to enable us to detect objects in images without training the neural network extensively. This opens new possibilities in computer vision field.

2.4.3 Enhancing COTS detection in underwater environments

There is no algorithm that will mitigate all the challenges affecting the detection of COTS in underwater environments. However, there are many techniques that have been developed by computer vision experts which can help to address COTS detection challenges. These techniques can be integrated for better detection. This section looks at some of these techniques, with particular emphasis on techniques like transfer learning, data augmentation, and enhancing image preprocessing techniques. Why these three techniques are vital is because they will help us answer our research questions. Firstly, whether transfer learning can effectively improve the model, and secondly, whether data augmentation and image preprocessing of underwater images can improve the visibility

and distinguishability of COTS from the background environment? Later, when we design and implement our COTS detector model, we will see if these have effects on the detection of COTS.

Image enhancement and preprocessing

Underwater images are often affected by varying lighting conditions, water turbidity, haze, blur, and color deterioration that are obstacles to COTS detection. There are various image enhancement or preprocessing techniques that can improve the quality of images for better detection (Saleh, A. et al., 2022). These algorithms include contrast enhancements, dehazing, and other algorithms that can perform image enhancements and recover some information from the poor-quality images. For example, contrast enhancement algorithms such as histogram equalization, adaptive histogram equalization, or contrast stretching can improve the visibility of COTS images when the images have lighting issues; dehazing algorithms restore contrast to underwater images affected by water turbidity and haze issues. Saleh, A. et al. (2016) pointed out that even using the basic image enhancement techniques has improved image quality and continued that some recent studies have improved the image quality by just using low-quality images and deep learning methods.

Sahu, P. et al. (2014) wrote about improving image quality by using Gabor filter. The improved images were then fed to an edge detector to extract features from underwater images, and another that used a color enhancement method to improve the color contrasts of underwater images. Another study was conducted where a de-hazing algorithm was used to improve hazed underwater images, restoring the attenuated data. Jian, M. et al., (2020), in their paper wrote about studies in which various methods were used to enhance underwater images, to reduce noise, to remove fog from images, to correct colors and for image recovery. Certainly, this thesis work will employ some image enhancing algorithms to improve image quality for better COTS detection.

Data augmentation and synthesis

Data augmentation mitigates the effects of having a small dataset and replicate external environmental conditions such as variable illumination, fluctuating contrast, and blurring (Samantaray, A. et al., 2018). Data augmentation applies transformations such as changing the brightness, contrast, hue, colour, saturation, flipping, rotation, mirror, scale, crop, and warp to the dataset Samantaray, A. et al., 2018; Wang, N. et al., 2022). Data augmentation mitigates against overfitting by synthetically producing new data samples. Deep learning thrives on large dataset. So, by synthetically making more data available, data augmentation makes deep learning model effective by preventing overfitting. This technique is used to improve the detectability of COTS by the detector (Mees, O. et al., 2019). Later, when we implement the COTS detector, we will see if the result supports this. In YOLOv8, mosaic augmentation on the training data is used. Mosaic augmentation is a type of data augmentation technique that accepts four random images from the training set and combines them into a single mosaic image (Reis, D., 2023). The resultant image, which contains a random crop from one of the four input images, is then used as an input image for the model.

Utilizing deep learning architectures

Deep learning architectures based on CNN have shown remarkable success in computer vision tasks. These architectures can learn from COTS complex features from underwater images, and this helps to improve the detection accuracy of COTS. As deeper they get, they are better at learning complex features from data, including underwater data (Wang, N. et al., 2022). Deep learning models such as Faster R-CNN, Single Shot Detector (SSD), and Region-based Fully Convolutional Networks (R-FCN) have shown near real-time object detection and high accuracy (Samantaray, A. et al., 2018). In addition to CNN-based networks, other networks like Siamese Networks learn to differentiate between similar and dissimilar images. This can be used for detecting subtle differences between COTS and the reefs, and between COTS and the background environments. Also, integrating attention mechanisms like transformer-based architectures or spatial attention has the

advantage of focusing the model on regions of interest in underwater images containing COTS (Khan, M. et al., 2023; Wu, T. & Dong, Y. 2023).

Transfer learning

Transfer learning is another great method in computer vision object detection that enhances performance. It involves using a pre-trained model as a starting point and subsequently fine-tune it for a specific object detection task. Transfer learning is desirable because it speeds up the training and improves the performance of a model, particularly when the data is limited or small. Pre-trained datasets such as MS-COCO, Darknets, ImageNet, and VGG have thousands of images, and during training the weights and biases learned from these datasets are transferred to the trained model. These are low-level weights and features that are common and transferrable from the pre-training dataset (Reis, D. et al., 2023; Gupta, A. et al., 2021). Ahmed, M. et al. (2023) described a study that used transfer learning in the detection of objects under low illumination (Sadagawa, Y. & Nagahara, H., 2020). Transfer learning has remarkably improved the performance of models. In this thesis work, we will find out if transfer learning will improve the visibility of COTS for better detectability.

Collaboration with data sharing

Right now, the CSIRO COTS detection dataset is the most popular dataset for COTS made public for educational purposes. It will be good if researchers and organizations cooperate and collaborate in creating larger, more diverse, and fully annotated COTS datasets. This dataset can be captured from different sources, such as remote sensing devices. These datasets that will be representative of the variations in underwater environments can be shared to promote education and research in COTS detection. By having COTS data available from different sources and representative of real-life situations, COTS detection can be improved (Samantaray, A. et al. 2018).

3 Methodology

We looked at various methods that have been used to detect the crown-of-thorns starfish and other marine life at large over the years in the previous chapter, and how the application of the cutting-edge deep learning techniques in object detection can provide a real-time and a more accurate detection alternative than the traditional or current methodologies. This chapter focuses on the research design or the design methodology for the study, preparing the way for the next chapter, the actual implementation of a deep learning model to detect COTS. The study will take the quantitative deductive approach as illustrated in the research onion model. The methods will be quantitative and experimental because this will be suitable for our research questions where we test our two hypotheses by performing experiments:

- H_0: Does transfer learning from pretrained models effectively improve the generalization of COTS detection models across varied and underwater environments?
- H_1: Do data augmentation and preprocessing methods improve the visibility and distinguishability of COTS from the reef background in underwater images and compensate for the class imbalance?

We have seen the reasons in favor of the argument for deep learning object detection techniques over the traditional or the current methodologies. By testing our hypotheses by means of experiments, we can be confident whether there are relationships between transfer learning, data augmentation, and data preprocessing and a better and accurate COTS detection. The first to discuss will be the data and collection methods. This is an important factor because a good understanding of our data will impact on the design of the methodology. Next, the discussion of the deep learning framework and architecture for this work and the justification for this architecture. Then, the metrics employed to assess the performance of the model - COTS detector.

3.1 The CSIRO dataset

This study uses secondary data. Our underwater dataset is called CSIRO dataset, released to the public for educational purposes. The dataset is for starfish detection, a large-scale annotated underwater image dataset from the Great Barrier Reef (GBR) to encourage research work on Machine Learning and AI-driven technologies to help find a solution to the crown-of-thorn starfish population outbreaks. The dataset is hosted also for a Kaggle competition in Machine Learning which is challenging the machine Learning community for a deep learning computer-based detection. CSIRO (Commonwealth Scientific and Industrial Research Organization) is an Australian Government agency that is responsible for scientific research. It teamed up with the Great Barrier Reef Foundation (GBRF) to sponsor the work for the dataset collection by a group of researchers, as mentioned in their paper (Liu, J. et al., 2021). CSIRO is working with industry, government, universities, and research organizations in many projects to bring solutions for food security and quality, clean energy and resources, health and wellbeing, resilient valuable environments, and innovative industries for Australia and its regions.

3.1.1 Data collection method

The dataset was collected using the GoPro Hero9 cameras attached to the bottom of Manta Tow board used by a trained observer or snorkeler-diver. The camera provides a wide range of views below the board and the reef under. The distance between the board and the reef can vary, but the speed is maintained constant. This makes it possible to cover 200 meters in two minutes, after which the diver stops and records data observed during the transect on a sheet of paper. The camera has a resolution of 3840x2160 and records videos at the rate of 24 frames per second. The data is further manipulated for AI-assisted annotations and quality assurance by annotation experts. With pre-trained COTS detection models, the COTS in the images are identified and marked by bounding boxes using annotation software. The data was collected in a single day October 2021. It must be noted that the images show variations in the lighting, visibility, coral habitat, depth, distance from the bottom of the manta tow board and

viewpoint (Liu, J. et al., 2021). Over 34k of these images were released for educational purposes, and this would be the dataset for this project.

In what ways is CSIRO dataset different from other popular conventional object detection datasets, such is MS-COCO, IMAGENET, PASCAL VOC, and others? Liu, J. et al., (2021) gave four reasons. First, there is only one class (the COTS class), for the CSIRO dataset, whereas other datasets contain many classes. For example, IMAGENET has almost twenty-two thousand (22k) classes, and MS-COCO has up to 80 classes. Second, the dataset naturally exhibits sequence-based annotations as multiple images are taken of the sane COTS as the boat moves past it. Third, a picture could continue one or many images of the COTS, and these could possibly overlap with each other. Fourth, the object of this dataset is to give the images of the COTS in a defined transect (Liu, J. et al., 2021).

3.1.2 Exploratory data analysis (EDA) of the dataset

The dataset images, when downloaded, were organized in a folder called [train images] that contained three folders labelled video_0, video_1, and video_2. These sub-folders contained underwater data images of the reef. These images were parts of videos taken by the manta row diver but were cut into frames of images as jpg files. There was a total of 23,501 files in the three folders. The image in figure 6 shows the distributions of these images in three folders. The folders have an unequal number of images. These folders represent different sections or regions of the reefs. It is good to have images from different sections for generalization. The train.csv and test.csv files provided some explanations for the training and test sets respectively. The train csv file also contained important information about annotations of COTS in the image files. The annotations showed the locations of COTS in the images.

Out of 23,501 images in the dataset, 4,919 had annotations while 18,582 did not. Well, since deep learning model needed a lot of data to train, we would use data augmentation to augment this dataset. Certainly, the number of images to train our model would increase by the time we perform data augmentation, as we will see later. We focused on

the images with annotations to train our model because they contained COTS images that would be used to train the model, to show the model what the COTS looked like, even though we would add some percentage of the images without annotations as background images. The reason will be explained later. An image can have one or multiple annotations. There was a total of 11,898 annotations in these 4,919 images. The computer vision model that would be used for the project will be a supervised learning algorithm and will need the labeling of the images. Happily, the labeling has been done already (annotations column in the train.csv file) and provided with the dataset. A close look at the annotation column shows that some images have one COTS image, while some others have multiple COTS images.

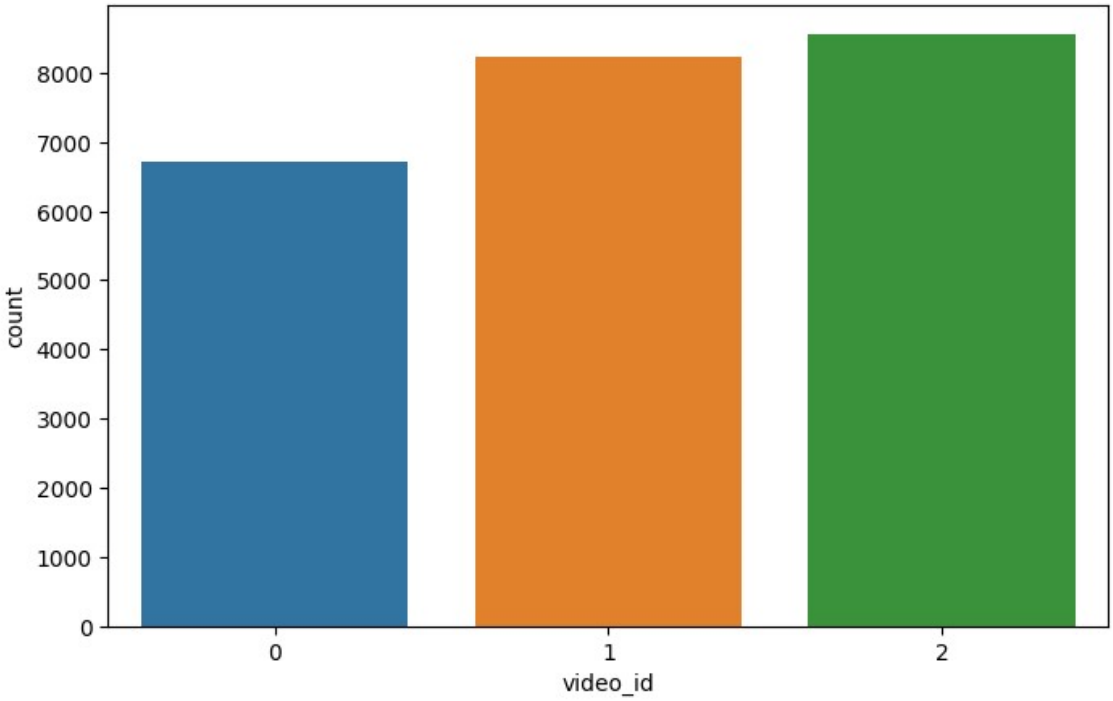

Figure 6. The distribution of images in three folders.

3.1.3 Data annotation and normalization

Now that we have the images from the dataset, the next step is to annotate these images. Annotation is the coordinates of bounding boxes to locate identified objects of a class in an image. The model uses the annotations to locate and draw bounding boxes on each

COTS object in an image. Object detection is a supervised learning because the objects in the images are supposed labelled for the model. All the objects we want the model to detect need to be annotated for the model. Annotating images is a tedious work, even the most tedious in modelling a deep learning computer vision detector. Gratefully, the CSIRO dataset came annotated by the publishers of the CSIRO dataset before releasing it to the public. This takes a lot of loads off the users of this dataset since they do not have to worry about using annotation software to perform own annotations. Besides, these annotations were done using AI-assisted and quality assurance process by expert annotators with the help of pre-trained COTS detection models (Liu, J. et al., 2021). This helped in the identification and location of COTS in the images. This information is contained in the train.csv file. The following image in figure 7 shows what it looks like:

image_id	video_id	sequence	video_frame	sequence_frame	annotations	no_annot
0-16	0	40258	16	16	[{'x': 559, 'y': 213, 'width': 50, 'height': 32}]	1
0-17	0	40258	17	17	[{'x': 558, 'y': 213, 'width': 50, 'height': 32}]	1
0-18	0	40258	18	18	[{'x': 557, 'y': 213, 'width': 50, 'height': 32}]	1
0-19	0	40258	19	19	[{'x': 556, 'y': 214, 'width': 50, 'height': 32}]	1
0-20	0	40258	20	20	[{'x': 555, 'y': 214, 'width': 50, 'height': 32}]	1

Figure 7. Sample of train.csv data frame showing annotations.

From the image in figure 7, we have ‘x’, ‘y’, ‘width’ and ‘height’. These are the x and y coordinates of the annotations for the bounding boxes for the starfish in the image, and the width and height of the bounding boxes. In other words, these are the (x_min, y_min) of the upper left corner of the bounding box within the image together with its width and height in pixels. We are thankful that the images were annotated for us by the publishers of the dataset. The figure shows additional information such as the number of annotations per image, image id and the sequence of each image in the video. However, to pass this annotation information for each image to our model, we must convert it from the following format:

[center_x, center_y, width, height]

Starfish 0.45625 0.31805555555555554 0.0390625 0.044444444444444446

This is called normalization, and we use the following formulas to do that.
our model does not understand this annotated format. The annotations must be normalized with respect to the width and height of the images instead of the bounding boxes.

$$x\_max = x\_\min + width \tag{1}$$

$$y\_max = y\_\min + height \tag{2}$$

$$center\_x = \frac{\frac{x\_\min + x\_max}{2}}{\text{width of the image}} \tag{3}$$

$$center\_y = \frac{\frac{y\_\min + y\_max}{2}}{\text{height of the image}} \tag{4}$$

$$width = \frac{x\_\max - x\_min}{\text{width of the image}} \tag{5}$$

$$height = \frac{y\_\max - y\_min}{\text{height of the image}} \tag{6}$$

Using Python script and the above formulas, the annotations from the dataset could be converted into text files, with the corresponding image filename as the name of the text files and the annotation coordinates into model's annotation formats. The following sample data below shows the coordinates for image 0-16.jpg stored in 0-16.txt file.

Starfish 0.45625 0.31805555555555554 0.0390625 0.044444444444444446

The labeling information above can be used to draw bounding boxes on the corresponding image. The following images of figures 8, 9, and 10 show samples printed from the

coordinates from their corresponding text files, which contain information about the bounding boxes for the COTS. The publishers of this dataset performed annotations on the images and included these annotations.

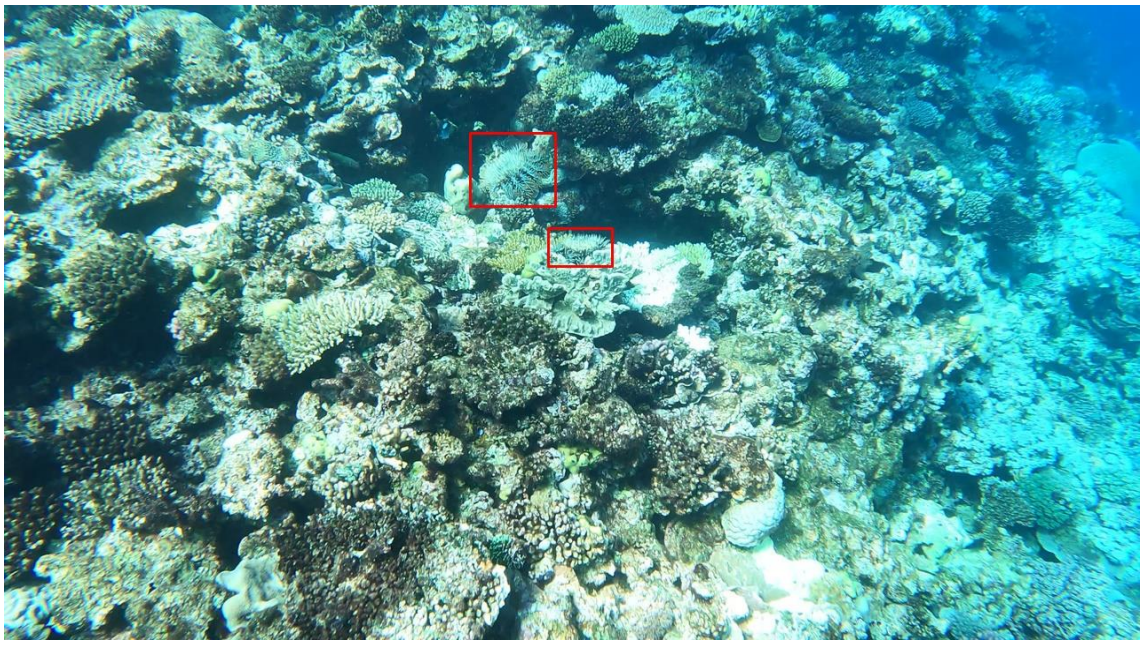

Figure 8. COTS **marked in red bounding boxes using the coordinates for image 0-45.jpg.**

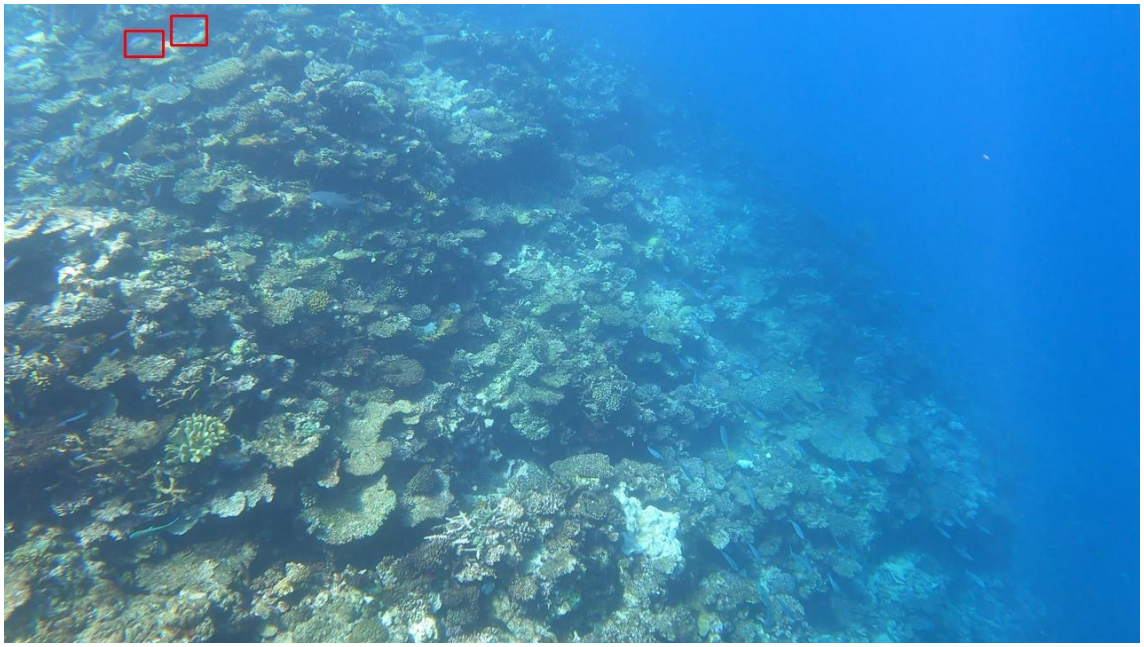

Figure 9. COTS marked in red bounding boxes using the coordinates for image 0-4538.jpg.

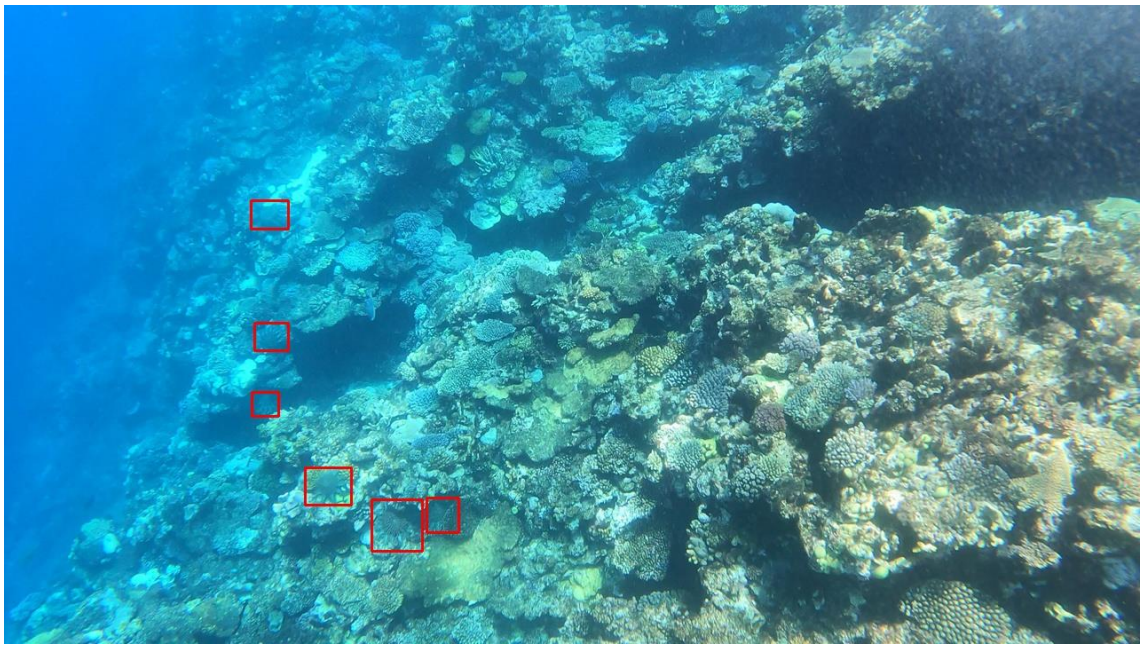

Figure 10. COTS marked in red bounding boxes using the coordinates for image 1-99.jpg.

3.1.4 Data splitting

The three image folders were all combined and the images separated into two classes: annotated and non-annotated image files. There were 4,919 images with annotations, and 18,582 without annotations. The annotated images were split into training 87%, validation 8%, and test 5% into their respective folders. The annotated images were shuffled properly before splitting. This made it possible to have well shuffled and well representative images from the previous three folders representing different underwater conditions where the images were taken. In addition, 10% of non-annotated images were chosen and added to the training set to serve as background images. But suffice it to mention that background images are images without COTS objects. They are added to the training set to improve performance during training. It will be good for the model to learn to distinguish the COTS from the background reef environments. More on background images in the next chapter on implementation.

It is a good practice in machine learning to split the dataset into train, validation, and test sets. The training set is used for training the model, validation set is used to evaluate the model after training with a new set of data, and the test set is used for actual testing of the model to see how it has performed on new and unseen data. The reason for doing this is to ensure that there is no overfitting. Overfitting is a situation that the model learns the train set so well but does not generalize well on the unseen data (test set). So, the practice demands that after training, validation and testing are performed using new data, that is the unseen data, that reflects the real-world data. That gives an idea of how well the model has learned by evaluating on new data.

3.1.5 Data preprocessing and augmentation

Image preprocessing involves resizing and normalizing of images (data) used for training, validating, and testing. In addition to image normalization discussed in the last two sections, resizing images to a uniform size ensures consistency and facilitates model training (Bahhar, C. et al., 2023; Ang, G. et al. 2023). The images used for modelling will be resized to 640 x 640 dimensions. To encourage efficient model performance, the images used in this work were all resized to 640x640 pixels and normalized as explained earlier (Pratama, Y. et al., 2021). During implementation, the data preprocessing pipeline will take care of resizing the images to the chosen sizes. The resized image data is normalized to a specific range, often [0, 1] or [-1, 1]. We have seen the formula used for normalizing the images in the last two sections, converting the images to the YOLO-required format. Then, the normalized image data is converted to a tensor as input to the model. This step is vital because it ensures that the model converges more efficiently during training.

The data processing pipeline will also include the preprocessing of the image images. These images are underwater images and are affected by certain underwater conditions such turbidity, lighting, and noise. They need to be processed for better image quality, which will in turn impact on the performance of the model. For this work, a python script was developed that used OpenCV to apply image enhancement and dehazing functionalities to the underwater images. It handles one image at a time. These functionalities

first convert images to grayscale and apply Dark Channel Prior (DCP) and Contrast Limited Adaptive Histogram Equalization (CLAHE) functions to the images. DCP is mainly used for image dehazing and CLAHE is used to enhance image contrasts while removing noise from low contrast portions. These methods are applicable to enhance underwater images and improving the quality. The enhanced images are then used for training the COT-detection model (Mousa, A., 2023; Wang, X. et al., 2022; Kaushik, S. & Vigneshwaran, P., 2022).

3.2 Deep learning framework

We have looked at object detection using deep learning methods. Convolutional neural network (CNN) has revolutionized the field of computer vision and has achieved impressive results for detection of objects in digital images (Patel, S. & Patel, A., 2021). A family of CNN-based models is YOLO. I have chosen as the base model for this thesis work. YOLO (You Only Look Once) is an object detection algorithm that uses a single neural network to predict bounding boxes along with class probabilities for each of the object in an image at the same time. The latest version is version 8 (YOLOv8). It performs image segmentation in addition to object detection with speed and accuracy (Viswanatha, V. et al., 2022). This makes YOLOv8 a good choice for real-time object detection.

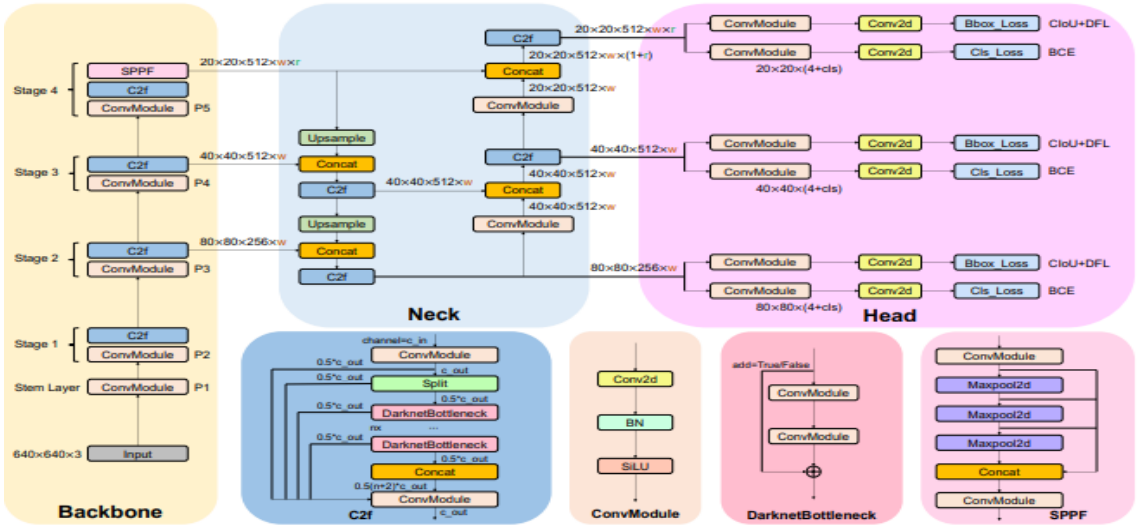

Figure 11. YOLOv8n (nano) architecture (Ju, R. & Cai, W., 2023).

Figure 11 shows the architecture of YOLOv8 model. Yolov8 architecture is divided into three components. The first component, the backbone network, uses the modified version of CSPDarknet53 or ResNet-50 deep convolutional neural network (CNN) to extract features from the input image (Ang, G. et al., 2023). This CNN is trained on a large dataset such as COCO and ImageNet to recognize high-level features in input images. The second component, the neck network, connects the backbone and the third, the head, and works to balance accuracy and speed. This accounts for its speed and efficiency. The third component, the head network, predicts the class and bounding boxes of objects in the image. It optimizes the Intersect over Union (IoU) and the non-maximum suppression (NMS) algorithm.

There are many reasons why YOLOv8 is a choice for this thesis work. They include according to these authors (Wu, T. and Dong, Y., 2023; Ma, M & Pang, H., 2023):

- **Improved Accuracy:** YOLOv8 is the latest version of the YOLO family. This latest version has built on the strengths of the previous versions. So, the latest version, YOLOv8, has the most improved accuracy because of new techniques and optimizations.
- **Enhanced Speed:** YOLOv8 has the best inference speed with high accuracy compared to other object detection models.
- **Multiple Backbones:** YOLOv8 has support for various backbones, such EfficientNet, ResNet, and CSPDarknet. Depending on the use case, one has the flexibility to choose from these to achieve the best result.
- **Advanced-Data Augmentation:** With YOLOv8 comes with advanced data augmentation methods such as MixUp and CutMix to improve the robustness and generalization of the model. There is the flexibility to include external data augmentation like Albumentations to improve the model.
- **Customizable Architecture:** YOLOv8 has a customizable architecture that allows users to modify the structure and parameters of the model to have a custom-made or tailor-made model.
- **Pre-trained Model:** Transfer learning is possible with YOLOv8, enabling users to use pre-trained models which improve the generalization of the mode. For example,

YOLOv8 can be pre-trained on COCO dataset which improves the model by transferring learned weights.

- **Adaptive Training:** By means of the adaptive training, the learning rate and balance of the loss function can be optimized which leads to improved model performance. This is particularly helpful since our data is underwater images (Liu, W. et al., 2022).

YOLOv8 comes in five different flavour models. They are as follow:
YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large). Their speeds and accuracies vary, and the platform usage can determine which to choose from. YOLOv8n is the fastest but the least accurate, while YOLOv8x is the slowest but the most accurate. I have chosen YOLOv8m and YOLOv8l as the base models for training. One reason for this choice is the computation power of the computer to be used.

3.3 Evaluation metrics

The performance of an object detection model can be measured using some metrics, and these metrics are also used in other fields to evaluate performance. These include Accuracy, Precision, IoU, Recall, PR curve, Average Precision, and others (Kaur, R. et al., 2022). For our dataset, the metrics that are used include Precision, Recall, F1-Score (based on confusion matrix) and Intersect over Union (IoU). A confusion matrix is a matrix that gives the summary of the performance of a machine learning model based on a set of test data. Table 1 illustrates the point. We can see that it is based purely on True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). True Positive is when the model predicts that it is a COTS, and it is. True Negative is when the model predicts that the object is not COTS, and it is correct. In False Positive, the model predicts that there is COTS image when there is not. The same goes with False Negative, when the model says it is not a COTS image when there is, the model gets it wrong. In fact, there is a COTS starfish image. By taking into consideration the positives and the negatives, it gives a more representative view of the accuracy of the performance than

just only the simple accuracy. Now, we will talk about the different metrics for a computer object detection model or algorithm.

	Predicted Class		
True Class		N	P
	N	TN	FP
	p	FN	TP

Table 1 The confusion matrix.

They are as follows:

- **Precision**: This is a measure of correctness that specifies the number of true positives over the total positive predictions (both true positives and false positives). This gives a good idea of correctness because unlike accuracy which tells the number of total positive predictions, precision gives an idea of the number of correct predictions and the number of incorrect predictions.

$$Precision = \frac{TP}{\mathrm{TP+FP}} \tag{7}$$

- **Recall**: Recall is another powerful measure of correctness that gives the idea of the number of true positives over the total number of true positive and false negative. Many call it sensitivity. It captures as many positives as possible.

$$Recall = \frac{TP}{\mathrm{TP+FN}} \tag{8}$$

- **F-measure/F1-Score:** The value for F1 falls between 0 and 1 and it is the harmonic mean of precision and recall. Unlike averages that are sensitive to extremely large values, F1-Score is not.

$$F1 - Score = 2 * \frac{(Recall * Precision)}{(\text{Recall} + \text{Precision})} \quad (9)$$

- **Intersection over Union (IoU):** Object detection is a prediction of the presence of an object and the location of the object in the image. This is accomplished by drawing a bounding box around the object. A good measure of the correctness of a model is the area of intersection of the bounding boxes of the real and predicted over the area of the union of the two boxes. The higher the IoU the better the performance of the object detection model. The value is 1 if the area of the predicted bounding box and the ground truth overlap perfectly, and 0 if they do not intersect each other at all. When IoU value exceeds the predefined threshold, it means that the object is properly recognized.

$$IoU = \frac{Area\ of\ Intersection\ of\ two\ boxes}{\text{Area of Union of two boxes}} \quad (10)$$

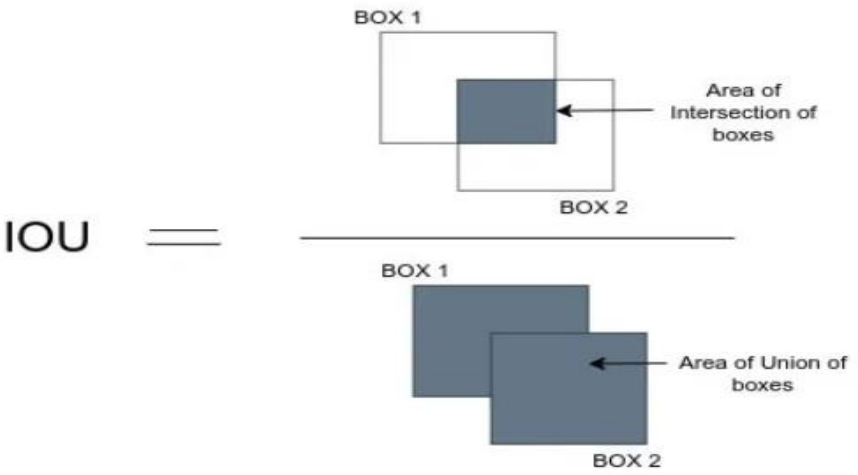

Figure 12. Pictorial representation of IoU.

- **Mean Average Precision (mAP):** This is an object detection performance metric that measures or assesses the quality of an object detection model by

considering both precision and recall at various confidence thresholds. It is useful when dealing with tasks that require ranking and localizing objects in images. A value of close to 1 means that the model is good at accurately detecting and ranking objects across different classes and confidence thresholds. Conversely, a value close to 0 means that the model is not accurate in identifying and localizing objects in various categories and different confidence levels. In this COTS detector, we are aiming at a high value of mean average precision which shows that the model is reliable in detecting and localizing COTS.

3.4 Efficiency improvement methods

Some of these methods reduce the losses in the model during training. Others transfer weights and biases learned from pre-trained model to the new model. While others augment the training set. These are discussed in the following sub-sections.

3.4.1 Loss function

The loss functions are functions that try to minimize losses in a model between the ground truth and the predicted. The aim is to continuously reduce the following losses. The graphs will display these losses after training.

- Objectness Loss (Confidence loss): This loss is a measure of how well the model predicts or estimates the confidence score of the object's presence in a grid cell. The model uses binary cross-entropy (logistic regression) to calculate this loss. The value of this loss is between 0 and 1.
- Classification Loss: Classification loss measures the error in predicting the correct class labels for the detected object. It uses categorical cross-entropy (SoftMax) to calculate this loss.
- Bounding box Location: Bounding box location loss measures the error in predicting the coordinates accurately. It uses mean squared error (MSE). Reducing this error means that the model correctly predicts the bounding boxes coordinates.

3.4.2 Non-maximum suppression (NMS)

This is a powerful technique used in object detection tasks to optimize bounding boxes. It removes duplicates or highly overlapping bounding boxes, making sure that only the most confident and accurate predictions are retained. When a model makes predictions, there are a lot of bounding boxes predicted; some overlap each other, and others have low confidence value. Non-maximum suppression comes to the aid by smoothening out the overlaps and reducing the number of redundancies and improving the precision of the object detection results. Reis, D., (2023) describes it as a 'filter' that filters out overlapping bounding boxes. This technique can be integrated into YOLO. Non-maximum suppression accomplishes this using a few techniques, such as confidence thresholding and iterative suppression.

In this chapter, I looked at some of the methodologies used in the design of the COTS detection model. Our design methodologies have a bearing to help us answer our research questions. We include methodologies such as transfer learning, data augmentations, and preprocessing our image data to improve the picture quality so that we can improve the COTS detection accuracy. In the next chapter, we look at the actual implementation or training of the model to test the hypotheses.

4 Experiments and results

This chapter focuses on the actual implementation of the computer vision COTS-detector using the state-of-the-art algorithm, YOLOv8. YOLO has lately been gaining momentum in real-time object detection, classification, and segmentation, for its speed and accuracy. It has also outperformed other earlier computer vision algorithms. In the last chapter, we designed the methodologies that will be used in this chapter for the actual implementation. First, the training of the COTS detector to learn patterns from the dataset and be able to detect COTS. The implementation will explain the necessary procedures, the pipeline, the hardware, and the software required for successful training of the model. And second, the results from the training are important because they will help us answer the research questions. We want to see if there is a connection between the transfer learning of the pre-trained models and the model's generalization. Also, we want to find out if data augmentation and image preprocessing can improve the COTS' detection accuracy by the model. First, we start with the setup for the experiment, both software and hardware. Then, we discuss the training of the model; first, the base model and subsequent models training with transfer learning, data augmentation, and data processing.

4.1 Experimental setup

YOLOv8 makes it easy to train our model on either a CPU or a GPU (Graphical Processing Unit), but preferably on a GPU. It will take ages to train on a CPU because it is computation-intensive, and faster on a GPU. The GPU takes over the huge image processing and computation involved as lots of images (thousands) are involved. My system (local machine) met the minimum hardware requirements for this, although it was not the best GPU available. It was an MSI Intel(R) Core(TM) i7-11800H @ 2.30GHz, 16.0 GB RAM, 134.8/931.5 GB disk, Windows 11, NVIDIA GeForce RTX 3060 GPU, 6144MiB system. To install and run YOLOv8, this was the list of software requirements that should be installed in addition to torch-2.0.1+cu117 CUDA driver. Appendix 1 shows the software and libraries.

I cloned YOLOv8 from Ultralytics GitHub repository to my local machine and installed all the libraries and dependencies for YOLv8 (Ultralytics is the organization that developed YOLOv8). I also installed Python and Python libraries such as Matplotlib, NumPy, Pandas, OpenCV, PyTorch and TensorFlow libraries. These libraries are image manipulation libraries and were vital to preprocess the image data. The training, validation and test images and annotation files were all organized strictly for YOLOv8, and all the information is specified for YOLO in a configuration file, '.yaml' extension. YOLO uses this configuration file to load the folders housing the train, validation, and test sets. The path to the configuration file is assigned to the data parameter of the train method for YOLO object.

4.2 Model training

To see the effects of using a pretrained model, data augmentation, and data preprocessing on our YOLOv8 model, I used YOLOv8m (medium) as a base model. I performed a series of training experiments. The first experiment was training the YOLOv8 model from scratch without any pretraining. This allowed the model to learn from scratch without transferring any weights and biases learned from any dataset. This was trained for 100 epochs (epoch is one round through the training set). Next, a pretrained YOLOv8m model based on COCO dataset was used this time, which means that the weights and biases learned from COCO dataset was transferred to this model. In addition, there was no data augmentation nor preprocessed training set used. This training was performed for 100 epochs. The third was a training using COCO-based dataset of YOLOv8m model and data augmentation for 100 epochs. The data augmentations applied included geometric parameters such as rotation, scaling, translation, flipping, and shearing. Included also were the color-space augmentation parameters, such hue and saturation. The next two trainings utilized pretraining, no data augmentation but instead, preprocessed training set was used for 100 epochs, and training set comprising of both original and pretrained sets for 100 epochs.

In the second round, the training was conducted using YOLOv8l model. The training was performed using the same steps outlined in the previous round. The only difference was that YOLOv8l was used in place of YOLOv8m and for 150 epochs. The results were recorded and displayed in the following sections, and in the next chapter, the analysis in the results. The best performing model was trained later for an epoch of 200 to obtain better performance and model convergence.

4.3 Results of the experiments

The experiments of this thesis were done not all at once but incrementally to see the effects of some concepts and parameters on the model training and, hence, their effects on the performance of the model were recorded. Here, it is good to highlight the results obtained from our experiments and see how the training fared. If the results are good, then we see the reasons to deploy it. If not so good, it will be tweaked a little for better performance. First, let us talk about the loss and the precision of the model.

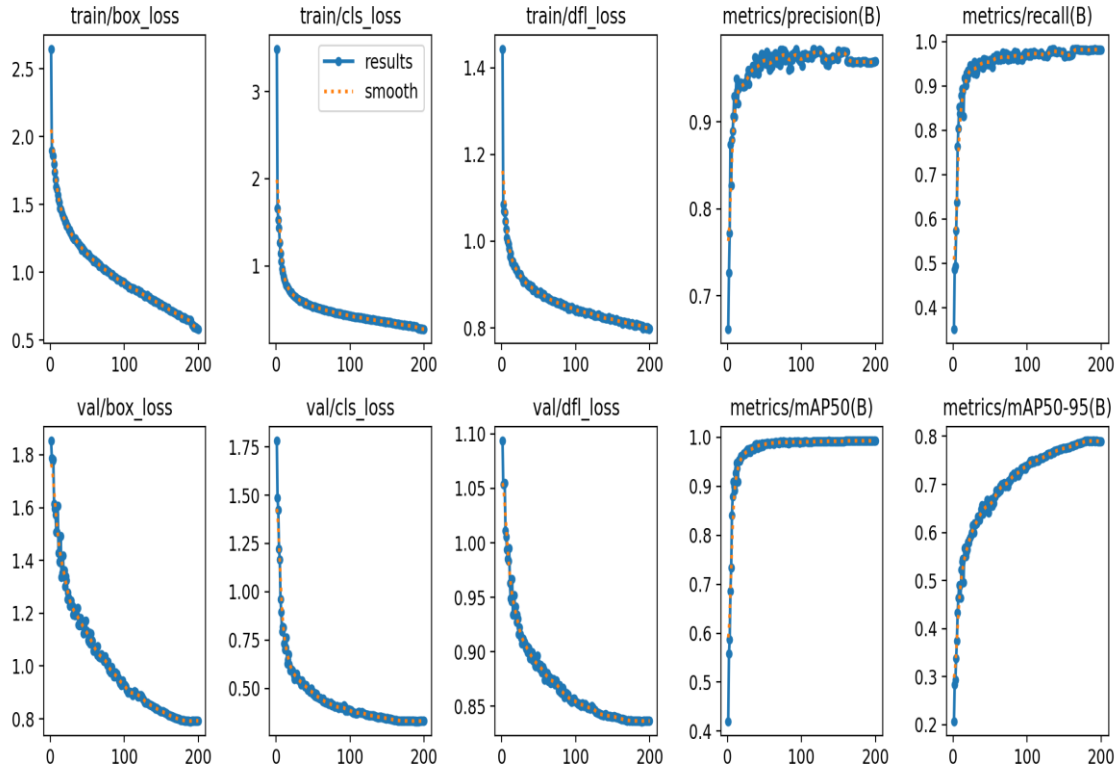

Figure 13. Training losses, training precisions and recalls for 200 epochs.

Figure 13 shows the losses, the precisions, and the recalls for the model training. The training losses are going down as expected, and the precisions and the recalls are going up. This shows that the model was really learning from the training set during training that stopped at 200 epochs. This is an indication that the model's overall performance was good, and it can be further trained. Next, the F1-Confidence curve. This is shown in the next figure, figure 14.

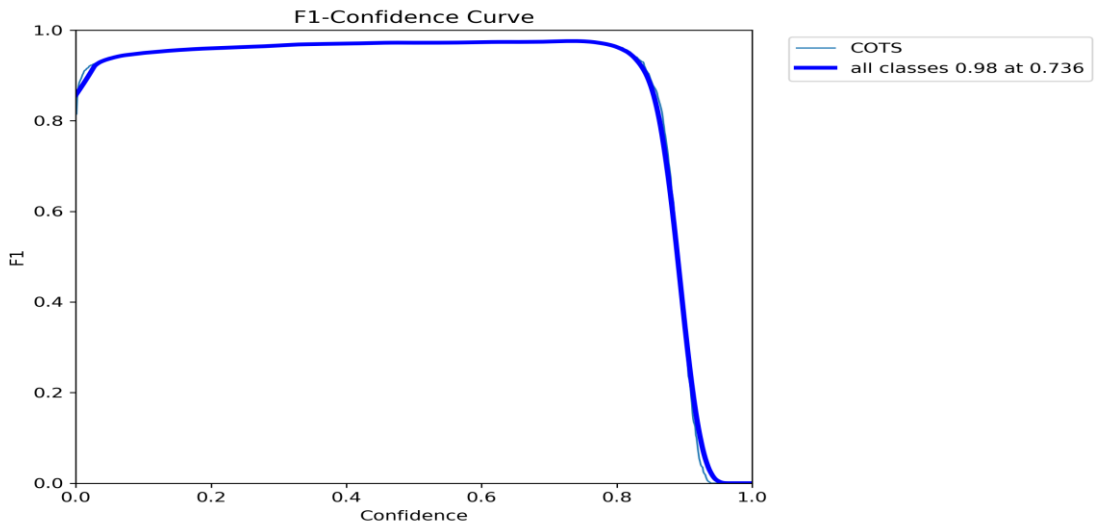

Figure 14. F1-Confidence graph.

From figure 14, we can see the value of F1-Score as 0.736. We know that there is always a trade-off between Precision and Recall as they go in the opposite direction. F1-Score (harmonic mean) tells us about the balance between both metrics for the model to perform well. It tells us that for this model, any precision from 0.736 and above is good. The model achieved a mean average precision of (0.79 mAP), meaning that it is a good model.

Every object detected as associated confidence score. The model uses probability to determine how certain it is to detect an object. It uses regression to determine the bounding boxes around the detected objects. As mentioned earlier, object detection deals with identifying an object (classification) and as well as localization (drawing bounding box around the object). It is interesting that the detector can handle these aspects very well.

As was said earlier, there is a trade-off between the precision and the recall. This is seen in the following figure.

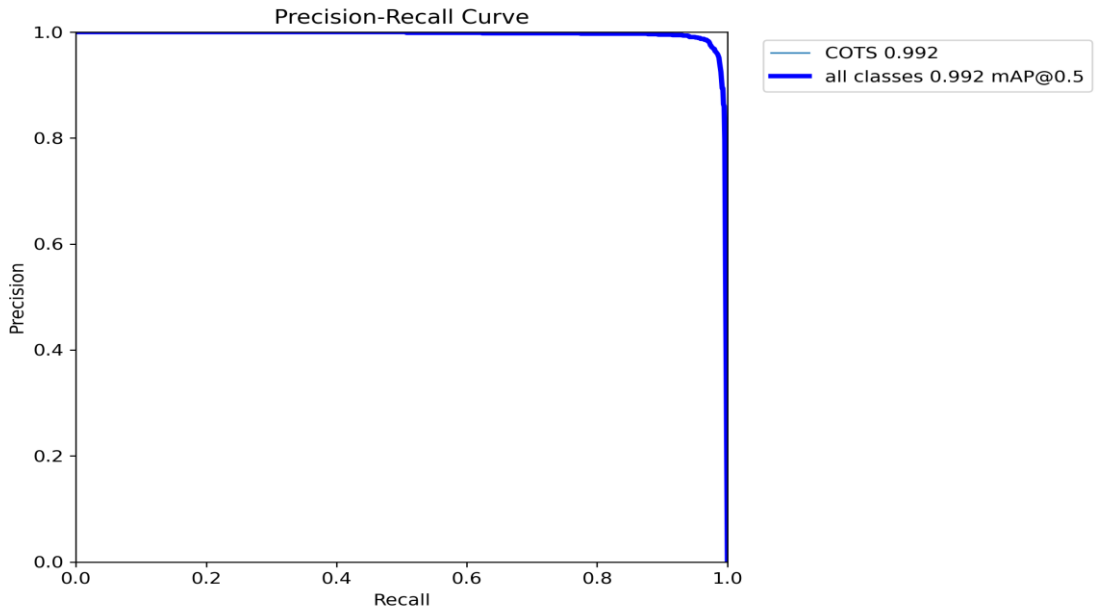

Figure 15. Precision-Recall curve.

Figure 15 shows the values of precision and recall various thresholds. Even though we want the precision to be high, we do not want to sacrifice the recall. The area under the precision-recall curve is a metric to show the overall performance of the detector. It tells us to detect most of the COTS, we need to set the threshold to 0.5. There is no problem if we miss a few COTS, in that case we want to maintain high precision over recall. We would have tried to maintain high recall and lower precision if missing COTS is critical. We can see the result of the model after training in figure 16. The model detected so many COTS on the training set and identified them with 0. However, figures 17 and 18 are of interest to us.

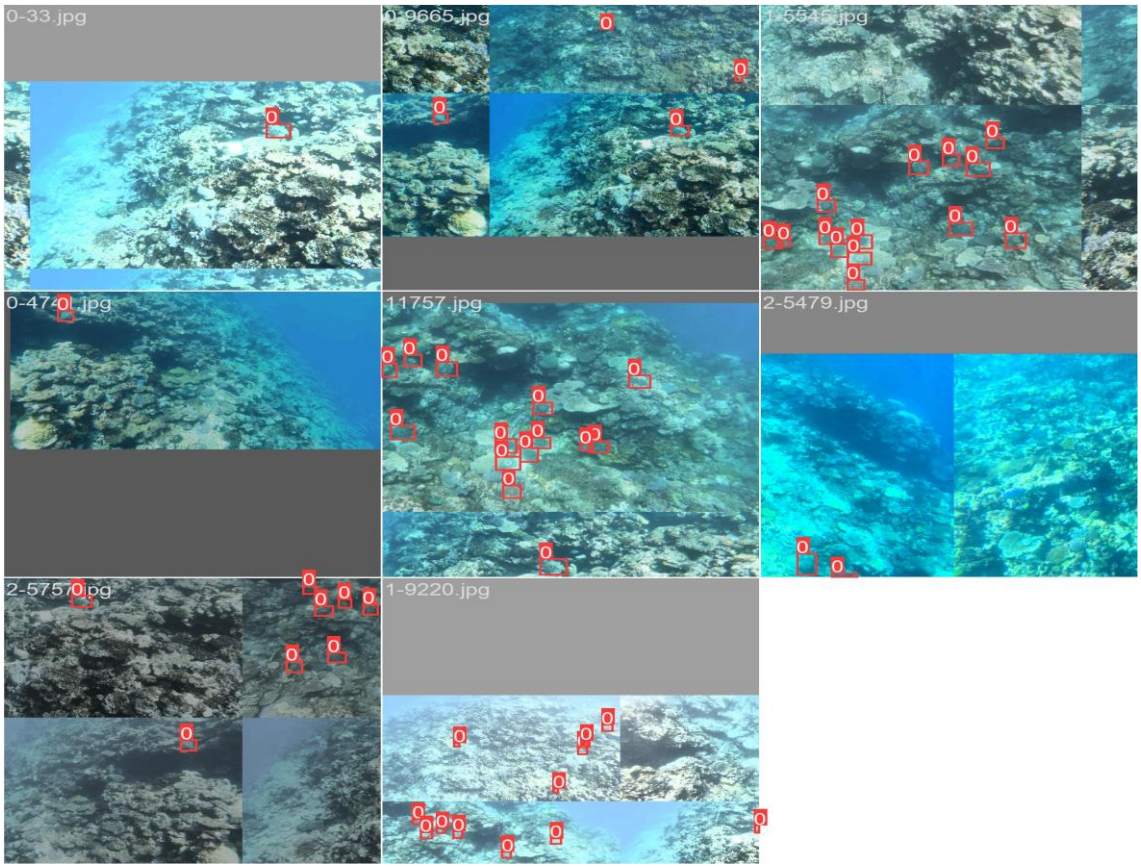

Figure 16. Sample images after training.

The training prediction was able to detect so many COTS objects, as we can see from figure 16. When compared with the ground truth, the prediction accuracy is high.

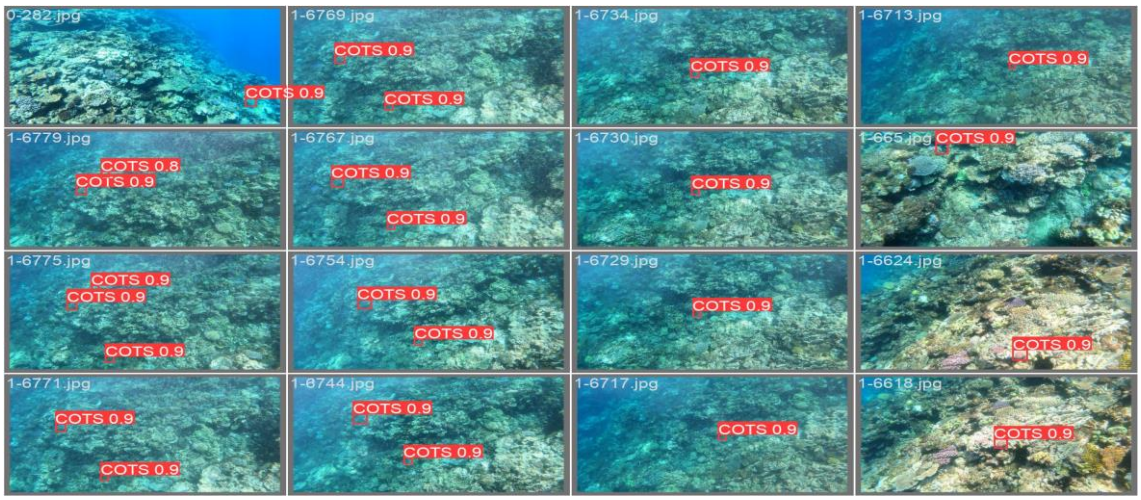

Figure 17. Validation sample showing the model's prediction on the validation set.

Figure 17, too, shows a high rate of detection by the COTS detector. When compared to the ground truth, we see that it detected so many COTS.

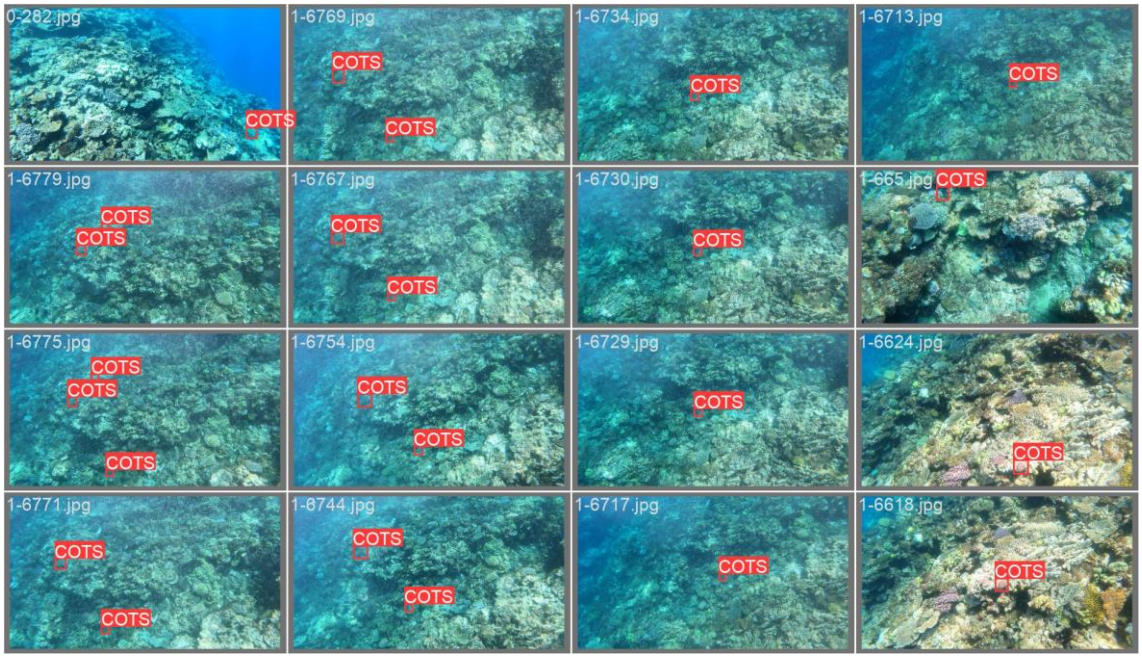

Figure 18. Validation sample showing the ground truth values of the validation set.

Figures 17 and 18 show the model's prediction on the validation set and the ground truth values respectively. Comparing both shows that they look alike, meaning the model was able to detect exactly most of the COTS, and their exact locations in the validation set images. This is a confidence boost that the model is performing well as expected. But once more, let us perform a test on the model using the test set to confirm if the results we got from model validation are justifiable.

The following table is the summary of the experiments and the results. It is good to look at the precisions when the confidence (IoU) was at 0.5 (mAP50) and between 0.5 to 0.95 (mAP50-95). Precision is the measure of how accurate the model makes correct predictions.

Table 2. The comparison of the base model and other parameters. P = Preprocessed data.

Model	Pre-train	Augmenta-tion	Prepro-cessing	Epochs	Precision (P)	Recall (R)	F1	mAP 50	mAP 50-95
YOLOv8m	No	No	No	100	0.960	0.957	0.958	0.981	0.674
YOLOv8m	Yes	No	No	100	0.972	0.974	0.973	0.987	0.717
YOLOv8m	Yes	Yes	No	100	0.973	0.967	0.970	0.988	0.702
YOLOv8m	Yes	No	Yes (P)	100	0.969	0.963	0.966	0.985	0.705
YOLOv8l	No	No	No	150	0.976	0.977	0.976	0.992	0.758
YOLOv8l	Yes	No	No	150	0.977	0.979	0.978	0.993	0.790
YOLOv8l	Yes	Yes	No	150	0.984	0.98	0.982	0.993	0.773
YOLOV8l	Yes	No	Yes (P)	150	0.979	0.978	0.978	0.993	0.786

Table 2 shows the models used as base models, when we used pretrained models (transfer learning), data augmentations, data preprocessing, the number of cycles through the training set (epochs), the precision, the recall, the F1, and the precisions with the confidence threshold at 50% (mAP50) and between 50% - 95% (mAP50-mAP95). For example, using YOLOv8m model trained from scratch, when the confidence threshold (IoU) is kept at 50% (0.5), the model makes average mean precision of 0.981mAP correction predictions, which is good. But, if the confidence threshold (IoU) is between 50%-95% (0.5-0.95), the average mean precision is 0.674 mAP, which is not so good. We can see that there is increase in the average mean precision when we trained with pretrained model, data augmentation, or preprocessed our training data. We get more increase in average precision when pretrained model was combined with both original and preprocessed train sets, recording 0.992 mAP on YOLOv8m and 0.995 mAP on YOLOv8l.

4.4 Model inference

Now, the model is set to be used for inference, to see how it performs on unseen data. Here, the test set is used. To perform test on the model, the predict method of the model object is called. The model is initialized with the 'best.pt' model from the training of the model. The following figures were obtained from the testing of the model with new or unseen data.

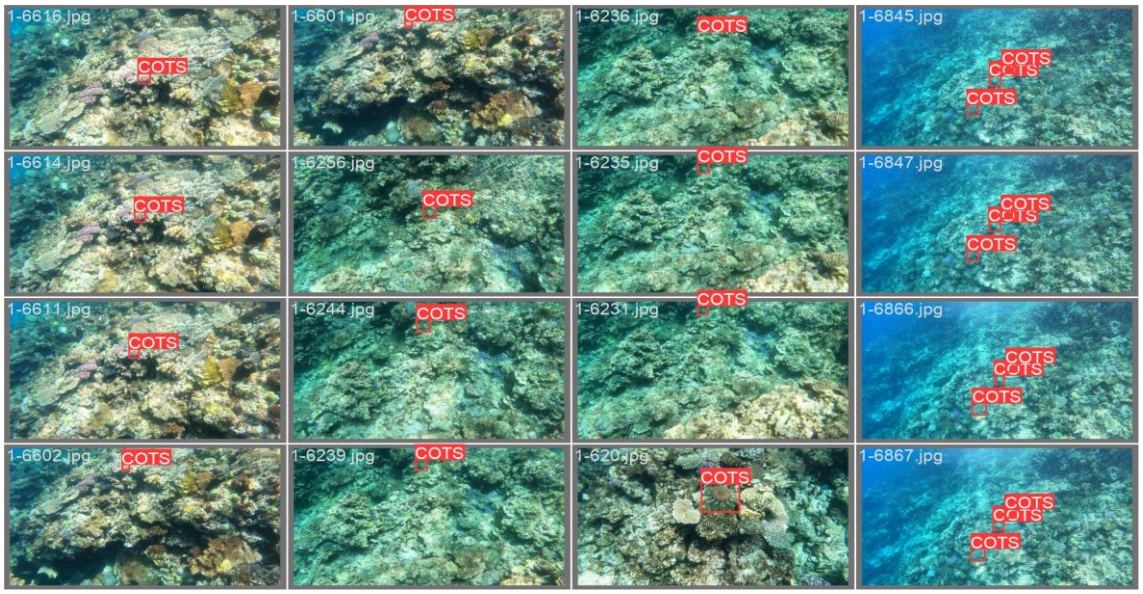

Figure 19. The ground truth of test set.

Figure 19 shows the ground truth of the test set. When compared to the prediction of the model in figure 20, the model got a lot of predictions correct.

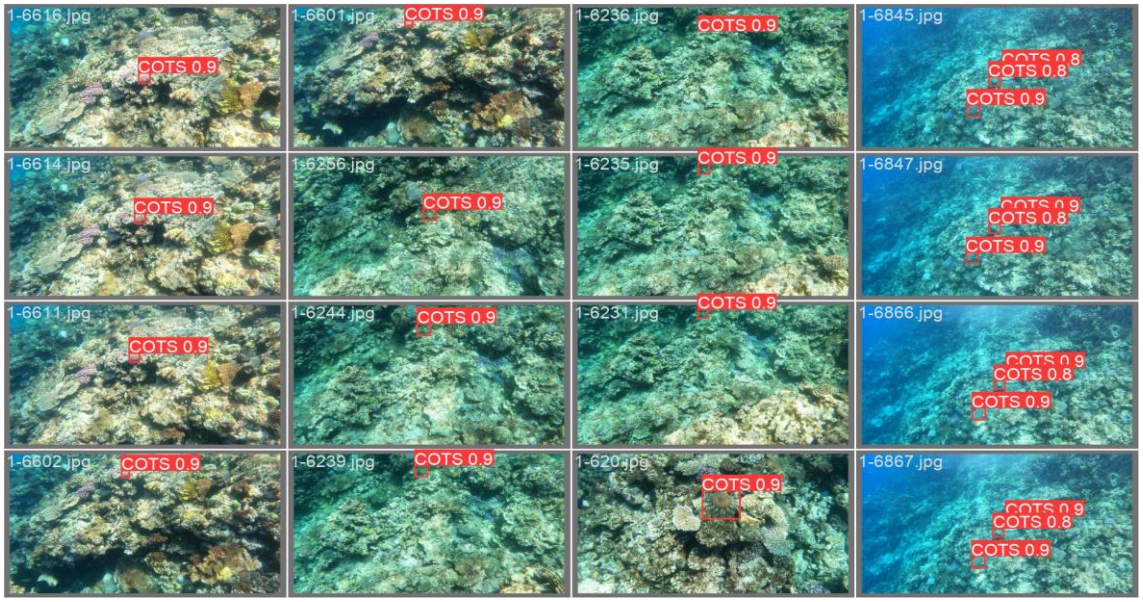

Figure 20. The predictions of the model on the test set.

Figure 20 is the sample of the prediction on the test. From table 2, we got the best average mean precision of 0.992 mAP with IoU of 50% for YOLOv8m and 0.992 mAP with IoU

of 50% for YOLOv8l. These are good predictions for the COTS-detectors. The COTS detector can detect COTS in both data images as well as video data.

5 Findings and analysis

This chapter will commence by presenting the evaluation of the results obtained in the last chapter with respect to our research questions. We have a model, but what do the results tell us about the accuracy of this model? Are there relationships between using a pretrained model, data augmentation, and data preprocessing and the performance of the model? This we explore in the first section and in the second section, we explore the limitation of the model.

5.1 Results and performance analysis

How has our model performed? Generally, good. But, what about regarding our research questions? The questions are:

- Can transfer learning from pretrained models effectively improve the generalization of COT detection models across varied and underwater environments?
- Can data augmentation and preprocessing methods improve the visibility and distinguishability of COTS from the reef background in underwater images and compensate for the class imbalance?

From table 2, the result summary, considering the first case when the YOLOv8m is trained from scratch and when a pretrained model is used, we can see that there is an improvement in the performance: Precision, from 0.960 to 0.972; Recall, from 0.957 to 0.974; F1, from 0.958 to 0.973. There is an improvement in the average mean precision for IoU=0.5, from 0.981 mAP to 0.987 mAP. The same is true in the second case with YOLOv8l model. The precision improved from 0.976 to 0.977, recall from 0.977 to 0.979. F1 and mean average precision increased as well, from 0.976 to 0.978 and 0.992 mAP to 0.993 mAP respectively. So, this answers the first research question. Higher precision, recall, and F1 show that there is an improvement in the generalization of the models to detect COTS across varied and underwater environments. We accept the first hypothesis.

What about the second research question? Table 2 shows that using YOLOv8m as a case, when there is data augmentation or preprocessing involved in the model, the

performance metrics improve. For example, comparing training with pretrained model and that involving data augmentation sees the mean average precision of the model using data augmentation is higher (0.988, 0.987), the pretrained model has a higher F1 value (higher by 0.003). This is beyond expectation because we expect data augmentation which involves data synthesis plus pretrained data to perform higher than the model that is only pretrained. The problem might be with the combination of data augmentation parameters applied. Experimenting with different combinations can help to see the combinations that will produce the better results. This should be investigated further.

The same thing applies to comparing pretrained YOLOv8m module and another using pretrained YOLOv8m plus preprocessed training set. For the former we have an F1 of 0.973 and average mean precision of 0.987 mAP for IoU=0.5 against the latter with F1 score of 0.966 and average mean precision of 0.985. We see that pretrained YOLOv8m performs better than the module that is pretrained plus data augmentation. Again, this is beyond expectation because we expect a model pretrained on a dataset with data augmentation to perform better in detecting COTS than just pretrained. This should be investigated on. As in the case of training a model from scratch and pretraining a model on a given dataset, there could be so many reasons for this. It could be that the preprocessed data introduced some noise to the dataset and ends up confusing the model when training, making the model unable to learn the features of the images. Of course, when the original training set is combined with the preprocessed set, there is a marked improvement. This could be explained because of using twice the amount of original training set (more data). Unfortunately, based on the results obtained do not accept the second hypothesis.

This model can also be evaluated based on the recommendations contained in the documentations of CSIRO dataset and YOLOv8. One of the tips by YOLOv8 for improving models is to ensure that the training dataset includes images from 1,500 and above for each class and that each class should have from 10,000 and above annotated instances. In our case, we had 4,919 images annotated as files. Each file had one or more

annotations, bringing it to a total of 11,898 annotations or instances of this one class, COTS class. YOLOv8 recommends image variety so that this is representative of the class and environment. In the case of CSIRO dataset, the dataset was indeed representative because the images were taken in different locations, different lighting, different angles and so on. These were organized in three different folders. Surely, this is good to improve the performance of the model.

Proper image labeling, or accurate and consistent image labeling in other words, is recommended by the documentation. This is because improper labeling will make the model not work. In our case, the CSIRO dataset came annotated by the publishers of this dataset. This removed lots of hard work that should have gone into labeling and annotations. The annotation phase is the hardest phase because time and care are needed in image labeling. This helped me concentrate on the job of training and programming. Of vital importance is the inclusion of background images in the training dataset. Background images are images without objects or classes. YOLOv8 recommends about 0-10% background images to be included in the dataset to reduce False Positives. This can lead to model performance enhancement. In the first and second training of the model, there were no background images added. I added 10% of the background images to the training set.

Based on the results obtained from this thesis experiment, the first hypothesis is true. Pretraining a model on a given or related dataset will transfer the weights and biases learned to the model, and this leads to improved performance. We say that transfer learning from pretrained models effectively improves the generalization of COT detection models across varied and underwater environments. For the second hypothesis, the result does not prove it right that data augmentation and preprocessing methods improve the visibility and distinguishability of COTS from the reef background in underwater images and compensate for the class imbalance. Even though it is true that data augmentation and preprocessing should improve the detectability of objects because of improved performance. I am still trying to figure out why the model performs very on our

training set but when the training set is augmented or preprocessed, there is not marked improvement in the performance.

5.2 Discussions and limitations

It is a good feeling to go through this journey of implementing a deep learning technique to develop a computer vision detector using the state-of-the-art YOLOv8. Even though there are positive achievements in the model, there are some limitations that affected this work. Unfortunately, the data augmentation and preprocessing methods included did not improve our model as expected. One reason is that it was quite challenging to find the combinations of data augmentation parameters that worked best to improve the model. Incompatible combinations might adversely add noise. I experimented with different combinations, especially those that will impact underwater images such as color, contrast, but there was no improvement. I am suspecting that the data augmentation parameters I chose inadvertently added noise to the images so that there is no marked improvement. Another is the processed images. It seems that the OpenCV functions for dehazing, Dark Channel Prior (DCP) and Contrast Limited Adaptive Histogram Equalization (CLAHE), ended up adding some noise to the data, making it difficult for the model to extract patterns from the images. Closely following this is the limited knowledge in image processing, which affected the efficiency of image enhancement.

There is also the hardware constraint. The use of computer systems with limited computational powers to process the images impacted the speed and memory capacity needed to train deep learning models. There was also data limitation, I had to work with data provided for the competition. The data may not adequately represent the variations in lighting, pose and size of COTS, or background. Again, the diversity of the COTS object could affect its detection by the model. I struggled with some concepts and tried to implement some of these in my work. Related to this is the real-time processing challenge. Meeting the real-time processing constraints while maintaining accuracy simultaneously can be a big challenge.

Some base models are faster but less accurate while others are more accurate but slower in performance. For instance, I used YOLOv8m and YOLOv8l. While YOLOv8m is faster but less accurate, YOLOv8l is slower but more accurate. YOLOv8x is the most accurate in YOLOv8 family, but because it is the slowest, I did not train with it. The deployment environment will also affect this model. This is because models perform differently under various conditions. For instance, this model will not run the same when deployed on a mobile application, on the cloud or other devices. Their computation power will affect the performance.

Another limitation is the choice of hyperparameters such as batch sizes. My system's computational power impacted the model training. I used batch sizes of 6 and 8. I observed a problem of memory with size 16, which would have been better and faster. I moved to Google Colab and signed up for Pro service. My first payment was not enough, and my use of GPU was restricted. I had to move back to use my system. The same was true of data augmentation. I chose data augmentation to overcome overfitting and to improve the model performance. But it exerted a lot of power on the resources of the GPU because it had to manipulate the images, like rotating, flipping, and blurring images to produce more. Another limitation I experienced was the issue of transferability. This model is only for this use case, for detecting COTS. This will not work for another use case. Using it for another use case means that we must train it on another custom dataset with its different annotations and tinker with the code also.

6 Conclusions

Deep learning computer vision has come a long way, and its application to COTS detection can help to save the oceans that have been threatened by various facts, including the overpopulation of the crown-of-thorns starfish. We have seen the computer vision journey and how the state-of-the-art computer vision is riding on the winds of previous computer vision techniques and development. Its application to COTS detection can lead to sustainable oceans. We have answered our two research questions. First, transfer learning from pretrained models can effectively improve the generalization of COTS detection models across varied and underwater environments because the model improved when using pretrained model. Second, data augmentation and preprocessing methods of our data do not improve the visibility and distinguishability of COTS from the reef background in the underwater images and take care of the class imbalance. They should have but unfortunately, they did not improve our models. I am still trying to understand why they did not in this case.

The implementation did not experiment with all possibilities there are because of the time, limited knowledge, and hardware constraints. This work is only limited to one class detection, that of COTS, and cannot be used for other use cases without adjustment to the model. But the knowledge in implementing this can be applied not only to COTS detection but the detection of other marine life using similar or related underwater images. Since they share underwater environment together, this work can be applied to marine life detection and underwater environments. This is one good implication for this study. Another implication is that this study will contribute to effective ecosystem management strategies based on the underwater images used. Also, this computer vision Google-sponsored competition plays a role in public awareness and education. It fosters collaboration and partnerships among many stakeholders such as governments, research institutions and corporate bodies. And finally, the detection methods of this project work can be part of a strategy to maintain the health of the ecosystems.

The ocean is packed with life of various kinds that not only support one another but support planet earth and its life forms. One area for further research and improvement is in real-time monitoring and reporting. In addition to detection, future work can be done on reporting their locations in the sea in real-time. This can enable timely intervention to know the exact locations of detected COTS to remove them and prevent damage. Another is to research image enhancement algorithms or dehazing techniques that are most effective in mitigating the effects of water turbidity and varying lighting conditions for better COTS detection. Future research can be on identifying discriminative features of COTS and minimizing noise since this is a challenge. Again, there is lack of research on ethical implications and potential impacts of AI-driven detection systems. This will be suitable for research purposes.

In conclusion, it is worth mentioning that this has been a learning journey. At the beginning of the thesis, I was scared not knowing if I could complete this and implement the COTS detection model in images and videos in real-time. This is a field that little has been written about and sometimes help is not forthcoming when you are stuck. However, it is a fulfilling experience.

References

Dayoub, F. et al. (2015). *Robotic detection and tracking of Crown-of-Thorns starfish. IEEE/RSJ International Conference* on Intelligent Robots and Systems (IROS), pp. 1921-1928.

Voulodimos, A. et al. (2018). *Deep Learning for Computer Vision: A Brief Review*.

Tamou et al. (2022). *Targeted Data Augmentation and Hierarchical Classification with Deep Learning for Fish Species I dentification in Underwater Images.* https://doi.org/10.3390/ jimaging8080214

Shi, P. et al. (2021). *Underwater Biological Detection Algorithm Based on Improved Faster-RCNN.* https://doi.org/ 10.3390/w13172420

Bresilla, K. et al. (2019). *Single-Shot Convolution Neural Networks for Real-Time Fruit Detection Within the Tree.*

Kayal, M. et al. (2017). *Bias associated with the detectability of the coral-eating pest crown-of-thorns seastar and implications for reef management.* http://dx.doi.org/10.1098/rsos.170396

Wang, J. (2022). *UTD-Yolov5: A Real-time Underwater Targets Detection Method based on Attention Improved YOLOv5*. https://doi.org/10.48550/arXiv.2207.0083

EM, D. et al. (2020). *Automating the Analysis of Fish Abundance Using Object Detection: Optimizing Animal Ecology with Deep Learning.*

Nguyen, Q. et al. (2022). *Detrimental Starfish Detection on Embedded System: A Case Study of YOLOv5 Deep Learning Algorithm and TensorFlow Lite framework. Journal Computer Institute.*

Zhang, Y. et al. (2022). *Early weed identification based on deep learning: A review*.

Yang, X. et al. (2020). *Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier*.

Zou, Z. et al. (2019). *Object Detection in 20 Years: A Survey.*

Rahman, A. et al. (2017). *Performance evaluation of deep learning object detectors for weed detection for cotton*.

Foxwell-Norton, K. (2017). *Saving the Great Barrier Reef from disaster, media then and now.*
https://doi.org/10.1177/0163443717692738

Géron, A. et al. (2019*). Hands-on Machine Learning with Scikit-Learn, Keras, and Tensor-Flow Concepts, Tools, and Techniques to Build Intelligent System*.

Muller et al. (2019). *Deep Learning for Dummies. John Wiley & Sons, Inc., New Jersey*.

Kumar, M. et al. (2022*). Beginning with Deep Learning Using TensorFlow, A Beginners Guide to TensorFlow and Keras for Practicing Deep Learning Principles and Applications. BPB Publications, India.*

Anami, B. et al. (2020). *Deep learning approach for recognition of yield affecting paddy crop stresses using field images.*
https://doi.org/10.1016/j.aiia.2020.03.001

Loy, J. (2019). *Neural Network Projects with Python, the ultimate guide to using Python to explore the true power of neural networks through six projects.*

Fletcher, C. et al. (2021). *Regional-scale modelling capacity for assessing crown-of-thorns starfish control strategies on the Great Barrier Reef. Reef and Rainforest Research Centre Limited, Cairns (59pp.)*.

Wu, X. et al. (2020). *Recent advances in deep learning for object detection*.

Zhao, Z. et al. (2019). *Object Detection with deep Learning: A Review. IEEE.*

Bao, Y. et al. (2019). *Computer vision and deep learning-based data anomaly detection method for structural health monitoring.*

Cha, Y. et al. (2018). *Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types*.

Zaidi, S. et al. (2022). *A survey of modern deep learning-based object detection model*.

Salma, P. et al. (2023). *A survey of underwater computer vision.*

Pathak, A. et al. (2018). *Application of Deep Learning for Object Detection.*

Pawan, S. (2022). *Capsule networks for image classification: A review.*

Ji, Y. et al. (2020*). CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances.*

Banwari, A. et al. (2022*). Comprehensive vision technique for freshness estimation from segmented eye of fish image*.

Mittal, P. et al. (2020*). Deep learning-based object detection in low-altitude UAV datasets: A survey.*

Li, Y. et al. (2022). *Key technologies in machine vision for weeding robots: A review and benchmark.*

Wang, C. et al. (2021). *Lychee Surface Defect Detection Based on Deep CNNs with GAN-Based Data Augmentation.*

Wang, N. et al. (2022). *Review on deep learning techniques for marine object recognition: architectures and algorithms.*

Modzelewska-Kapitula et al. (2022). *The application of computer vision systems in meat Science and Industry - A review.*

Kaur, R. et al. (2022). *A comprehensive review of object detection with deep learning. Panjab University, Chandigarh, India.*

Pooloo, N. et al. (2021). *Monitoring Coral reefs Death Causes with Artificial Intelligence.*

Pratchett et al. (2014). *Limits to Understanding and Managing Outbreaks of Crown-of-Thorns Starfish (Acanthaster spp.). Oceanography and Marine Biology: An Annual Review, 52, 133-200*.

Babcock et al. (2016). *Assessing Different Causes of Crown-of-Thorns Starfish Outbreaks and Appropriate Responses for Management on the Great Barrier Reef.*

Kroon, F. et al. (2021). *Fish predators control outbreaks of Crown-of-Thorns Starfish.*

Asad, H. et al. (2020). *The Computer Vision Workshop. Develop the skills you need to use computer vision algorithms in your own artificial intelligence projects.*

Clement, R. et al. (2005). *Toward Robust Image Detection of Crow-of-Thorns Starfish for Autonomous Population Monitoring. In: Australasian Conference on Robotics and Automation.*

Lamons, M. et al. (2018*). Python Deep Learning Projects. 9 projects demystifying neural networks and deep learning models for building intelligent systems*.

Howse, J. et al. (2020*). Learning OpenCV 4 Computer Vision with Python 3. Get to grips with tools, techniques, and algorithms for computer vision and machine learning.*

Rahmad, C. et al. (2020*). Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection.*

Johnson, B. et al. (2019). *Applied Supervised Learning with Python. Use scikit-learn to build predictive models from real-world datasets and prepare yourself for the future of machine learning.*

Saleh, A. et al. (2022). *Computer vision and deep learning for fish classification in underwater habitats: A survey.*

Mahony, N. et al. (2019). *Deep Learning vs. Traditional Computer Vision.*

Xie, Z. et al. (2019). *A Face Recognition Method Based on CNN.*

Patel, S. et al. (2021). *Object Detection with Convolutional Neural Network.*

Ren, S. et al. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposed Networks.

Liu, J. et al. (2021). The CSIRO Crown-of-Thorn Starfish Detection Dataset.

Reis, D. et al. (2023). Real-Time Flying Object Detection with YOLOv8. Georgia Institute of Technology.

Saitoh, K. (2021). Deep Learning from the Basics. Python and Deep Learning: Theory and Implementation.

Gupta, A. et al. (2021). Deep Learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues.

Martínez, J. (2021). TensorFlow 2.0 Computer Vision Cookbook. Implement machine learning solutions to overcome various computer vision challenges.

Planche, B. et al. (2019). Hand-On Computer Vision with TensorFlow 2. Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras.

Jiang, C. et al. (2022). Object detection from UAV thermal infrared images and videos using YOLO models.

Abirami, B. et al. (2020). Gender and age prediction from real time facial images using CNN.

Hall, M. et al. (2017). The potential role of the giant triton snail, Charonia tritonis (Gastropoda: Ranellidae) in mitigating populations of the crown-of-thorns starfish. Australian Institute of Marine Science.

Ju, R. & Cai, W. (2023). Fracture detection in pediatric wrist trauma x-ray images using YOLOv8 Algorithm.

Jun, G. et al. (2023). A novel application for real-time arrhythmia detection using YOLOv8.

Shetty, A. et al. (2021). Face recognition using Haar cascade and LBP classifiers.

Lin, T. et al. (2018). Focal loss for dense object detection.

Kumar, A. et al. (2020). Object detection system based on convolutional neural networks using single shot multi-box detector.

Viswanatha, V. et al. (2022). Real time object detection system with YOLO and CNN models: A review.

González-Sabbagh, et al. (2023). A survey on underwater computer vision.

Ditria, E. et al. (2020). Automating the analysis of fish abundance using object detection: optimizing animal ecology with deep learning.

Shanahan, J. et al. (2020*). Introduction to Computer Vision and Real Time Deep Learning-based Object Detection.*

Hong, H. et al. (2014). *Visual quality detection of aquatic products using machine vision.*

Thomas, T. et al. (2021*). Estimation of coral reef area through 2D images: Deep learning way using UNET.*

Ioannidou, A. et al. (2017*). Deep learning advances in computer vision with 3D data: A survey.*

Moniruzzaman, M. et al. (2017*). Deep learning advances on underwater marine object detection: A survey.*

Pham, Q. et al. (2019*). A *3D dataset: Towards autonomous driving in challenging environments.*

Hasirlioglu, S. et al. (2019*). Challenges in object detection under rainy weather conditions.*

Mees, O. et al. (2019*). Choosing smartly: Adaptive multimodal fusion for object detection in challenging environments.*

Pal, A. et al. (2017*). Deduce: Derive scene detection methods in unseen challenging environments.*

Hashmi, K. et al. (2022*). Exploiting concepts of instance segmentation to boost detection in challenging environment.*

Mukherjee, R. et al. (2021*). Object detection under challenging lighting conditions using high dynamic range imagery.*

Maiettini, E. et al. (2018*). Online object detection: A robotics challenge*.

John, V. et al. (2019*). RVNet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments.*

Ahmed, M. et al. (2021*). Survey and performance analysis of deep learning-based object detection in challenging environments.*

Jian, M. et al. (2020*). Underwater image processing and analysis: A review.*

Fayaz, S. et al. (2022*). Underwater object detection: Architectures and algorithms – a comprehensive review.*

Qin, H. et al. (2020*). When underwater imagery analysis meets deep learning: A solution at the age of big visual data.*

Pedersen, M. et al. (2019*). Detection of marine animals in a new underwater dataset with varying visibility.*

Blowers, S. et al. (2020*). Automated identification of fish and other aquatic life in underwater video.*

Ahmed, H. (2020*). Applications of Support Vector Machine Learning in Oceanography.*

Moura, M. et al. (2010*).* Sea Level Prediction by Support Vector Machines Combined with Particle Swarm Optimization. In 10th International Probabilistic Safety Assessment & Management Conference, At Seattle.

Ogunlana, S. et al. (2015*). Fish classification using Support Vector Machine*. African Journal of Computing & ICT 8.

Han, F. et al. (2020*). Underwater image processing and object detection based on deep CNN method*.

Sahu, P. et al. (2014*). A survey on underwater image enhancement techniques*.

Cao, Z. et al. (2016*). Marine animal classification using combined CNN and hand-designed image features*.

Pratama, Y. et al. (2021*). Application of YOLO (You Only Look Once) V.4 with preprocessing image and network experiment*.

Samantaray, A. et al. (2018*). Algae detection using computer vision and deep learning*.

Khan, M. et al. (2023*). Identification of crown-of-thorns starfish (COTS) using convolutional neural network (CNN) and attention model*.

Wu, T. & Dong, Y. (2023*). YOLO-SE: Improved YOLOv8 for remote sensing object detection and recognition*.

Sasagawa, Y. & Nagahara, H. (2020). Yolo in the dark-domain adaptation method for merging multiple models. In European Conference on Computer Vision; Springer: Berlin/Heidelberg, Germany, 2020; pp. 345–359.

Bahhar, C. et al. (2023*). Wildfire and smoke detection using staged YOLO model and ensemble CNN*.

Ang, G. et al. (2023*). A novel application for real-time arrhythmia detection using YOLOv8*.

Liu, W. et al. (2022*). Image-adaptive YOLO for object detection in adverse weather conditions*.

Ma, M. & Pang, H. (2023*). SP-YOLOv8s: An improved YOLOv8s model for remote sensing image tiny object detection*.

Mousa, A. (2023*). Underwater image enhancement using customized CLAHE and Adaptive Color Correction*.

Wang, X. et al. (2022*). Underwater fish image enhancement methods based on color correction*.

Kaushik, S. & Vigneshwaran P. (2022*). Underwater image enhancement using deep learning*.

Appendices

Appendix 1. Software and library requirements for YOLOv8

```
# Ultralytics requirements
# Example: pip install -r requirements.txt
# Base ----------------------------------------
matplotlib>=3.3.0
numpy>=1.22.2 # pinned by Snyk to avoid a vulnerability
opencv-python>=4.6.0
pillow>=7.1.2
pyyaml>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.8.0
torchvision>=0.9.0
tqdm>=4.64.0
# Logging -------------------------------------
# tensorboard>=2.13.0
# dvclive>=2.12.0
# clearml
# comet
# Plotting ------------------------------------
pandas>=1.1.4
seaborn>=0.11.0
# Export --------------------------------------
# coremltools>=7.0  # CoreML export
# onnx>=1.12.0  # ONNX export
# onnxsim>=0.4.1  # ONNX simplifier
# nvidia-pyindex  # TensorRT export
# nvidia-tensorrt  # TensorRT export
```

```
# scikit-learn==0.19.2  # CoreML quantization
# tensorflow>=2.4.1  # TF exports (-cpu, -aarch64, -macos)
# tflite-support
# tensorflowjs>=3.9.0  # TF.js export
# openvino-dev>=2023.0  # OpenVINO export
# Extras --------------------------------------
psutil  # system utilization
py-cpuinfo  # display CPU info
thop>=0.1.1  # FLOPs computation
# ipython  # interactive notebook
# albumentations>=1.0.3  # training augmentations
# pycocotools>=2.0.6  # COCO mAP
# roboflow
```