COMP9517 Project Report

Qi Yang
z5351944

Zihang Xu
z5286348

Chengxu Liu
z5444925

Zihong Ouyang
z5263915

Zhengye Ma
z5158505

Abstract

Projecting the environment of the Great Barrier Reef is an important task. Crown-of-thorns starfish(COTS) is a kind of harmful species which is harmful to the Great Barrier Reef. Thus, detecting the underwater COTS accurately is an indispensable technology, it can help people clear them which are attached on the Great Barrier Reef efficiently. However, detecting the COTS is a very challenging task due to the confusing shapes and noises from the underwater images. This task is even more difficult for human beings. In recent years, as the development of deep learning technologies progresses, detecting and locating the COTS from an underwater images became easier. So far, we tried using the latest deep learning models and improved the performance of it. Furthermore, we also tried using the traditional computer vision method to finish this task. According to our experiment, it is extremely hard for traditional methods to produce a satisfactory result, while the deep learning methods could easily get satisfactory performance. Besides, our improvement approach which is called WIoU could improve nearly 3.7% F2 score and 5.5% from Yolo5(nano) in this specific task. The maximum F2 score from our test is 0.82.

ACM Reference Format:
Qi Yang, Zihang Xu, Chengxu Liu, Zihong Ouyang, and Zhengye Ma. 2022. COMP9517 Project Report. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In this project, we tried several different methods to finish the starfish object detection task. To begin with, this task is a little different from other common object detection tasks, there are characteristics of:

- The detection speed should be as fast as possible. Since the background of the the image is not static, people will dive into water and use cameras to capture the image in real-time. The detection speed might need to be less than the video frame interval.
- The computation cost of the method should be as small as possible. Since underwater devices are usually very portable, their battery and performance are limited. Thus, the method should run on small devices like a smart phone.

To tackle these problems, we choose to use Yolo5(nano)[1]. It is a very light deep learning model, which can be deployed on mobile devices. Meanwhile, the accuracy of this model is acceptable and the speed of it is much faster than the other methods.

We successfully used Yolo5(nano) to realise the real-time detection of COTS. Then, we try to improve it. The basic idea is: Yolo5 is designed to conduct the general detection task, but the detection of the COTS is obviously different. Firstly, finding out COTS is a single object detection task, the feature of it can be easier to extract by the deep learning model. Secondly, the size of the COTS satisfies the Gaussian distribution according to our data set analysis. Thus, we try to improve the model in this direction.

Moving on, we mainly focus on optimising the loss function[2] [3] of Yolo5 and designed a new IoU formula which is called WIoU. The WIoU is based on the CIoU formula and it will give the different loss weight to the different objects, which depends on the probability of the width height ratio.

According to our experiment, the improvement of our approach is early 3.7% F2 score and 5.5% from Yolo5(nano). In the meantime, it also works in Yolo5(small) and improves both F1 and F2 score.

The intuitive traditional method we used is to iterate through the whole image and analyze the possibility of each region visited having a starfish object or part of an object. Lower possibilities will be neglected and higher ones will be further analyzed. There are some situations that the object image will be in multiple regions. To deal with those situations, we merge those regions for a final analyzation.

Another traditional method we tried is to use the selective search API in Opencv to locate the target object using segmentation, by calculating similarities between new region sets and subsets and adding strategies, we are able to produce bounding boxes and finally plot the bounding boxes that has less area difference onto the image.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Literature Review

2.1

In object detection, to locate the specific position of the target, usually the image is divided into sub-regions/patches, and then the sub-blocks are used as input and sent to the target recognition model.

The most straightforward approach to molecular blocks is the sliding window approach. The sliding window method is to exhaustively enumerate all sub-image blocks on the entire image according to the size of the sub-blocks.

However, the sliding windows approach is time-consuming and labor-intensive. That's why Region proposals were researched, hoping to provide all the locations where objects appear in the image, and wrap them with bounding boxes. In a good Region proposal, not every object must exist, and there will be many boxes without objects (false positive), among these Region proposals, we must be able to find the bounding box that frames the object.

The most famous method for obtaining Region proposals is Selective search, which has also been successfully applied to R-CNN and Fast R-CNN. This is because Selective search is fast to compute and can give high-quality Region proposals.

2.2

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. In simpler terms, for an image, each pixel is a piece of data and what image processing does is extract only useful information from the image, hence reducing the amount of data but retaining the pixels that describe the image characteristics.

Colour information can help us to identify and locate a target object at a macro level, for example we can use a colour histogram to see the colour distribution of a target area and then analyse the colour characteristics of the target we want to identify.

Shape features can help us to accurately use boundary information to assist us in target recognition, but shape features are not suitable for representation as feature vectors and are difficult to implement in practice.

At some point, we may want to extract texture features to further help us identify objects, as we have already exhausted the colour and shape features. Grey scale co-occurrence matrix (GLCM) and local binary patterns (LBP) are both recommended texture features

2.3

Since objects can be in different images, the size, shape and location of the object may vary, thus this traditional method might not be suitable for object detection. Deep learning is commonly used in object detection due to its accuracy and efficiency. It is a special type of machine learning and it contains several data analysis layers to extract higher level

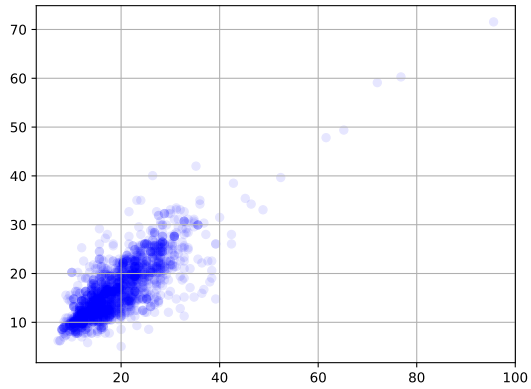

Figure 1. Objects Size Distribution

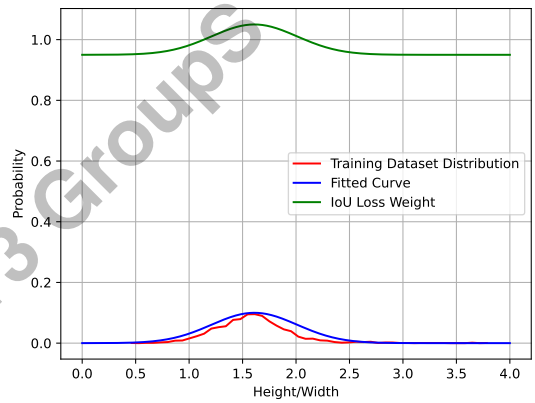

Figure 2. Probability Distribution

features from data. The layers can be divided into three parts which is: input layer, hidden layer and output layer. Each layer will produce an intermediate feature. The purpose of this extraction is to perform a non-linear high transformation of the feature. The goal of the deep learning model is to find the optimal weights and biases that minimize the loss in the given task, then the loss can be described as the error between the predicted and true results. After training the deep learning model, it can easily find out the specified object in different images.

3 Methods

3.1 Deep Learning Method

Overall, the task of detecting starfish from underwater images is very difficult. As it is even difficult for human beings, our first attempt is one of the latest deep learning models Yolo5. There are many versions of Yolo5, including nano, small, large and so on. We choose the nano version because the model might need to be transported onto the mobile devices.

3.1.1 Improvement. In fact, Yolo5 is a high-performance model that can be widely used in many fields. But for this specific task, there are still some space to improve it.

After we successfully train a model and that can detect the COTS accurately enough, we begin to consider the improvement of this model. There are many directions of improving this deep learning model, such as changing the backbone network, changing the augmentation method, changing the loss function and so on. On this point, we think changing the loss function has the most potential to improve performance. That is because the size distribution is regular and we could use a weight to give the different objects the different loss weight.

Thus, we perform a further analysis of the size distribution, the Figure 1 presents both the width and the height distribution. We can see that the height-width ratio might satisfy the Gaussian distribution. Then, we use Figure 2 to display the distribution of the probability in different height-width ratios. The red line represents the real distribution of the height-width ratio, the blue line is the fitted curve by the Gaussian distribution formula, and the green line represents the weight value of our approach.

We redesign the IoU loss function of Yolo5, the original loss function of Yolo5 uses CIoU as the box location loss function. This is the formula of our new-designed IoU which is called WIoU:

$Weight = sup \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot exp(-\frac{(x-\mu)^2}{2\sigma^2})$

$WIoU\ Loss = CIoU\ Loss(1 - (\frac{sup}{2} - Weight))$

σ and μ could be got by fitting the cure of probability distribution of height-width ratio. Sup is the hyper-parameter, 0.1 is the best value according to our experiment.

3.2 Selective Search Method

The Selective Search algorithm implemented in OpenCV was first introduced by *Jasper R. R. Uijlings* in their 2013 paper, Selective Search for Object Recognition[], the process can be broken down into 6 steps:

1. Generate a region set R
2. Calculate the similarity S = {s1, s2, …} of each adjacent area in the area set R
3. Find the two areas with the highest similarity, merge them into a new set, and add them to R
4. Remove all subsets related to step2 from S
5. Calculate the similarity between the new set and all subsets
6. Skip to step2 until S is empty

For step0: Selective Search works by over-segmenting an image using a super pixel algorithm (Jasper R. R. Uijlings used the Felzenszwalb method from Felzenszwalb and Huttenlocher's 2004 paper, an Efficient graph-based image segmentation).

From there, Selective Search seeks to merge the super pixels to find regions of an image that could contain an object.

For step1: Selective Search merges super pixels in a hierarchical fashion based on 5 key similarity measures

1. **Color similarity**: Compute a 25-bin histogram for each channel of an image, concatenating them together, obtaining a final descriptor that is *25×3=75-d*. Color similarity of any two regions is measured by the histogram intersection distance.
2. **Texture similarity**: For texture, Selective Search extracts Gaussian derivatives at 8 orientations per channel (assuming a 3-channel image). These orientations are used to compute a 10-bin histogram per channel, generating a final texture descriptor that is *8x10x=240-d*. Histogram intersection is once again used to compute texture similarity between any two regions.
3. **Size similarity**: The size similarity metric that the selective search method prefers that smaller regions should be grouped before larger regions. By enforcing smaller regions to merge earlier, we can help prevent a large number of clusters from swallowing up all smaller regions.
4. **Shape similarity/compatibility**: The idea behind shape similarity in selective search is that they should be compatible with each other. Two regions are considered "compatible" if they "fit" into each other (as if filling gaps in our regional proposal generation). Furthermore, shapes that not touch should not be merged.
5. **Meta-similarity**: A final meta-similarity acts as a linear combination of the color similarity, texture similarity, size similarity, and shape similarity/compatibility.

3.3 Iterative Search Method

This method is quite intuitive. The simplest way to detect Starfish is to try to search it throughout the whole image. However, the disadvantage is obvious, the computation cost greatly outweighs other methods. And the advantage is that this idea is simple and easy to implement. 3 Based on the analysis of object image size, it is clear most of the objects are of size (40, 40), and very few objects have size greater than (80, 80) (Almost None of the image has size greater than (120, 120)). Hence, the box size (40, 40) is used considering most objects can be bounded by mostly 1-2 boxes and max 3 boxes.

The method contains following steps:

1. Divide the image into many small regions of size (40, 40). (For those at boarders, ignore pixels that are outside of the original image)
2. Loop through each region and predict the probability using the classifier trained by SKlearn api and record the probabilities as well as their positions if probability is good enough otherwise ignore it (using threshold).

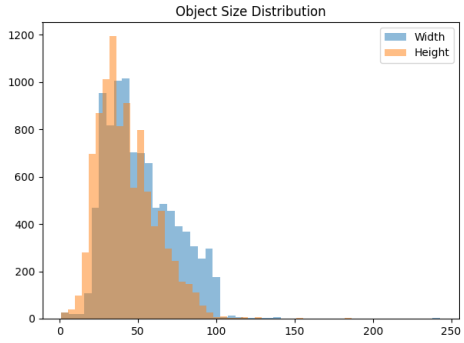

Figure 3. Object Image Size

3. Sort information recorded before by probability value (from highest to lowest).
4. Loop through sorted list and check if its 8 neighbors also have some probabilities being part of a starfish image.
5. For each box in previous step, use a larger box to cover the original box and its neighbors, if necessary, then pop them from the list and continue.
6. Every box obtained in Step5 is larger than the actual object, so move each side towards the center if the probability predict on the new box is higher, if not then draw the box in the original image.

4 Experimental Results

4.1 Metric Introduction

There are several ways to evaluate object detection models we used such as YOLO, including mAP and Fn score.

- mAP(Mean Average Precision): mAP compares the real bounding box and the detected box and returns the score based on the difference, a higher score means more accurate detection.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

  Where AP is the average precision of the class and n is the number of classes.
- F1 score: F1 score measures the performance of a classifier by examining how well the classifier can distinguish positive cases, it is derived from the harmonic mean of the true positives + the false positives and the true positives + false negatives, also known as the precision and recall. The closer the score is to 1 the more accurate the model is.

$$F1 = 2\frac{(tp+fp)(tp+fn)}{tp+fp+tp+fn}$$

  Where tp is true positives, fp is false positives and fn is false negatives
- F2 score: F2 score has the same basis of F1 score, both are calculated from the harmonic mean of precision and recall, however F1 score has a equal weight to both precision and score, and F2 score is more weighted towards the recall, which is where the percentage of the correctly identified cases among all the real positive cases. Same as the F1 score, the closer the score is to 1 the more accurate the model.

$$F2 = 5\frac{(tp+fp(tp+fn)}{4(tp+fp)+tp+fn}$$

For this particular project, we prioritized the Fn score model, especially the F2 score model, weighting more on the recall section means that we are more tolerate towards the false positive cases, in other words we rather detect more false starfish than missing a real one.

4.1.1 Data set Preparation. To test our methods, we generate a new data set from the original data set for these reasons:

1. The image is too large to train a deep learning model. Most modern models use the input image size around 512x512, but the original image size of the GBR data set is up to 1280x720 because it comes from a video.
2. The amoung of samples are too many, and most of them are similar. Because the data set is generated from three videos, the adjacent frames will be very similar. In addition, there are many sample images without object position labels, this kind of images should be excluded. Thirdly, due to the computation of our devices, we could not use a very large data set. Thus, we dropped the number of samples to make sure the workload of the computation is acceptable.
3. The label format is not a standard format, which needs to be converted into a format that we can use.

At this point, we collect 4919 samples from the original data set. Then we randomly select 800 samples as the training data set and 200 samples as the testing data set. After that, we reshape the images from 1280x720 to 512x288. Finally, we convert the labels to coco format.

4.2 The result of our approach

Figure 7 presents the performance of our WIoU on Yolo5(nano) approach and Figure 8 presents that on Yolo5(small). Overall, the experiment results is acceptable according to Figure 4, Figure 5 and Figure 4. All these example detection results are from the test dataset which is not known by the model before detecting them. Because the yolo5 is a very efficient object detection model, the baseline is good enough. The F1 score of it is 0.812 and the F2 score of it is 0.778. However, the F1 score after our improvement is 0.840 and F2 score is 0.821. The improvement of our approach is about 4% difference in this specific task.

4.2.1 Experiment Results.

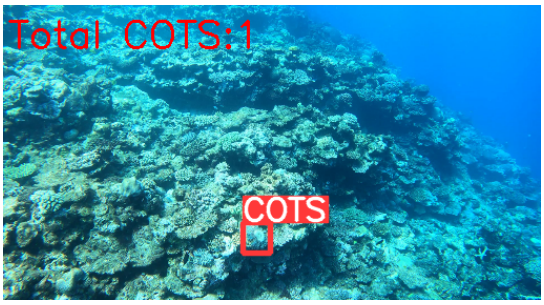

Figure 4. Detection Result by Yolo5

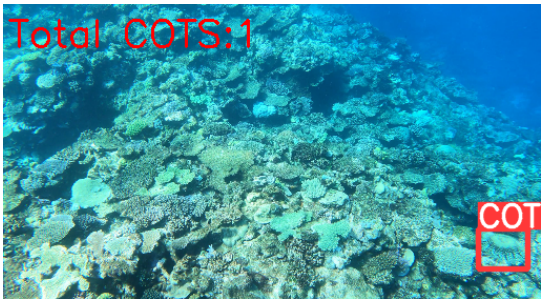

Figure 5. Detection Result by Yolo5

Figure 6. Detection Result by Yolo5

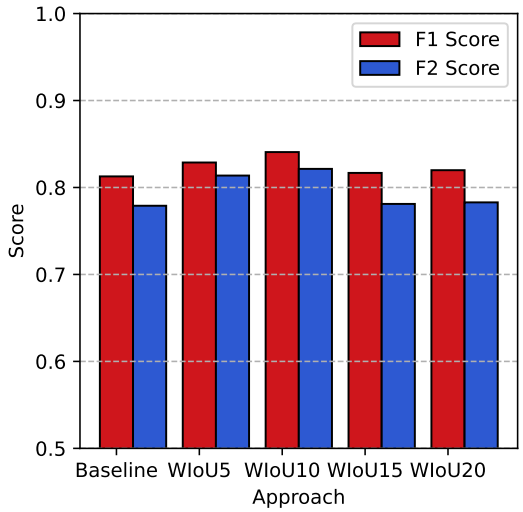

Figure 7. Fn Scores by Yolo5(nano)

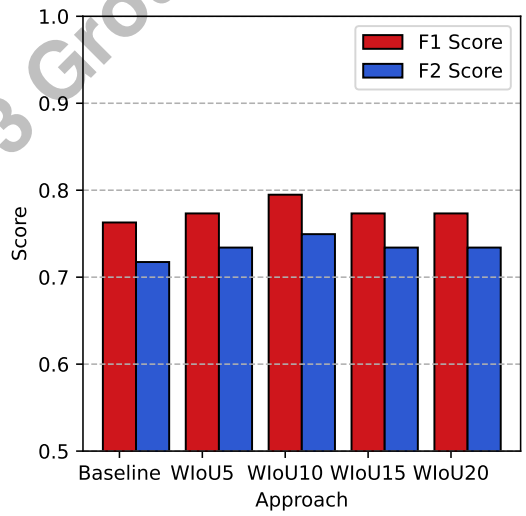

Figure 8. Fn Scores by Yolo5(small)

4.3 Selective Search Experiment and Results

We used the Opencv *createSelectiveSearchSegmentation* API to create Selective Search Segmentation Object using default parameters and add Strategy base on color and fill. After the algorithm generate the region bounding box. I compute each bounding box area and compare to the true bounding box area. If the area difference less than 200 we add the bounding box to the image. One of the plot results is shown in figure 11

We notice that the result of this traditional method is not really ideal, the reason we suspect is that the starfish in the sea are too similar to the background. Such as figure 12 and figure 13

4.4 Iterative Search Experiment and Results

We use SKlearn API to train the classifiers with 1600 starfish images as positive samples and extract 400 random background images as negative samples: 1

This Classifier will take in an image and output the possibility of starfish.

Based on the results of various classifiers we tried, SVC (Support Vector Classifier) is put into use as it can find the optimal hyperplane that divides the data into two classes, which is exactly what we need to distinguish between the starfish and the background.

We then further adjust the parameter value of the classifier and obtained the following results: 2 By examining the table,

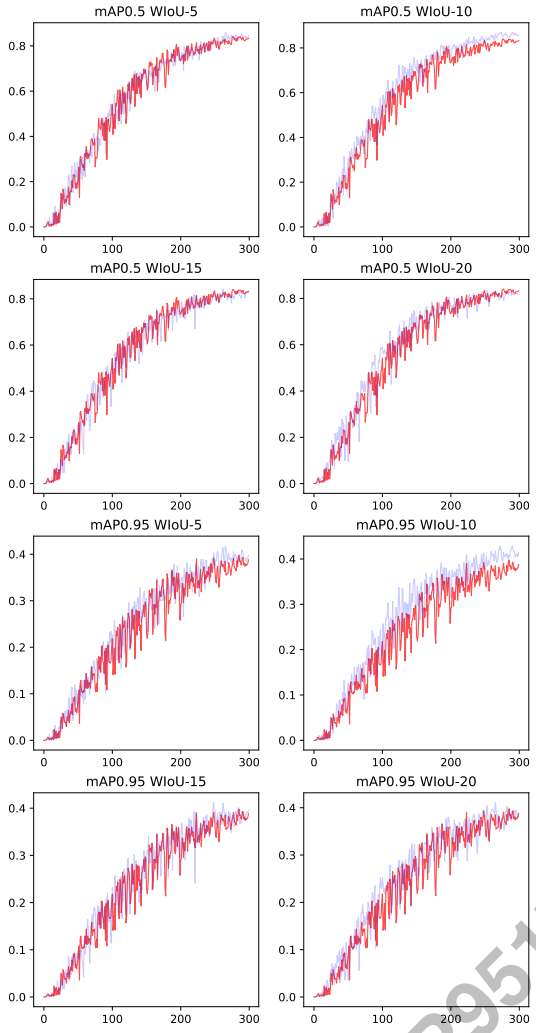

Figure 9. Training tendency by Yolo5(nano)

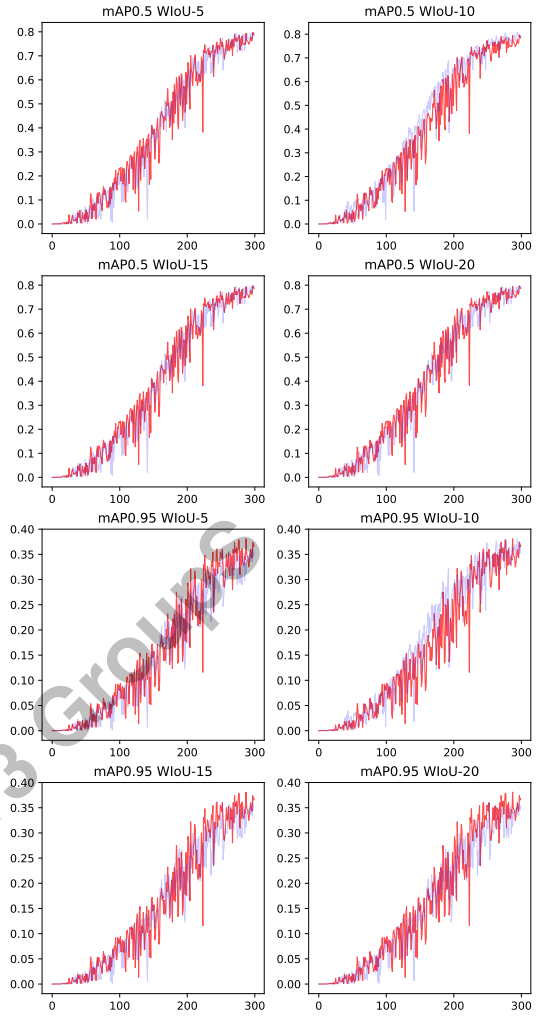

Figure 10. Training tendency by Yolo5(small)

Model	F1 score	ACC score
KNeighborsClassifier(n_neighbors=3)	0.8037	0.7875
SVC(C=10)	0.8608	0.8625
SVC(C=1, gamma=2)	0	0.5175
GaussianProcessClassifier()	0	0.5175
DecisionTreeClassifier(max_depth=5)	0.6979	0.6775
RandomForestClassifier(max_depth=5, max_features=1, n_estimators=10)	0.7097	0.7075
MLPClassifier(alpha=1, max_iter=500)	0.5085	0.6375
AdaBoostClassifier()	0.7321	0.7475
GaussianNB()	0.6621	0.6300

Table 1. Classifiers Training Results

we then pick classifier SVC(C = 10) for highest F1 and ACC scores.

The detect result is as follow: 14

SVC C value	F1 score	ACC score
C = 1	0.8356	0.8175
C = 5	0.8674	0.8575
C = 10	0.8703	0.8625
C = 15	0.8662	0.8575
C = 20	0.8692	0.86
C = 30	0.8668	0.8575

Table 2. SVC Training Results

This method is not working very well, we suspect the reason is the starfish image shares a lot of common features with the background including color, textures, etc.

4.5 Deep Learning Method Detection Result

The YOLO5 model captures starfish: 15 16

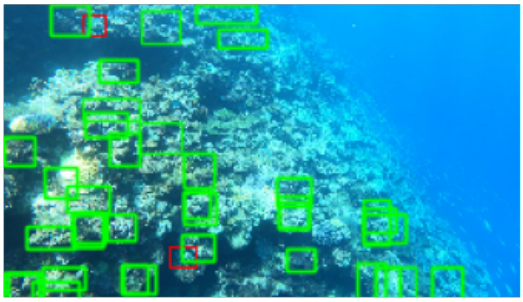

Figure 11. Selective Search Result

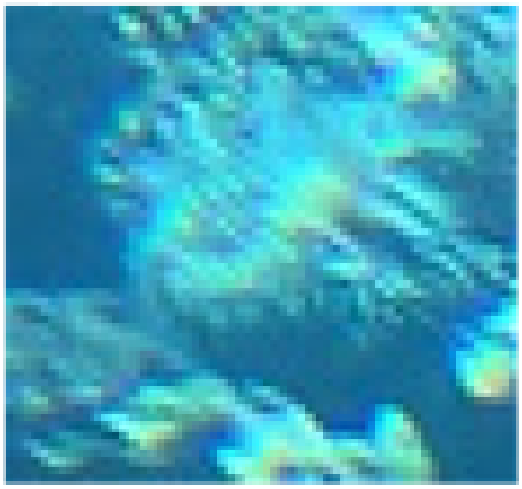

Figure 12. Starfish example 1

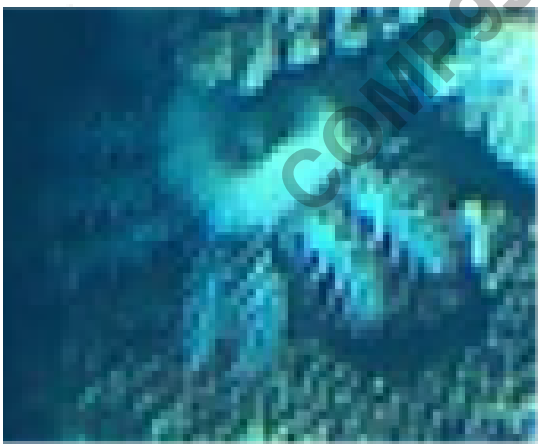

Figure 13. Starfish example 2

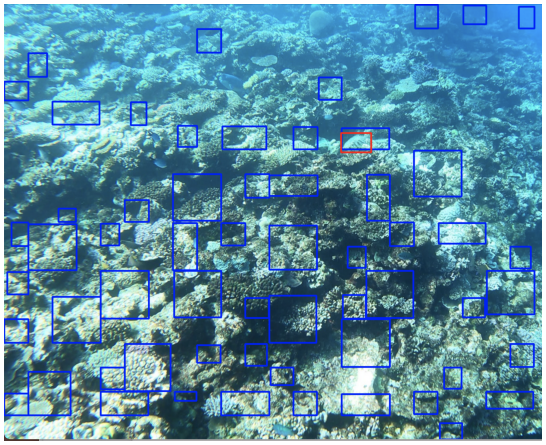

Figure 14. Iterative Search Result

Figure 15. Iterative Search Result

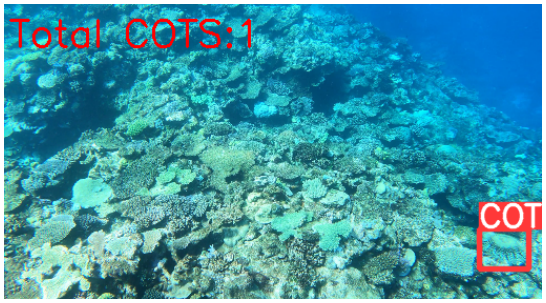

Figure 16. Iterative Search Result

5 Discussion

The Iterative search method can not distinguish between background and object very clearly, it often categorizes background as object. Also, It is hard to find an appropriate static threshold for all of the images as the crucial image feature, like size and shape, varies. The high similarity color or shape area will also be treated as the object. Besides, the iteration and predictions requires huge time complexity, and the sliding window of the iterative algorithm is set empirically and does not adapt to the specific area of the image, which also results in a high complexity of the algorithm. However, efficiency is exactly what we need. Hence, it is not suitable to use this method to detect starfish. We trained a binary classification model to help us identify objects, and eventually choose to use the SVC model, but this classification was too coarse, and it was difficult to set classification thresholds

based on this, despite having previously obtained RGB histogram distributions and texture features from our data set analysis.

The selective search method does a better job of distinguishing the background and the starfish, but for this object detection task it is still not ideal as it is hard to find out the exact object, the similar color or shape area will be treated as the object. However compare this method to the iterative method the accuracy is increased dramatically.

The Yolo5 method after sufficient training can detect object efficiently, F1 scores consider recall and precision to be equally important, F2 scores consider recall to be twice as important as precision.So comparing the F2 score, according to the chart, WloU10 approach behaves the best and Baseline has the least score. And compared to the best result in Kaggle, some models have better F2 score more than 0.9. So we think that the size of the training set sample data and the training method may be factors that affect the final result score

6 Conclusion

Three different methods were implemented to detect starfish objects. The given image sets were used to train and test models. By comparing the detection output, the deep learning method using YOLO5 gives more promising result. And other two methods have obvious deficiency in detection.

6.1 What Worked

The deep learning method captures starfish with minor errors. Selective and Iterative methods can recognize Starfish in some extent as well.

6.2 What Did Not Work

The major problem that both two OpenCv methods have is object miss-detection. This is caused by similar color or shapes on edges of the bounding box. Even though we try to extract starfish image into training, but there are still a small part of pixels, compared to the real object part, belong to the background. This might 'confuse' classifier to make the right prediction.

And the iterate search method cannot deal with the scenario where several objects overlap, as it will combine those into one box.

6.3 Possible Improvement

The Starfish object and background feature can be more precisely extracted by detecting the Object edge and separate them as much as possible. With those features, classifier can be better trained to detect Object.

Even the deep learning method can be further improved by configuring the hyper-parameters of the YOLO5 model.

7 Reference

References

[1] Alexander Wong, Mahmoud Famuori, Mohammad Javad Shafiee, Francis Li, Brendan Chwyl, and Jonathan Chung. Yolo nano: a highly compact you only look once convolutional neural network for object detection. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 22–25. IEEE, 2019.

[2] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.

[3] Hongyu Zhai, Jian Cheng, and Mengyong Wang. Rethink the iou-based loss functions for bounding box regression. In *2020 IEEE 9th joint international information technology and artificial intelligence conference (ITAIC)*, volume 9, pages 1522–1528. IEEE, 2020.