COMP 9517 Group Project

Group Name: Group GO GO GO

Group Member and ID:

Name	Student zID
Zehao Li	z5212851
Zheyuan Shao	z5334189
Xiyu Huang	z5373216
Xiaolan Zhang	z5400028
Haoyu Zang	z5326339

1. Introduction

The Great Barrier Reef is located in Queensland, and it is currently the largest coral reef in the world. But its habitat is currently under threat, partly due to the overpopulation of crown-of-thorns starfish (COTS). COTS feed on corals, and their overpopulation can cause severe damage to the Great Barrier Reef.

Detecting COTS outbreaks usually requires large-scale video surveillance, which would be inefficient if humans simply observed the COTS in videos. Hence, it is essential to analyze video images with the help of computer vision.

In this project, we use computer vision techniques to analyze the captured COTS video images and help people to observe better and count the number of COTS. We preprocess the dataset. Then we test two algorithms, Faster R-CNN and YOLOv5, and improve both. And we propose a VGG+SVM approach for target detection. The following article will describe how we carried out this project.

2. Literature Review

In this section, we will briefly overview the content of our methods.

2.1 Single Image Haze Removal Using Dark Channel Prior

In the paper Single Image Haze Removal Using Dark Channel Prior, the authors find a statistical law, Dark Channel Prior, which shows that in a fog-free image, every local area has some very dark parts. Since haze is always white, once the image is affected by moisture, these otherwise dark parts become gray [1]. Therefore, as long as the very dark details of each pixel in the image are gathered in a certain way, they can form a dark channel map in the picture. Based on this rule, the authors propose a method for image haze removal. In this project, we used this method to improve the clarity of the images in the database and to make the starfish in the pictures more easily identifiable.

2.2 YOLOv5

YOLOv5 detects object models trained on the COCO dataset with Test Time Augmentation (TTA) [2], model ensembling, and much simple functionality included. The new feature of YOLOv5 contains Serialization of models, distributed training, FP16 model storage and reasoning, breakpoint training, Tensorboard training visualization, second stage classifier, model ensembling, model pruning, and many great features. Progress of training and detection of YOLOv5 can be broken into three parts.

2.2.1 **Handle Data**

Format the dataset and represent them in the following format to make it easier to read for the next steps < object-class-ID> <X center> <Y center> <Box width> <Box height>

2.2.2 **Configuration Files**

There are mainly three files that Yolov5 takes to the configuration system. 1. The data-configurations file shows the parameters of datasets that provide the paths to the trains, validation, and many parameters. 2. The model architecture can be seen in the model-configurations file. 3. The hyperparameters-configurations file contains hyperparameters for the training, which includes the learning rate and much information

2.2.3 **After Treatment**

1. Training from scratch. 2. Transfer learning. 3. Feature extraction 4. Fine Tuning

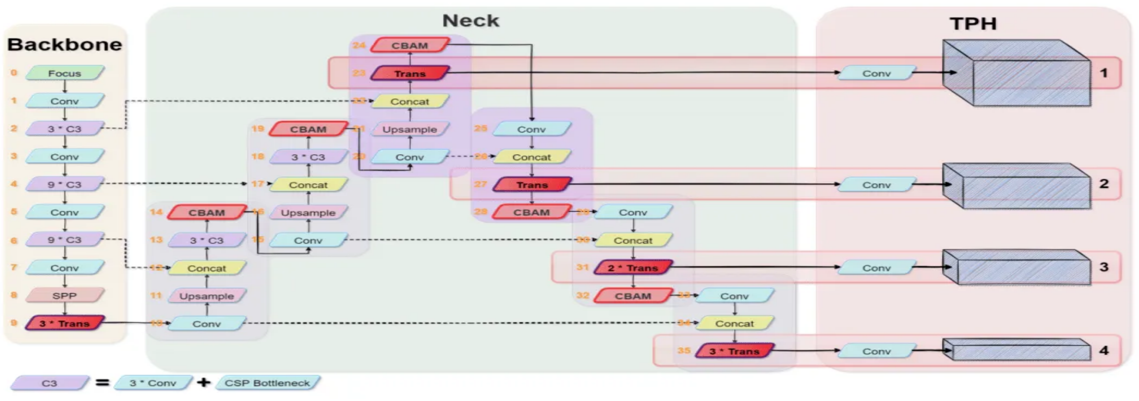

Figure 1. TPH-YOLOv5 [3]

2.3 R-CNN and Faster R-CNN

2.3.1 R-CNN

R-CNN, also known as Region with CNN feature, is considered the pioneering work in target detection using deep learning. The R-CNN consists of three main modules, the first module is used to generate a series of candidate regions, the second module uses a convolutional neural network to create feature vectors of a specific length, and the third module uses a class-specific support vector machine (SVM) to classify each region [4]. The three modules of R-CNN and their operational flow are more clearly depicted in the following diagram.

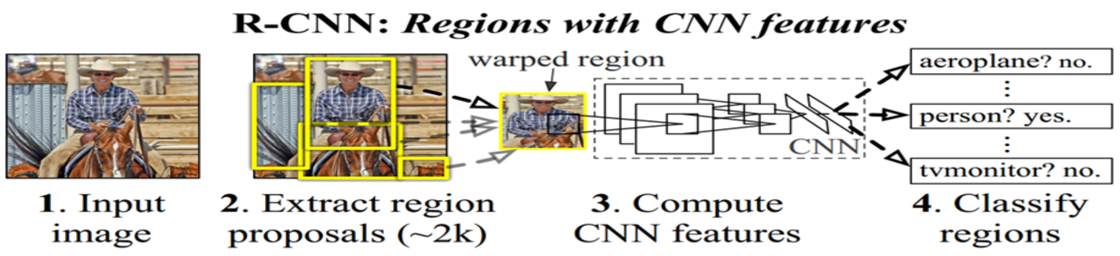

Figure 2. R-CNN modules and processes [4]

2.3.2 Faster R-CNN

Faster R-CNN is an improvement on Fast R-CNN [5]. In Faster R-CNN, a new algorithm, Region Proposal Networks (RPNs), was born. RPNs use a deep convolutional neural network to implement the process of computing region proposals. It can significantly reduce the time of candidate region generation algorithms by sharing convolutional layers with Fast R-CNNs [6]. Previously, it could take 2s to generate candidate regions, but RPNs only take 10ms, which is a significant improvement.

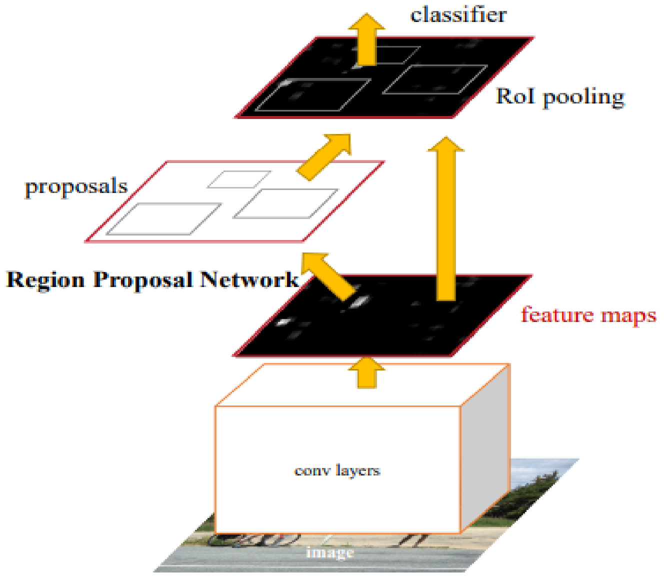

Figure 4 . RPNs flow chart [6]

2.4 FPN

Feature pyramid networks (FPN) use skip connection to fuse low-level features and high-level features to build top-down paths, which are eventually detected at multiple levels[7]. Its structure is shown in the diagram below.

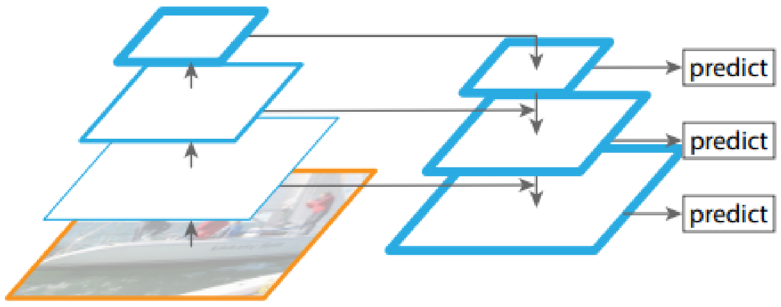

Figure 5. The structural diagram of the FPN[7]

2.5 SVM

Support vector machine (SVM) is a generalized linear classifier that uses a supervised learning method to classify binary datasets [8]. Its decision boundary is to calculate the maximum-margin hyperplane towards the learning sample.

SVM is a sparse and robust classifier that uses the hinge loss function to compute empirical risk, and its solving system has a regularizer that will optimize structural risk. The kernel method is used for non-linear classification within SVM, a standard kernel learning method.

There are two main theories of SVM. The first main theory is Linear Classification. Within Linear classification, there are three main features. 1. Linear separability: Given data and learning targets within the classification problems, if a classification problem has linear separability, there will be two parallel hyperplanes as the boundary to sort the samples. All positive and negative samples positioned above the edge can be regarded as support vectors. 2. Loss Function: If a classification problem has no linear separability, using a hyperplane will cause classification loss. Hinge loss will minimize the surrogate loss. 3: Empirical risk and structural risk: If a classification problem is a regularizer problem, the two theories above will not be suitable. Hence, empirical and structural risk will be used to compute the minimum risk. The second leading theory is the Kernel Method. We can see the flow in figure 5.

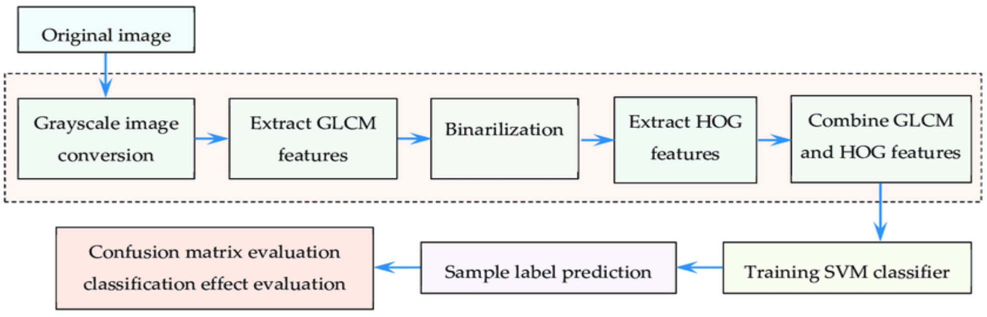
Original image
Grayscale image conversion
Extract GLCM features
Binarilization
Extract HOG features
Combine GLCM and HOG features
Training SVM classifier
Sample label prediction
Confusion matrix evaluation classification effect evaluation

Figure 5. The flow diagram of SVM [9]

2.6 VGG

VGG has no massive difference from the traditional CNN model. They even use a similar pipeline. However, there are four features for VGG. 1. Small convolution kernel, the VGG replaces all kernels With 3X3 layers. 2. Small pool stride. VGG contains 2X2 max pool stride. 3. Deeper layers and broader features of the map. 4. Total Convolution. We can see the network of VGG in figure 6 [10].

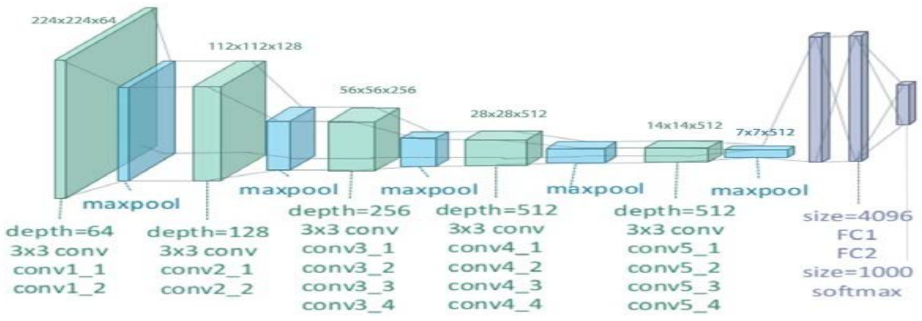

Figure 6. The network of VGG-19 [11]

3. Methods

In this section, we explain the methodology and process of our project in more detail. The specific function of our project is 1. pre-processing the data, 2. testing both Faster R-CNN and YOLOv5 algorithms, 3. modifying the two algorithms in the previous step, and designing a VGG+SVM method.

3.1 DataSet Pre-processing

The dataset we use is from the TensorFlow - Help Protect the Great Barrier Reef in the Kaggle competition [12]. For the data part, we extract the data with labels from the train.csv file and record it in a JSON file. We filtered 4919 images with tags from 23501 photos. We divided the resulting JSON file and images with tags into a training set and a test set in the ratio of 8:2. We then converted the JSON file into XML and TXT format and used the method mentioned above in Single Image Haze Removal Using Dark Channel Prior write our code to improve the clarity of the image. The image below compares the image before and after it has been processed.

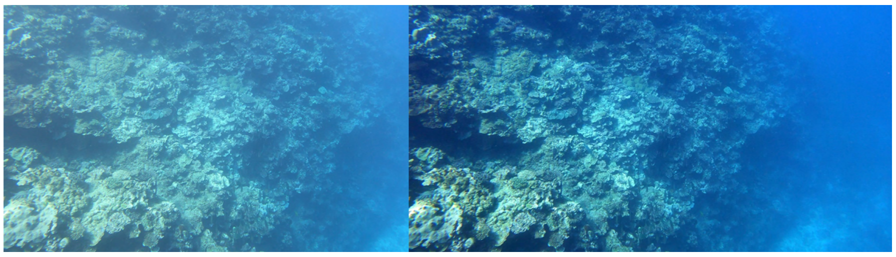

Figure 7. The original image in the dataset and the image with improved clarity

3.2 Faster R-CNN

We first use the Faster R-CNN combined with the training set to iterate 12 times and output the loss curve and the recognition accuracy mAP curve. We need to change the evaluation metric to F2 to combine COCOAPI with a Faster R-CNN to get F2. When identifying high-density starfish categories in a dense multi-size scene, because the starfish are different in size and have a close distance, it will cause the model to identify errors. Firstly we Replace the Faster R-CNN backbone network VGG16 with ResNext101. It helps to extract higher-level semantic information when extracting the features of starfish. Secondly, we add FPN in Faster R-CNN. It makes training more efficient through feature fusion and more suitable for multi-scale starfish detection. For further improvement, we changed the learning rate to 0.001 from 0.01 and the epoch from 12 to 24, which comprehensively improves the detection accuracy.

3.3 YOLOv5

First, we used the yolo original algorithm to recognize starfish, and we found the starfish was incomplete. We need to strengthen the network's feature fusion capability in this case. Therefore, I changed PANET to BiFPN.

After we changed it to BiFPN, too many parameters (46825853) would cause Cuda to go out of memory. Thus we changed the backbone to ghostnet (29223526). Next, we improved the evaluation metric to calculate the F2 score. Last but not least, we lessen the plot to output the total number of starfish and different colors bounding boxes. In this case, the improved Yolov5 algorithm solved the problem that the original Yolov5 may not contain all cirrus within the bounding box.

3.4 SVM

The algorithm uses VGG19 for feature extraction, the image will be resized to 244*244 pixels, and the VGG19 will return a feature matrix of size 1000*1. These features are entered into the SVM model to train the classifier and then used the fitted model to predict the area of the test image. The image will be split using the sliding window algorithm in the test process. The window size is using 16*16, and each movement is 8 pixels. Then using VGG19, again extracted the features for each cell and put them into the SVM model to predict. After predicting all the cells, the program will make them into a class matrix. Checking the predicted class in this matrix could merge the nearest cells into predicted boxes. Finally, calculate the values between the target box and each predicted box, and pick the highest one.

4. Experimental Results

The table listed below is the F2 score of our methods.

F2(IoU=0.5:0.95)		
	Original Algorithm	**Improved Algorithm**
YOLO	**0.688**	**0.703**
FASTER R-CNN	**0.644**	**0.653**
VGG+SVM	**0.00287**	**0.00287 (Average IoU = 0.0023)**

Table 1. F2 for all three methods

4.1 Yolo

For Yolo, our hardware desktop uses the windows platform. The graphic card is RTX 3080ti with 12G video memory. Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz CPU and 32G RAM. We trained 4919 pictures using python, 、pytorch、cuda+cudnn、unbuntu. The maximum epoch for yolo is 100, and the learning rate is. After we improved the yolov5 algorithm, our FPS can reach 166 when IoU is between 0.5:0.95.

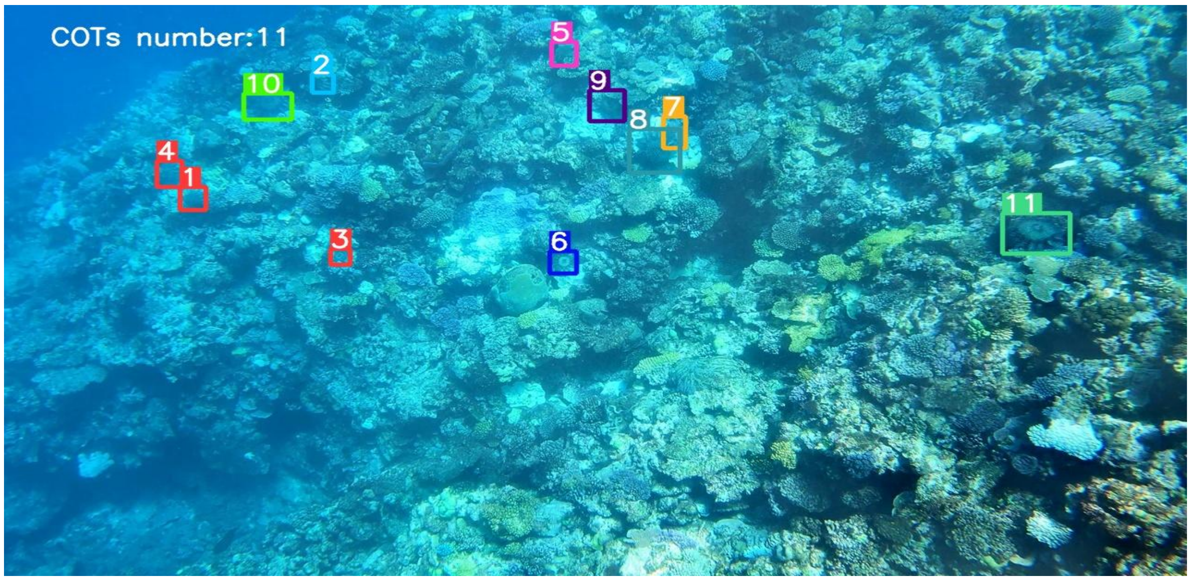

Figure 8. The output image of the original YOLOv5

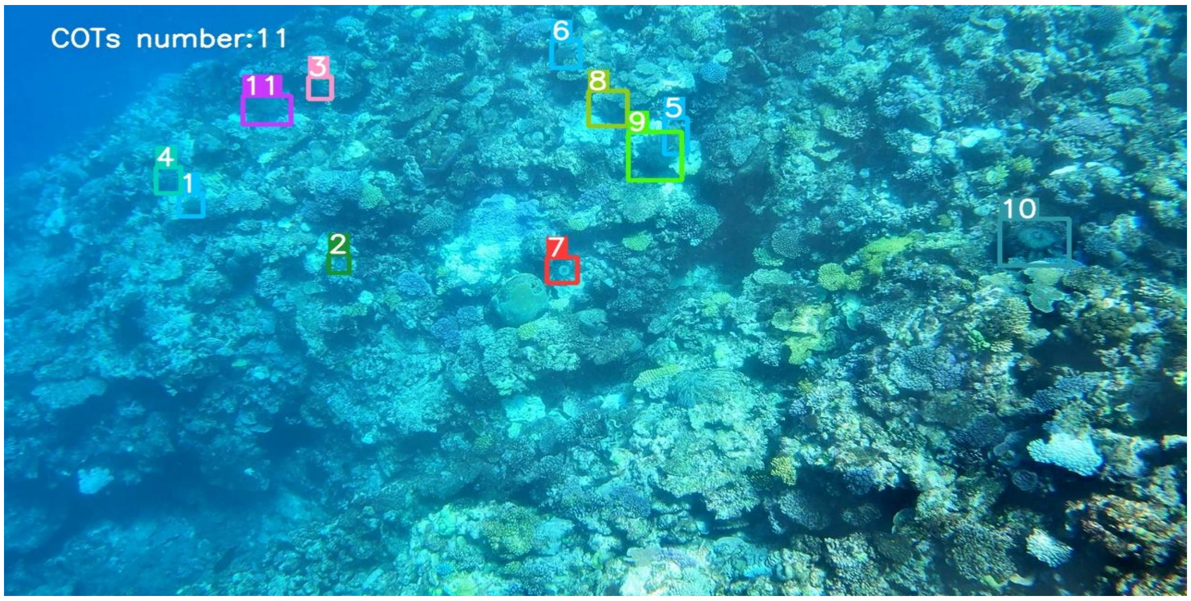

Figure 9. The output image of improved YOLOv5

4.2 Faster R-CNN

For Faster R-CNN. We used a computer with platform Windows server 2012, Intel Xeon Gold 5117 CPU, and NVIDIA Tesla V100 GPU to train 4919 starfish pictures. The software environment is Python 3.7, Cuda10. 2, and cudnn7.6.5. Experience based on Pycharm and Anaconda. The dataset contains 4919 starfish pictures and, after using labelImg to mark the starfish location and type information. We found there are 11897 marked outboxes. Our maximum epochs are 12, and the learning rate is 0.01. Each epoch will use a training set of 984 pictures to evaluate mAP, Original Accuracy F2 is 0.644, and our algorithm has improved it to 0.653 when IoU=0.5:0.95. The performance is the following. The first one is the result of the original Faster R-CNN (Figure

10), which number is 12, but the accurate result is 11, so the original Faster R-CNN has room to improve. The second one is the result of improved Faster R-CNN (Figure 11). The result is 11, which is an accurate result. It proves the improvement is practical.

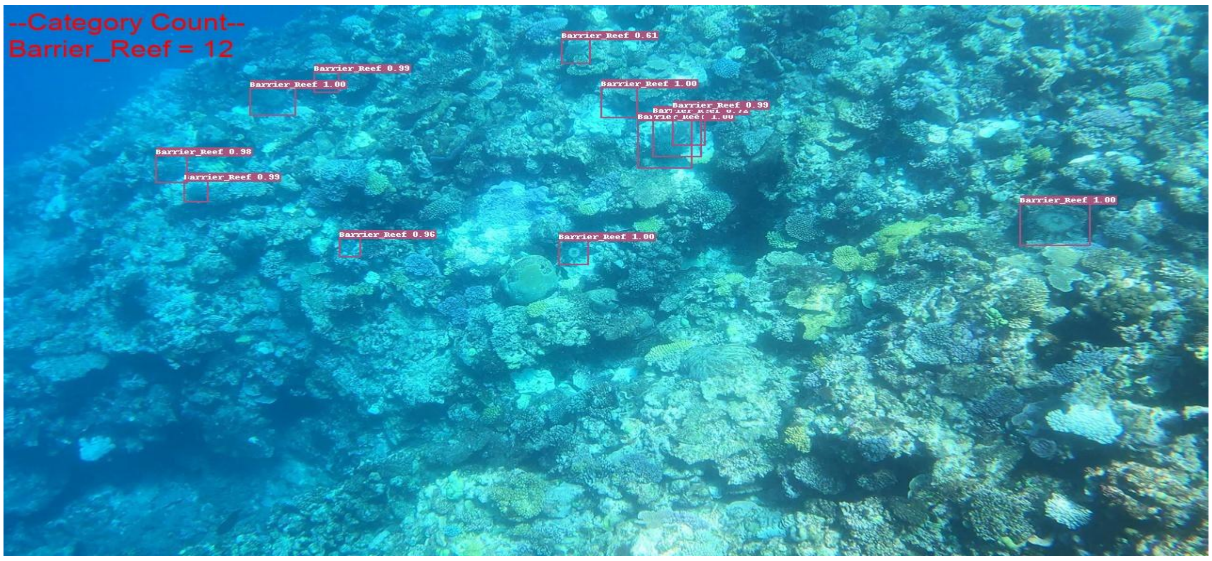

Figure 10. The output image of the original Faster R-CNN

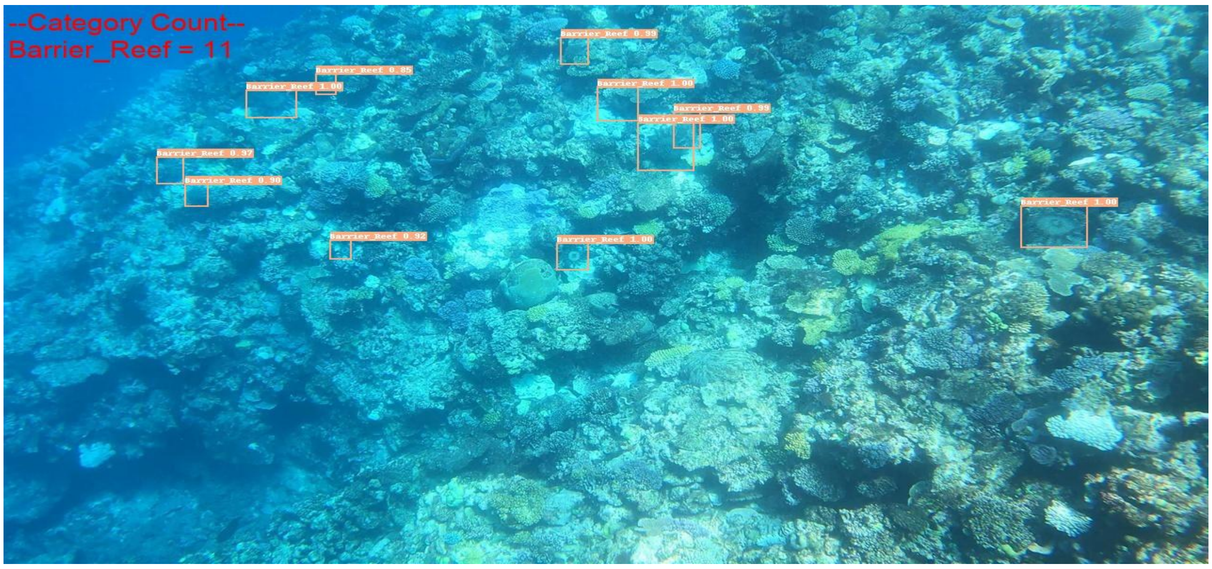

Figure 11. The output image of improved Faster R-CNN

4.3 SVM

The device is based on the 10th i9 and rtx3070, with 32GB RAM. The running environment is based on python 3.8.8, torch version 1.9.0, learn version 1.1.2, and OpenCV version 1.1.2. The training set with a size of 17518, and the test set contains 4380 images. The SVC, NuSVC, and OneClassSVM model in sklearn.SVM has been tried. The results are relatively small.

Figure 12. The output image of SVM

5. Discussion

In this project, we used three methods to train the datasets to find the most suitable method. Yolo works the best for sure, Faster R-CNN also works great, but there is still a distance from the result that Yolo worked out. SVM still get a long way to go. The reason for that is the Yolov5 uses many different times to detect different sizes of objects respectively. Faster R-CNN [13] generates region proposals by using a novel RPN. Unlike traditional algorithms like selective search, Faster R-CNN saves time. SVM has high efficiency in high-dimensional spaces.

The disadvantages of each method also are apparent. Yolov5 will lose many positions, making it difficult to detect small objects. Faster R-CNN's network takes much time to converge because all samples are correlated. SVM is unsuitable for large datasets and performs terribly if the dataset has more noise.

YOLO's running speed is 166, and its accuracy is F2=0.703 when Iou=0.5:0.95. Faster R-CNN's average FPS is 3.2, and its accuracy is F2=0.653 when Iou=0.5:0.95. SVM takes approx 5 mins to implement a picture and result in output, and its accuracy is F2=0.00287. The rubric reason for that is that those pictures are highly similar, which is difficult for SVM to detach, and VGG has errors while doing the resize work. In this case, we can see that Yolo is our best choice.

6. Conclusion

For this project, I used and improved Yolov5, a Faster R-CNN algorithm, to make it more accurate, and we also improved its efficiency. We also developed an algorithm on our own, which is VGG+SVM.

Yolov5 works excellently. Its accuracy F2 is high, and it is efficient. Faster R-CNN is also significant because its F2 can reach 0.653 when IoU=0.5:0.95. SVM did not work because its algorithm is unsuitable for this dataset.

We will recommend using Yolo for future work for yolo's incredible performance. Yolo's high efficiency and accuracy make it the best real-time object detection system. For future work, We would like to add the DCN function to detect starfish in different sizes to improve accuracy. Faster R-CNN. Furtherly, by replacing sift+bow with selective search, SVM can extract more typical features for training. For Yolov5, we can add the CBAM mechanism to prevent the original network from having no attraction problem. In this case, we can add more starfish features to improve accuracy.

Reference

[1] K. He, X. Tang, and J. Sun, “Single Image Haze Removal Using Dark Channel prior,” *IEEE Xplore*. [Online]. Available: https://ieeexplore.ieee.org/document/5206515. [Accessed: 08-Nov-2022].

[2] “Introduction,” *YOLOv5 Documentation*. [Online]. Available: https://docs.ultralytics.com/. [Accessed: 18-Nov-2022].

[3] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, “TPH-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios,” *TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios – arXiv Vanity*. [Online]. Available: https://www.arxiv-vanity.com/papers/2108.11539/. [Accessed: 18-Nov-2022].

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *arXiv.org*, 22-Oct-2014. [Online]. Available: https://arxiv.org/abs/1311.2524. [Accessed: 08-Nov-2022].

[5]R. Girshick, “Fast R-CNN,” *arXiv.org*, 27-Sep-2015. [Online]. Available: https://arxiv.org/abs/1504.08083. [Accessed: 08-Nov-2022].

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *arXiv.org*, 06-Jan-2016. [Online]. Available: https://arxiv.org/abs/1506.01497. [Accessed: 08-Nov-2022].

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” *arXiv.org*, 19-Apr-2017. [Online]. Available: https://arxiv.org/abs/1612.03144. [Accessed: 18-Nov-2022].

[8] D. K, “Top 4 advantages and disadvantages of support vector machine or SVM,” *Medium*, 26-Dec-2020. [Online].Available:https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107. [Accessed: 18-Nov-2022].

[9] “Flow diagram of support vector machine (SVM) image processing ...” [Online]. Available: https://www.researchgate.net/figure/Flow-diagram-of-support-vector-machine-SVM-image-processing-classification-algorithm_fig1_334461329. [Accessed: 17-Nov-2022].

[10] G. Boesch, “VGG very deep convolutional networks (vggnet) - what you need to know,” *viso.ai*, 05-Dec-2021.[Online].Available:https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/#:~:text=VGG%20stands%20for%20Visual%20Geometry,ground%2Dbreaking%20object%20recognition%20models. [Accessed: 18-Nov-2022].

[11] "Illustration of the network architecture of VGG-19 model: Conv means ..." [Online]. Available: https://www.researchgate.net/figure/llustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means_fig2_325137356. [Accessed: 17-Nov-2022].

[12] "Tensorflow - help protect the Great Barrier Reef," *Kaggle*. [Online]. Available: https://www.kaggle.com/competitions/tensorflow-great-barrier-reef/data. [Accessed: 08-Nov-2022].

[13]A. F. Gad, "Faster R-CNN explained for Object Detection Tasks," *Paperspace Blog*, 09-Apr-2021. [Online]. Available:https://blog.paperspace.com/faster-r-cnn-explained-object-detection/#:~:text=Faster%20R%2DCNN%20is%20a%20single%2Dstage%20model%20that%20is,vector%20from%20each%20region%20proposal. [Accessed: 18-Nov-2022].