Protecting Barrier Reef from Outbreak of COTS

Purvam Sheth, Nihar Patel, Vedant Thakore, Dhairya Purohit

***Abstract*—The outbreak of COTS i.e. Crown Of Thrones Starfish causes great harm to the coral reef of the surrounding region. One such scenario took place at Great Barrier Reef, Australia. In order to protect the aquatic life from COTS which eats up the faster growing corals and significantly damages the reef and to make sure of the sustained development of Barrier Reef, we should have a keen check over the population of 'COTS' in the entire region. Manual processes can be futile in marine life over a large area. The problem can be solved by using Objection Detection a key aspect of Computer Vision. Contributing to the fact in such case we need more precision in terms of timely detection than in terms of accuracy. Referring to the current state-of-art models for real time we tried implementing the novel Yolov5 model both small and medium but the results in terms of medium were quite remarkable, Yolov5 is the most powerful objection detection algorithm at present, to better apply it in the actual environment, especially in the supervision of COTS. Implementing it we obtained accuracy nearly....**

***Index Terms*—COTS, Coral Reef, Object Detection, YOLOv5, CNN**

I. Introduction

Computer vision technology uses a variety of imaging systems instead of visual organs as input means, using computers to replace the brain to complete the processing and interpretation of visual information. With the continuous development of computer vision technology, we can identify objects under complex conditions even when human vision fails. One such problem identified here is detection of COTS.

The problem of COTS detection has been studied since past few years as their outbreak causes problem for the growth of coral which in turn affects the aquatic life. COTS are crown of thrones starfish that are found in Great Barrier Reef and known for their growth rate and help to the coral reef. The main purpose is to maintain the equilibrium in the growth of corals, they tend to eat the fast growing corals making it easy for the slow growing corals to cop on to. This also helps in enhancing the diversity of the coral reef.

But if there is an outbreak of COTS in specific region they tend to eat smaller corals too making it unsustainable as they can grow upto 80 cm. The starfish are quite difficult to detect in underwater environment and it becomes a tedious job to manually identify them and locate the place of outburst. Here, the autonomous system comes in action. In order to make sure the sustained development of Barrier Reef, we should have a keen check over the population of COTS in the entire region. Manual processes can be futile in marine life over a large area, so with the help of Computer Vision we will make process a lot easier and accurate by creating Object detection model, which can be trained from underwater videos of coral reefs [1]. This helps the researchers and the department of tourism to maintain the Great Barrier Reef and develop it as a natural spot.

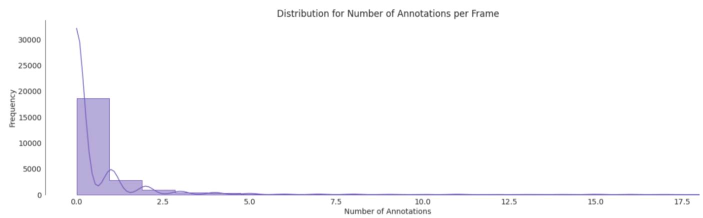


Fig. 1: Distribution for Number of annotaitons

The data set is obtained from Kaggle Challenge on COTS detection. Each data set contains videos with video id frame number and sequence, sequence frame and annotation details for training data while for test data it is hidden. There are total 3 videos. 1 video is split into 4 sequences, while the other 2 videos are split into 8 sequences each and for such data we have to look for better object detection architecture. Here, the data-set contains frame of which nearly 80% have no annotation as shown in Fig. 1, so it becomes difficult to train the model. Moreover, it becomes difficult to train models with such data set. Instead of working on complex model we decided to work on quicker and less complex models. Identifying the starfish in real time is an crtitical task, so the work proposes the use of Yolov5 for the same.

II. Literature Review

A. Object Detection

The object detection problem is the crux of computer vision. The detection is divided into 3 main parts; classification, detection and segmentation. The images are classified based on the labels, the objects are detected in the form of boxes around them and segmented from the image at pixel level with annotation provided. Here, the focus is on object detection rather than segmentation.

To detect the object with accuracy and working on real-time data is a challenging task to be looked at. There are basically two sections of object detectors; one stage detection and two stage detection. As the names suggest the one step are easier to implement and gives high speed inferences such as YOLO and SSD. Contradiction to them there are two stage detection that are more complex, takes more time but have high localization accuracy such as RCNN, RFCN and Mask RCNN [2]. Most

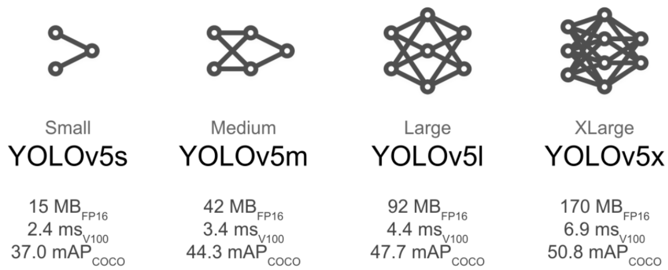


Fig. 2: Yolo Models

of the present literature studies RCNN for object detection as it provides better accuracy, but RCNN is not bulit for real-time object detection, as it uses selective search algorithm and takes huge time to train the network. Faster RCNN has high precision detection but significant disadvantage is less detection speed. Moreover, feature extraction is also not specifically prioritized in such case [3].

As the challenge here is to implement the object detection algorithm on a real-time data we have to use a single channeled forward pass network with single stage as major priority is less detection time.

B. YOLOv5

YOLO came into existence in mid 2020. Yolo stands for You Only Look Once. The existence of Version 5 does not affect the existence of previous versions as it came into existence during the heated presence of v4 as it uses darknet. YOLOV5 can achieve 140 FPS on Tesla P100 rapid detection, YOLOv4 is only 50 FPS. Meanwhile, the size of YOLOV5 is only 27 MB, while the size of YOLOv4 using Darknet architecture is 244 MB. YOLOV5 also has the same accuracy as YOLOV4 [4]. Yolov5 has all the advantages over previous models in terms of speed and accuracy [5]. The network architecture of Yolov5s has been shown in Fig. 2. The YOLO family of

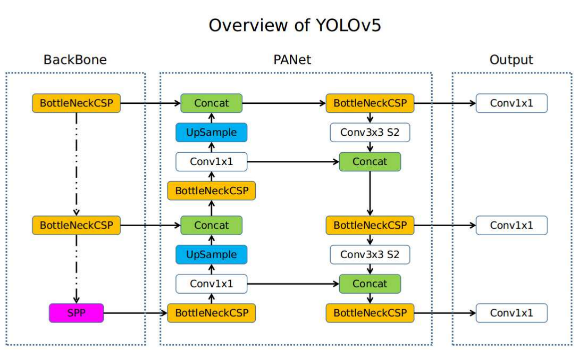


Fig. 3: Yolov5 Architecture

models consists of three main architectural blocks: Backbone, Neck, and Head. (i) YOLOv5 Backbone: it employs CSP-Darknet as the backbone for feature extraction from images

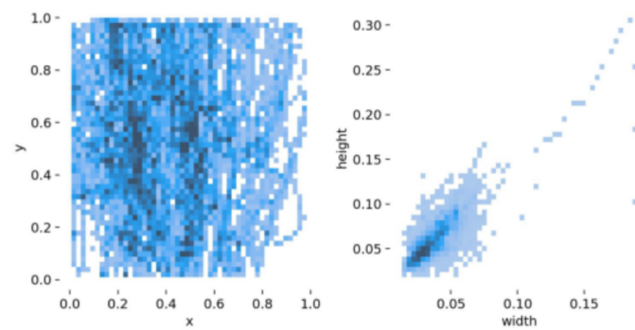


Fig. 4: Correlation matrix

consisting of cross-stage partial networks (ii) YOLOv5 Neck: it uses PANet to generate a feature pyramids network to perform aggregation on the features and pass it to Head for prediction (iii) YOLOv5 Head: it has layers that generate predictions from the anchor boxes for object detection

Yolov5 has various models from small(15 MB); ccan be used on mobile to extra-large(170MB); for ITS as shown in Fig. 3. The models are different on the bases of the network they provide, the larger the model gets the more complex it becomes and in turn increases the prediction time. We have compared v5s and v5m in terms of true positives and compared the results.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1\ score = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} \tag{3}$$

III. Implementation

Here, for better accuracy and fast detection we have implemented YOLOv5 (You Only Look Once) as it is fast and detect the object just at one look so we can work on real-time data-set [6]. YOLOv5 is single CNN structure. It requires only one forward path. It is capable to process images at 45 - 155 fps(frames per second). Here, the training data is in frames so we need not to worry for fps. It outperforms the RCNN and Faster RCNN in such cases.

YOLOv5 has many models based on the network of neurons making it more complex and involving larger number of parameters. In our case, we have implemented YOLOv5s with 37.2 mAP. Each video is divided into sequence frames ranging from 71 frames per sequence all the way to 3,000 frames per sequence. The target variable is annotation as most frames (80%) have no annotation they are simply discarded. The data-set consists of 3 videos having total of 23000 frames of which only 5000 are annotated as shown in Fig. 6. The implementation process consists of three processes;

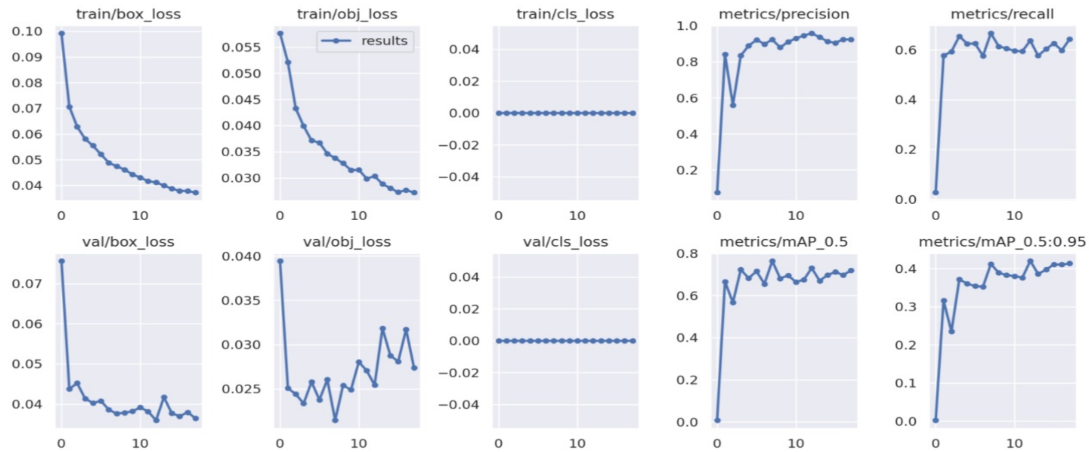


Fig. 5: Evaluation Matrices

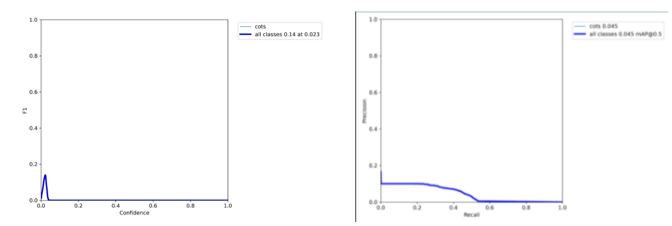
Fig. 6: Training data set

TABLE I: Simulation Parameters

Parameter	Value
Epoch	18
Batch Size	12
iou_t	0.5
Learning Rate	0.01
Box Loss Gain	0.05
Scale and Flip	0.5

Residual Blocks, Bounding Box Regression and Intersection Over Union (IOU). The residual blocks divides images in various SxS dimension grids, then bounding box highlights the images from an object and then the IOU describes how the boxes overlap. Yolov5 by default takes input of 640x640x3 and provides 3 outputs 20x20x255, 40x40x255 and 80x80x255 from the head.

The image is split in a grid that has the same dimension for each "tile". We add the bounding boxes that identify each object. The bbox has the following format: [width, height, class, bx, by], where [bx, by] represents the center of the object, height and width of the object and class notation. Intersection Over Union: this technique is used so the bounding box "catches" the object fully (and doesn't leave any part of



(a) (b)

Fig. 7: Previous Results

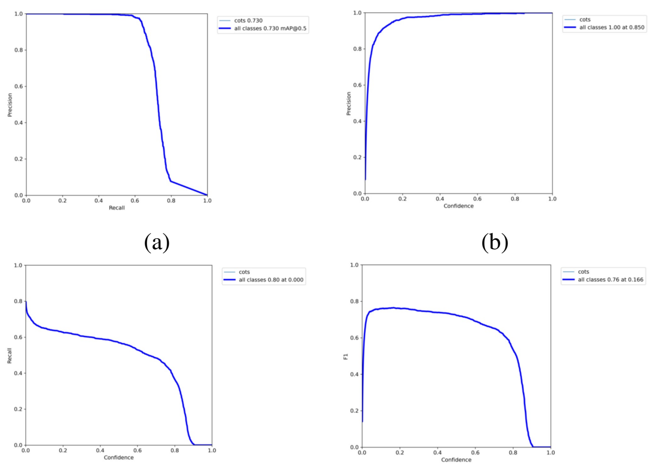
it uncovered, neither it is too large for the object). The IOU=1 if the predicted and actual box are identical. The COCO annotation boxes are converted to YOLO boxes decreasing

the number of points required to present the box. So, the data is prepared, images and labels are copied to YOLO and model is trained.

The training shows the correlation between the x, y coordinates the height and the width of the image. The class distribution and instances at which the COTS occurs. The images are distributed in batches of size 12. Using batch normalization we can make learning better. The rest of the parameters are shown in Table 1.

IV. Results

Currently, we have work on the data set as it contained videos. Pre-processing the data annotating the images and augmenting images to preform different operations like Flip, Scaling, Translation, shearing and Rotation on the image to check the detection accuracy of the model for increase in dataset. The annotations in COCO format are converted into two point format.The results shown are for, YOLOv5m with size = 500, batch size of 12 and for 18 epochs compared to previous Yolov5s with batch size of 8 and for 3 epochs. The annotations were quite promising. Fig . 4 shows the correlation between x, y, height and width of the images or label correlogram. Fig. 5 shows different evaluation matrices



(a) (b)

(c) (d)

Fig. 8: Results

for the model. The validation loss is significantly reduced and the mAP for threshold of 0.5 to 0.95 has been considered and precision has also been increased in comparison to previous small model. Fig. 7 shows the previous PR curve and F1 curve we obtained while implementing Yolov5s. Fig. 8 shows different results 8(a) show the PR (Precision Recall) curve for the model, 8(b) Precision Confidence Curve, 8(c) shows Recall Confidence curve and 8(d) shows the F1 Confidence curve. All the results justifies better results than the previous work. As shown in eq. (1),(2) and (3) the values are closely related to the ability of model predicitng true and false positives and negatives.

Fig. 9 shows the confusion matrix of the model. It is an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning. The diagonals show the values of true positive the model obtains true positive value of 0.62 while the false positives are 0.38 compared to the Yolov5s that had true positive value 0.29. The model took almost 5 hours to train on Kaggle we previously tried implementing it on collab but the session automatically stopped after around 6 hours at 11th epoch so we planned on implementing it on kaggle with GPU. Working on GPU we had TESLA P100 with 2 CPU cores and 13 GB RAM and it gave us around 55 fps at full resolution. Overall results were quite promising on real time data and efficiently detected the COTS.

V. Conclusion

This work puts forward the scope of using Yolov5 for the prospect of real time object detection which in this is detecting COTS, Crown of Thrown Starfish. The camera must be passed under water to get the videos that can be split into frames for detection. The experimentation on Kaggle data-set shows an accuracy of nearly 62% an precision of about 90% for Yolov5m which outperforms othe classical models in terms of detection time and accuracy. The limitation of newer data-set was an issue as we had restricted amount of video data, and annotation was not possible. There are newer models of Yolo available like YoloR and YoloX the future work could be extended onto them.

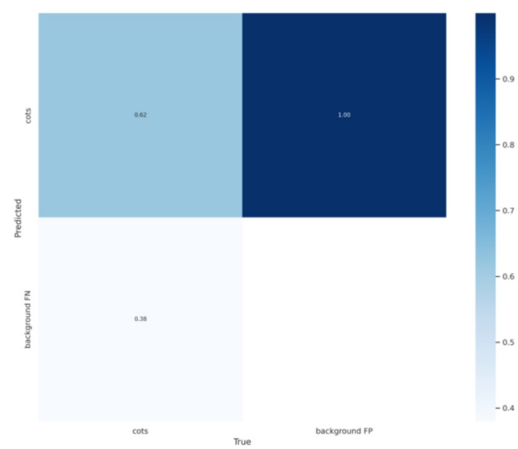


Fig. 9: Confusion Matrix

References

[1] J. Liu, B. Kusy, R. Marchant, B. Do, T. Merz, J. Crosswell, A. Steven, N. Heaney, K. von Richter, L. Tychsen-Smith *et al.*, "The csiro crown-of-thorn starfish detection dataset," *arXiv preprint arXiv:2111.14311*, 2021.

[2] W. Wu, H. Liu, L. Li, Y. Long, X. Wang, Z. Wang, J. Li, and Y. Chang, "Application of local fully convolutional neural network combined with yolo v5 algorithm in small target detection of remote sensing image," *PloS one*, vol. 16, no. 10, p. e0259283, 2021.

[3] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved yolov5," *Remote Sensing*, vol. 13, no. 9, p. 1619, 2021.

[4] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Gui, and W. Wang, "Face mask recognition system with yolov5 based on image recognition," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, 2020, pp. 1398–1404.

[5] U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, no. 2, p. 464, 2022.

[6] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and automated detection of surface knots on sawn timbers using yolo-v5 model." *BioResources*, vol. 16, no. 3, 2021.