Coral Reef Detection and Analysis

GROUP -12

Sumanth Reddy Alluri, Amir Fleurizard, Joshua Good

Contents

Motivation and Problem Statement

Motivation and Problem Statement :-

- Coral reefs are highly diverse ecosystems, supporting more species per unit area than any other marine environment.

- Human-driven activities like overfishing, destructive fishing practices, shipping, and pollution further degrade coral reef ecosystems.

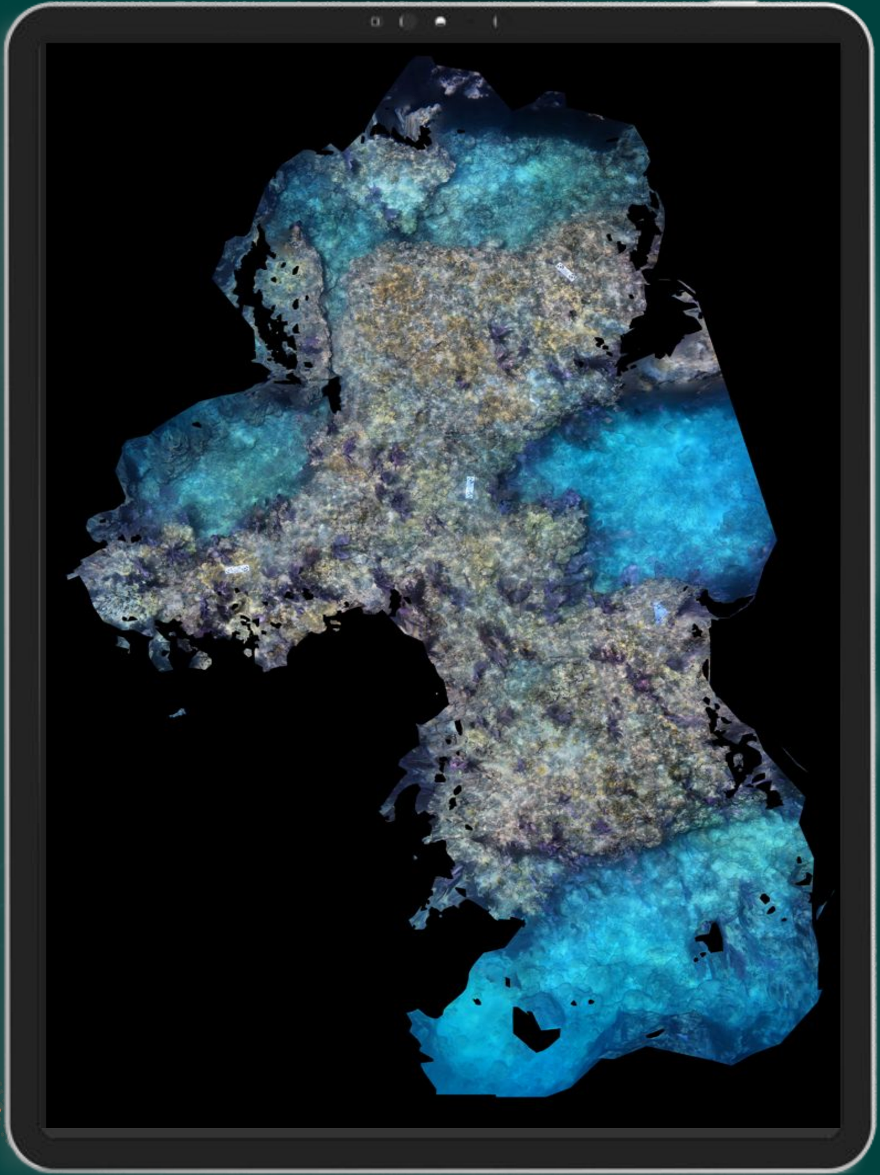
- Restoration efforts are underway, but measuring progress is difficult and time-consuming.

- Innovative and automated approaches can enhance the accuracy and efficiency of measuring coral growth.

- By improving measurement capabilities, we can better understand reef health, implement targeted interventions, and ensure long-term resilience for coral reefs.

- In this project, our primary goal is to develop a system that automates the detection and classification of corals.

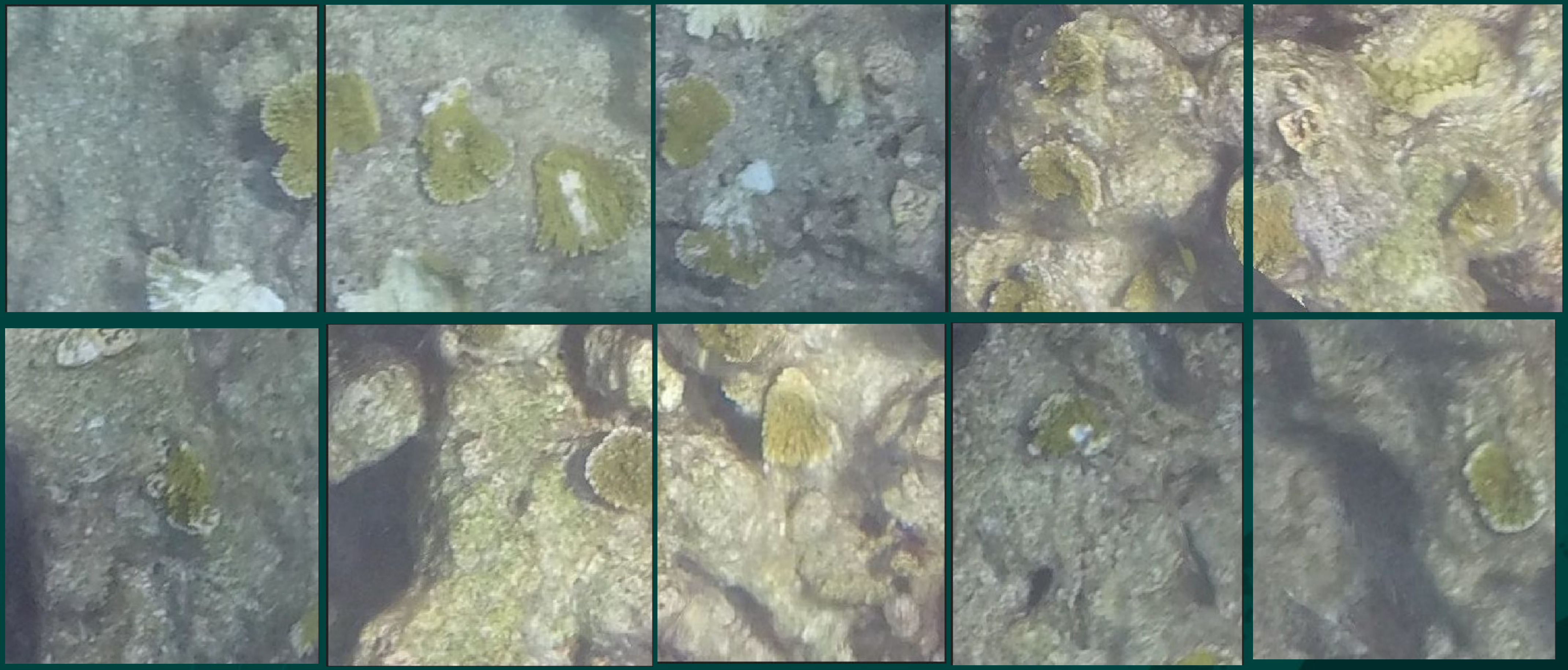
Dataset Overview



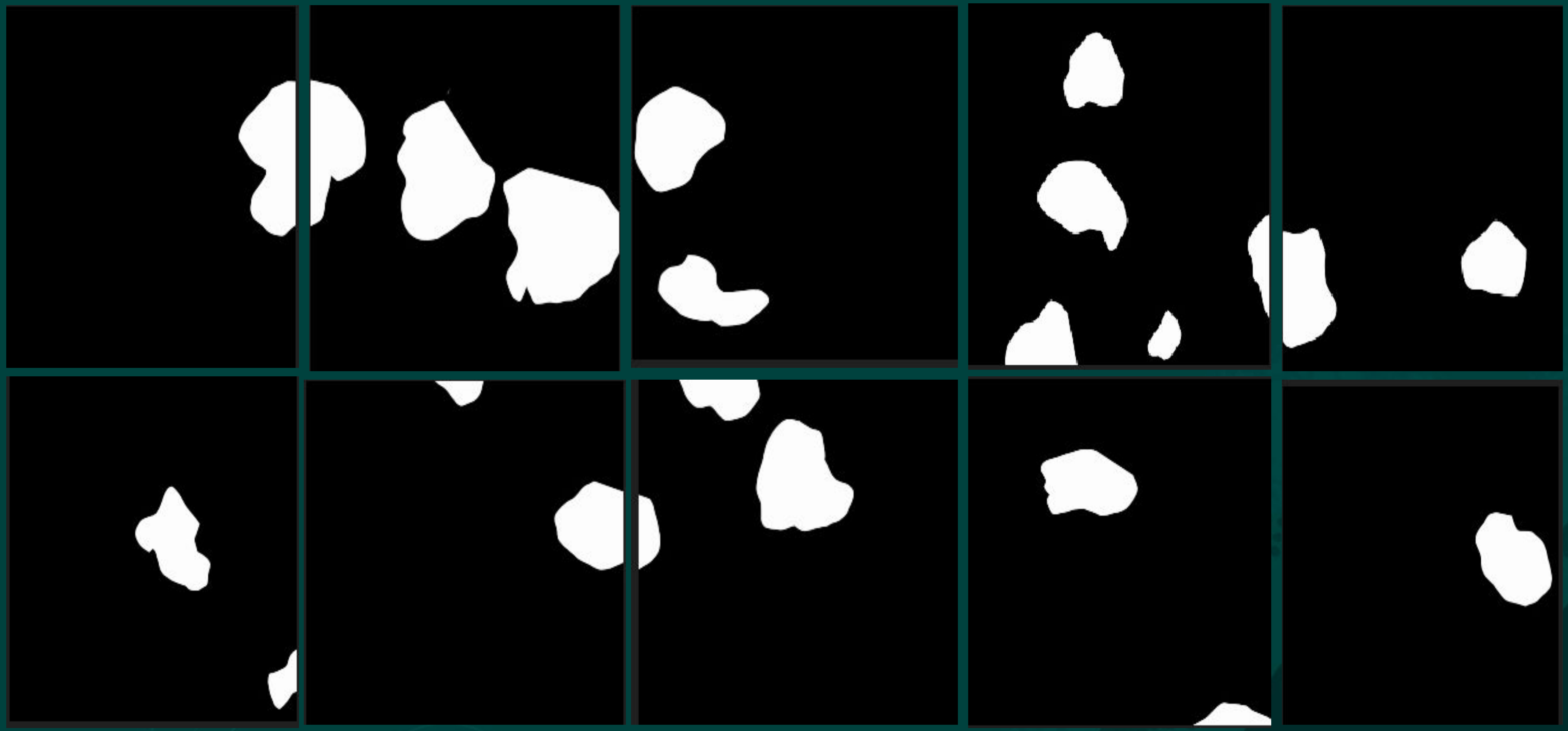
Dataset Overview:

- 16 Mosaic Images
- 16 Mosaic Keys
- Patch size: 256
- Useful Images: 4856

Patched Mosaic Images :-



Patched Keys Images



Methodology

YOLO V7

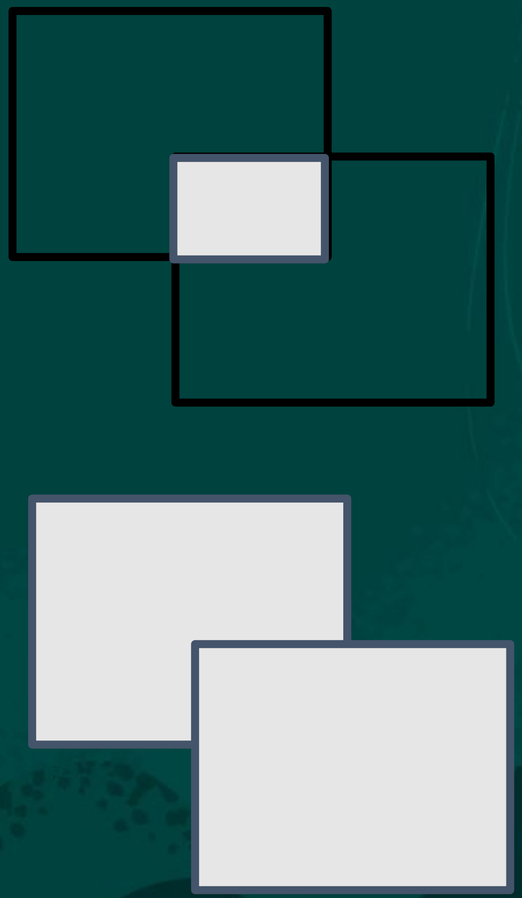
- “You Only Look Once” One-shot object detection
- Upgraded from the currently-used YOLO V5
- Provides a stronger, more efficient network architecture
- More accuracy, higher quality loss function, increased label assignment and training efficiency
- Computationally cheaper than other models for deep learning
- Effective with small datasets

YOLO V7 Compared with state of the Art

- YOLOv7-tiny-SiLU is 127 fps faster and has a 10.7% higher AP.
- YOLOv7 achieves a frame rate of 161 fps with a 51.4% AP.
- PPYOLOE-L achieves a frame rate of only 78 fps with the same AP.
- YOLOv7 uses 41% fewer parameters than PPYOLOE-L.
- YOLOv7-X is 31 fps faster than YOLOv5-X (r6.1) with similar scale.
- YOLOv7-X reduces parameters and computation by 22% and 8% respectively compared to YOLOv5-X, while improving AP by 2.2%.

IOU

- Intersection Over Union
- Calculates ratio of intersection to union of two bounding boxes
- Useful for narrowing in a bounding box to the object more precisely
- During training, this is used to compare our detected bounding boxes to the ground truth. A value closer to 1 implies our bounding box is accurate
- IOU threshold is commonly utilized to distinguish between a true positive detection and a false positive.
- For instance, if the IOU surpasses a predetermined threshold (e.g., 0.5 or 0.75), the detection is identified as a true positive. Conversely, if the IOU falls below the threshold, it is categorized as a false positive.
- IOU is an essential component for calculating other evaluation metrics used in object detection, such as Average Precision (AP) or Mean Average Precision (mAP). These metrics take into account precision, recall, and IOU to measure the overall performance of an object detection model.



IOU Calculation

To compute the IOU, we take the following process:

Locate our object
Determine the size and shape of our bounding box
Calculate the intersection of this box to our ground truth
Calculate the union of our boxes
Divide intersection/union to find our ratio

0<IoU<1

YOLO V5

- Training is faster on custom dataset.
- Slightly less accurate than than YOLO V7.
- Memory utilization is stable.
- YOLO V5 MAP (mean average precision) on the COCO dataset is 55.0.

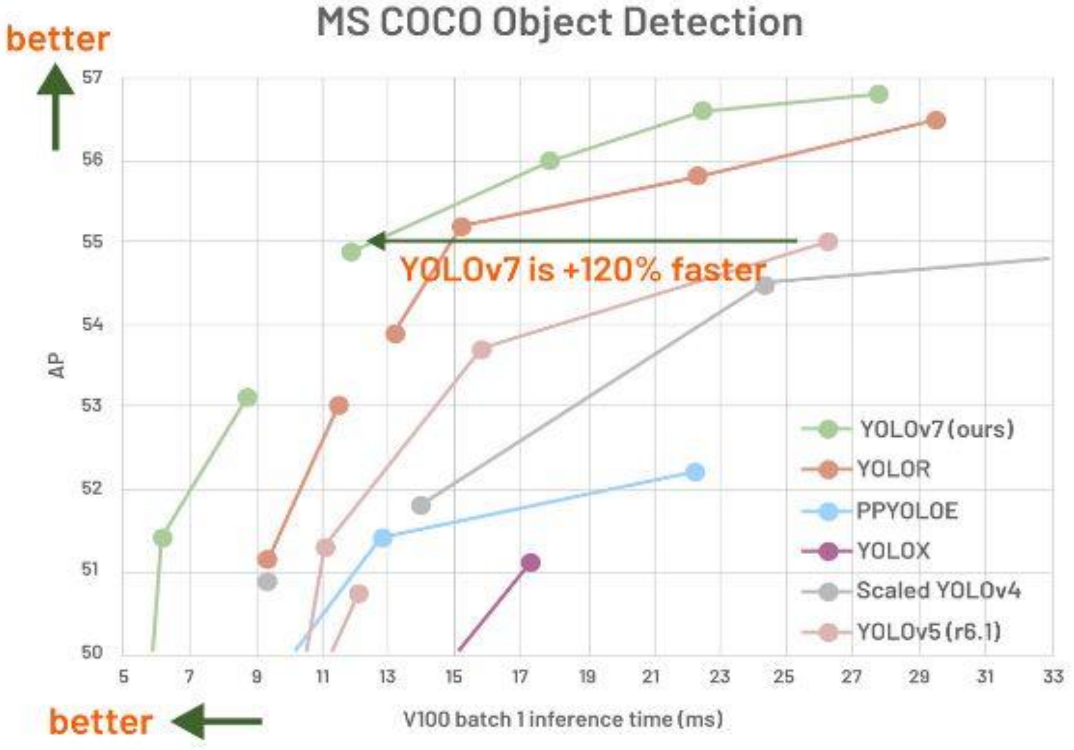
YOLO V7

- Training is comparatively slower on custom dataset.
- More accurate.
- Memory Utilization is Unstable
  .
- The MAP (mean average precision) of YOLOv7 on the COCO dataset is 56.8

YOLOv5

MAP: 55%

INCLUDED SEGMENTATION AS SECONDARY MODULE



YOLOv7

MAP: 56.8%

INCLUDED SEGMENTATION AS SECONDARY MODULE

YOLOv5 and YOLOv7 Accuracy Comparison

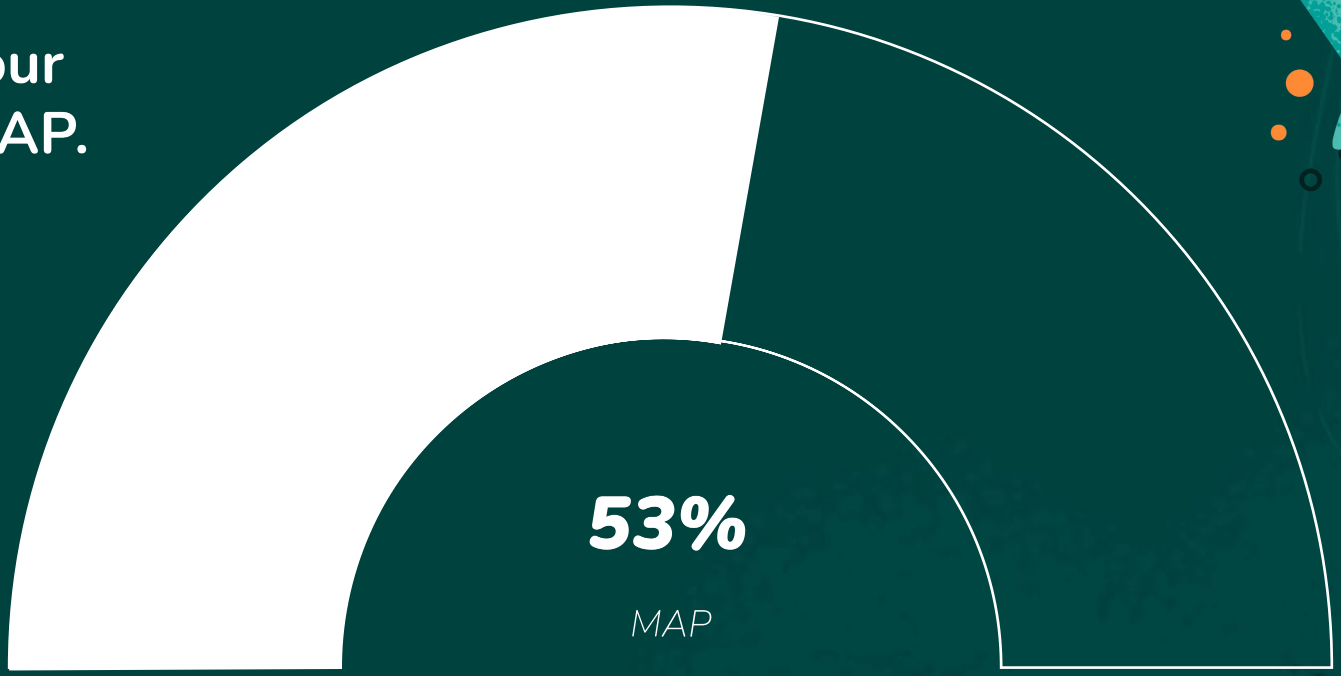
Results

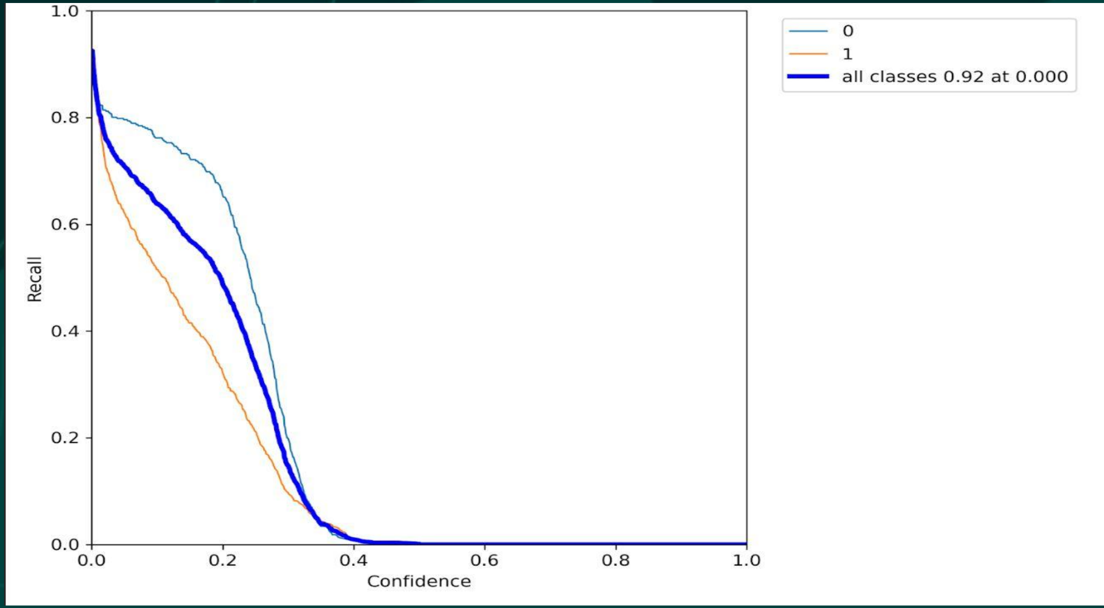
Results:

After applying YOLO V7 on our dataset, we achieved 53% MAP.

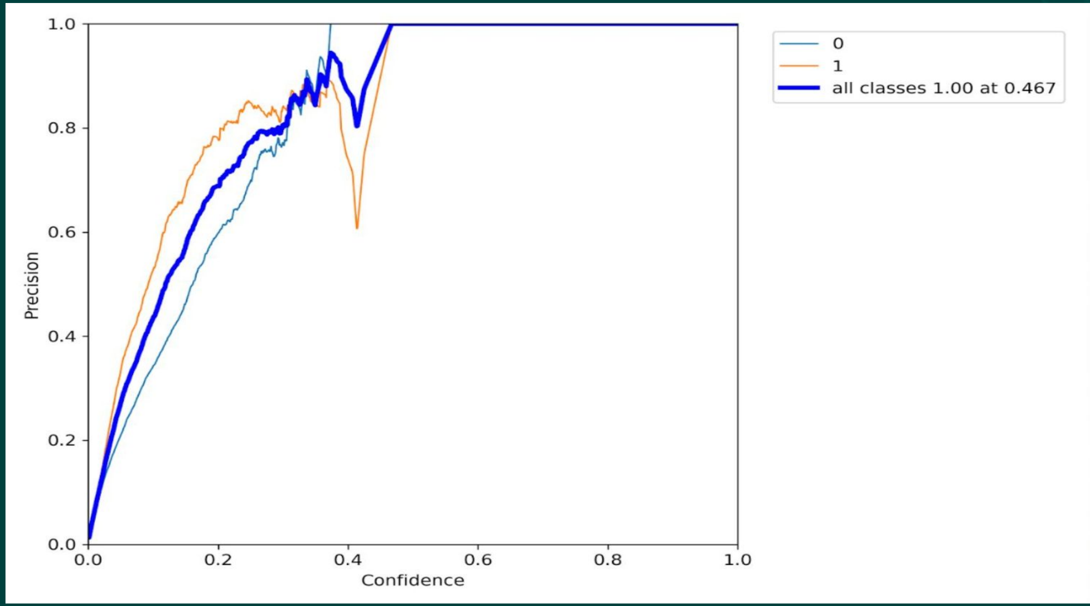
<u>HyperParameters:</u>

Image_size: 256
Workers: 4
Batch Size: 4
Epochs: 300

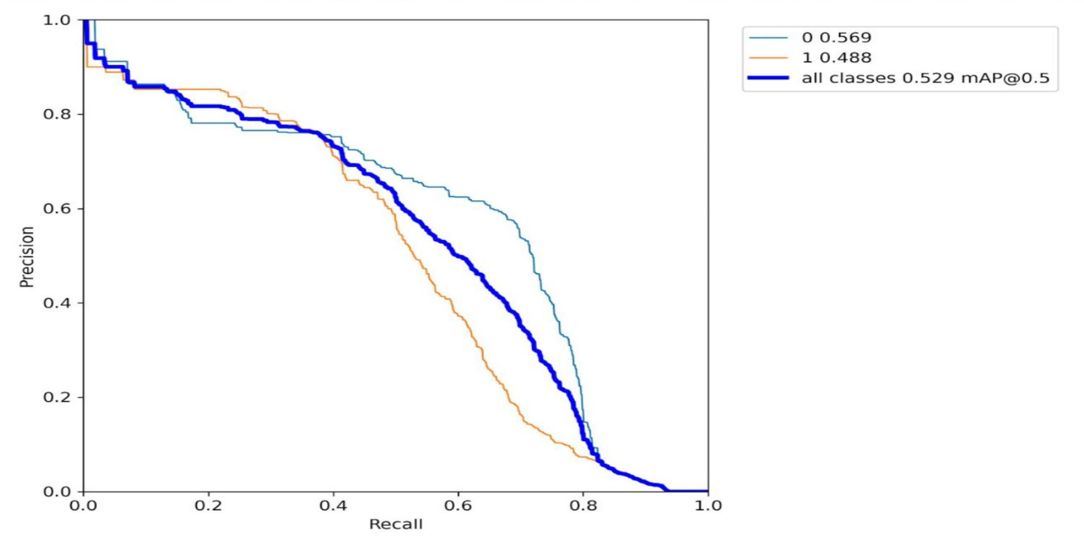




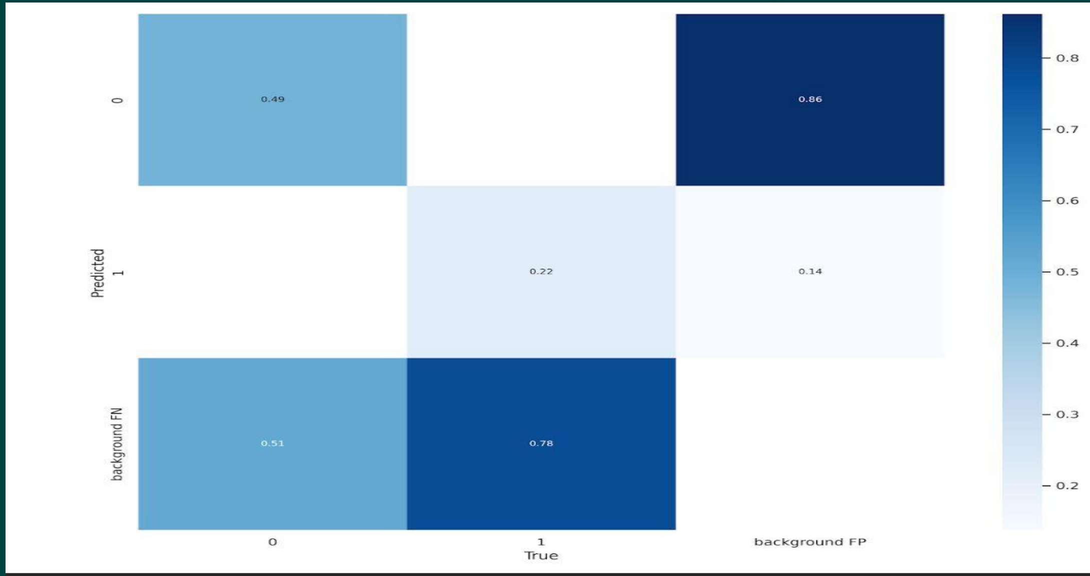
Recall
Confidence
0
1
all classes 0.92 at 0.000



Precision
Confidence
0
1
all classes 1.00 at 0.467



Precision
Recall
0 0.569
1 0.488
all classes 0.529 mAP@0.5



Predicted
True
background FN
background FP
0.49
0.86
0.22
0.14
0.51
0.78

Elkhorn Detected Images :-

Staghorn Detected Images :-

Problems Faced

Deformable DEtection TRansformer

- We began by using the Deformable Detection Transformer (DETR) as our implementation method.

- We encountered a problem during model training because we needed to convert the provided images into the COCO data format, which is a JSON format commonly used in object detection tasks.

- When we tested the model, we wrote the test.py file from scratch and looked at some online resources for reference.

- Although we were able to successfully implement the Deformable DETR model, we did not obtain conclusive results when using it to detect coral.

OutPut Of Deformable DETR

Conclusion

In conclusion, the objective of our project was to automate the detection of various types of corals. The results obtained from applying YOLO V7 were promising and demonstrated the potential for further improvement. As YOLO V7 is still undergoing experimentation and refinement, there is a possibility of achieving even higher accuracy in future iterations. Additionally, with access to better GPU resources, the model can be trained more effectively, leading to improved performance. Overall, this project lays the foundation for continued work and advancement in automating coral detection using YOLO V7.

Thank You

Any Questions!!!