TensorFlow - Help Protect the Great Barrier Reef

1st Lo Ho Chan
Faculty of Computer Science and Engineering
NSW, Australia
z5372602@ad.unsw.edu.au

2nd Yuqi Liu
Faculty of Computer Science and Engineering
NSW, Australia
z5377632@ad.unsw.edu.au

3rd Xinchen Zou
Faculty of Computer Science and Engineering
NSW, Australia
z53782402@ad.unsw.edu.au

4th Yang Wang
Faculty of Computer Science and Engineering
NSW, Australia
z5285515@ad.unsw.edu.au

5th Wanting Zhou
Faculty of Computer Science and Engineering
NSW, Australia
z5347036@ad.unsw.edu.au

***Abstract*—This paper proposes two developed models for image enhancement and object detection. To improve the detection accuracy, pre-processing is essential, sharpen kernel and Fast Nl Mean are used for image enhancement, while there may some noise in each image. In the starfish detection part, we implemented YOLO V5 and Faster R-CNN. Joint intersection (IOU) and F2 score are the standards to evaluate the model performance. In our project, we mainly focus on the comparisons between two models and Faster R-CNN with or without any image enhancement.**

***Keywords*—object detection, image enhancement, YOLO V5, Faster R-CNN, IOU, F2 score**

I. Introduction

At present, the analysis of object motion by computer vision can be decomposed into 1) Motion segmentation and object detection 2) Target tracking 3) Action recognition and behavior description. Object detection is the basic task of object motion analysis, at the same time, as the most basic task of target motion analysis, the effect of object detection directly affects the effect of follow-up tracking, action recognition, behavior description and other higher-level tasks.

The goal of this project is to develop an ideal method to accurately identify starfish in real-time by building an object detection model trained on underwater videos of coral reefs. Nowadays, researchers have explored object detection with many algorithms to cope with a large diversity over the past few years, however, it is still a very challenging task to perform accurate, complete, efficient, and reproducible recognition and analysis of the relevant image information due to the large volume and complexity. [1] In this task, there are several outstanding problems in COTs detection, for example, the COTs may hide under the surroundings such as rocks and fish, which may fail to be detected. Another problem is that the color of some COTs is like that of seawater, this may keep the starfish hidden in the sea.

In this paper, we will use the datasets from the Kaggle COTS detection competition, and the training set consists of three videos containing in total tens of thousands of images with corresponding manual annotations (bounding boxes around COTS objects). [2] We tried two models Faster R-CNN, which is one of the SOTA models for Object detection, and YOLO V5, which will be compared by using various image pre-processing e.g. sharp kernel, cv2.fastNlMeansDenoisingColored() to see the different outcomes the models achieve. According to the score of F2, we finally get the conclusion that YOLO V5 works better.

The rest contents of this paper are structured as follows. Section II reviews relevant techniques in literature, with the background of the methods we use. The analysis and implementation of the methods we use will be explained in Section III. Section IV is going to show the experimental results of the developed methods and evaluate the performance. A discussion of the results and method performance will be shown in section V. Last section VI concludes the accomplishments of this research and some further plans.

II. Literature review

Computer image recognition has been developed for decades, this technology is usually applied on lands, such as vehicle identification, classification of plants and animals, or face recognition which is now indispensable for each of us to use our mobile phones. On the contrary, the application of computer image technology in the ocean is still in its initial stage of development. However, as various underwater vision-related topics such as marine biology and marine archaeology have become more and more important to scholars in recent years, we are not ignorant of how-to better process images of underwater creatures. The processing and analysis of underwater images are particularly important for exploring and protecting the entire global environment and maintaining the balance of biodiversity in the water. This is especially urgent for a country like Australia, which is surrounded by the sea.

Although the underwater creature image is also taken by a professional camera, it does not restore 100% of the original appearance of the creature on land. This is mainly because the light is refracted and scattered at different depths in the sea,

which causes the deformation of the photographed object. [3] In addition, unlike blue and green light, red or orange light has a longer wavelength and is more easily and quickly absorbed by the water. Therefore, underwater images tend to have a blue or green tint.For these reasons, underwater images tend to have poor visibility, low definition, low contrast, and high chromatic aberration. Therefore, it is especially important to enhance the image before underwater image object detection.

Existing image enhancement methods can be broadly classified into four categories: specialized hardware-based methods, physical model-based methods, non-physical model-based methods, and deep learning-based methods. [4] The first hardware-based method generally improves the clarity of imaging by means of a free-rising deep-sea tripod. However, the cost is too high, and the contrast of the image is not greatly improved, so it is not suitable for most underwater scenes. The second method, based on physical models, relies on a series of a priori models that we calculate manually and then invert the underwater imaging to improve clarity. This method is not always effective because it relies on our hand-calculated prior models. So, if these prior models are wrong when inverting the underwater image, then instead of enhancing the image, we may make the image more blurred. The third method belongs to the traditional image enhancement methods, such as histogram equalization, contrast limiting and image fusion. These methods can directly change the pixel values of the image to achieve colour correction and contrast enhancement. [4] However, this method tends to over-enhance or over-saturate, so this method is also not applicable to underwater image enhancement.

Recently, many researchers have combined the advantages of deep learning with traditional image enhancement techniques to build a convolutional neural network (CNN) with multicolour spatial embedding to train underwater images. [5] The deep learning-based underwater image enhancement method not only gets rid of the limitation of equipment and does not require manual computation in advance, but also can alleviate the problem of overfitting, the biggest limitation is that sufficient high-quality training data of underwater images are required. Due to the open source and free nature of OpenCV, we can easily call various built-in computer vision functions to compare which pre-processing method can get better visual results. [6] Finally, we choose the Non-Local Means denoising function in OpenCV for image enhancement.

Support Vector Machine (SVM) based on a Histogram of Oriented Gradient (HOG) is a traditional target detection algorithm, often used for pedestrian detection. It has better generalization performance and is not easy to over-fit. However, traditional SVM can only do binary classification, and it becomes slow when encountering large-scale training samples. In addition, the disadvantage of HOG is that it is difficult to deal with occlusion problems, such as excessive human pose motion or object orientation change, and it is not easy to detect. So, this paper does not choose this model to detect starfish.

The R-CNN algorithm uses unsupervised selective search (SS) to recursively merge regions of the input image with similar colour histogram features and then generates multiple candidate regions. [7] The CNN is then used to perform high-level feature extraction, and finally, the extracted features are fed to the SVM classifier for classification. Since R-CNN needs to do feature extraction for each candidate region, it leads to a large amount of overlapping information in feature extraction, which is inefficient. The Fast R-CNN algorithm is an improvement of the R-CNN algorithm that uses FC layer instead of the previous SVM classifier and linear regressor for object classification and detection frame correction. [8] This has the advantage of forming a whole with the feature-extracting CNN, which greatly enhances the integration of detection tasks and improves computational efficiency, but still makes it difficult to achieve real-time detection. [9] The Faster R-CNN algorithm is further optimized for Fast R-CNN. The detection efficiency is greatly improved by using a region proposal network (RPN) instead of a selective search (SS). So, this paper tries this model to detect starfish.

The Faster R-CNN model has a clear division of labour between the RPN and R-CNN phases, [10] which brings an improvement in accuracy but is relatively slow and has not yet reached real-time in practical implementation. To improve this shortcoming, Redmon et al. [11] proposed the algorithm of YOLO back in 2015 and proposed the idea that subregions are responsible for detection, which greatly improves the detection speed without reducing the accuracy. Since its proposal, it has been continuously updated with new versions, from V2, V3, and V4 all the way to V5, V6, and V7. YOLO V5 has many different network architectures and is widely used for target detection in various fields. [12] For example, WenZe Fan et al. demonstrated in their experiments that the correctness of anomaly detection by using YOLO V5 for Chest Radiograph was 7.28% higher than that by using Faster R-CNN. [13] Mosaic data augmentation is also effective in solving the "small object problem", which is the most painful part of model training. [14] Therefore, in this paper, we choose to use YOLO V5 as another model to detect starfish and compare and analyze with Faster R-CNN.

In general, combined with the literature review and the characteristics of underwater images, we decided to use the Non-Local Means denoising function in OpenCV to achieve image enhancement, and then adopt Faster R-CNN and YOLO V5 algorithm for object detection.

III. Methods

In this paper, we broke the methods into three main steps, which are data preprocessing, image enhancement and model implementation.

A. Data Preprocessing

As all the images are captured from three underwater videos continuously, which have more than 23,000 images, and this is not realistic to read all images for the train-test-validation split, as it will generate an extremely large train-test-validation portion, and not all images have the COTs (the image has no annotation), as the figure (Fig 1.) shown below.

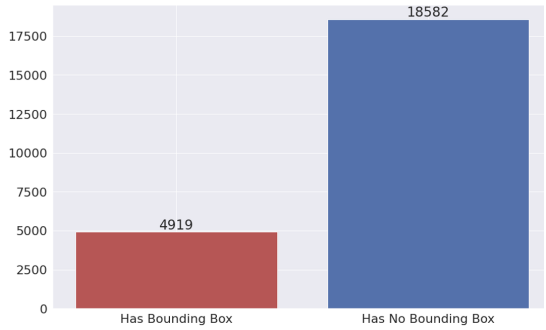

Fig. 1. The count of images with/without bounding box

After removing those images without a bounding box, there are only 4919 images for us to analyse. Then the proportion of train-validation-test is 60-20-20

B. Image Enhancement

As mentioned before, there are a few problems in COTs detection, one is that perhaps the starfish is blocked by the fish, as shown in Fig 2.

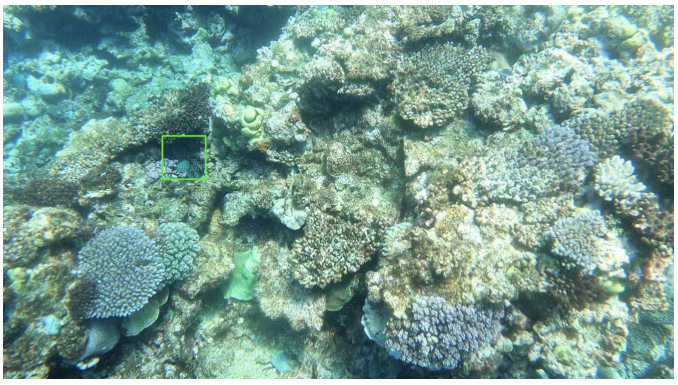

Fig. 2. The starfish is blocked by the fish

Another problem is that obviously, the given image has some sort of noise such as the colour of the starfish being like that of seawater, so this may be difficult to detect the starfish, see Fig 3.

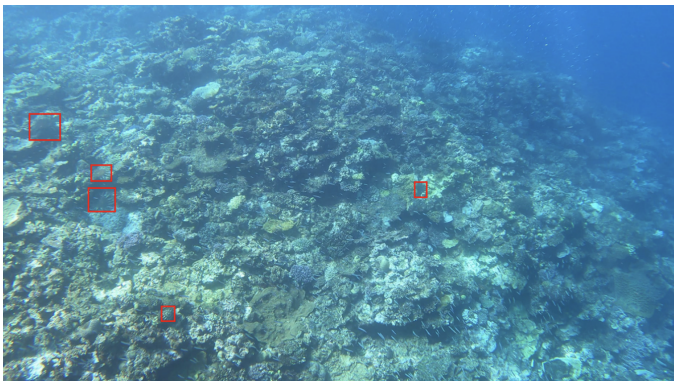

Fig. 3. The colour of starfish is so similar to the colour of the sea

To overcome these problems we mentioned before, we proposed two approaches to image enhancement one is using sharpen kernel.

- **Sharpen kernel**

In image processing, many filters (filter functions) use kernels, which means a set of weights, to decide how to calculate new pixels by using the pixels around a point, so the kernel can also be called the convolution matrix. The matrix can perform harmonic or convolution operations on pixels in a region, the kernel-based filter is usually called the convolution filter.

Suppose that there is a matric with a number from 0 to 255, and each number represents a number, as an image is 2-dimensional. A kernel is also a simple 2-dimensional matrix, and the 3*3 sharpen kernel is shown below:

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \tag{1}$$

A kernel works by operating on these pixel values using straightforward mathematics to construct a new image. The following figure is an example of kernel operation. [15]

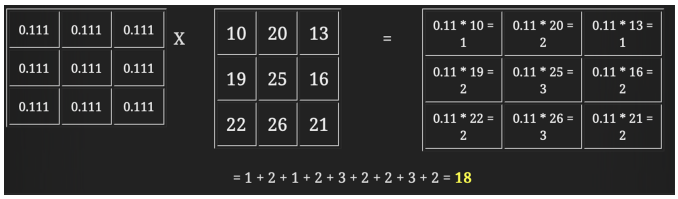

Fig. 4. An example of kernel operation

As each pixel is processed, a new image emerges based upon the calculated values.

- **Fast Non-Local Means Denoising**

The other method of image enhancement is called non-local means denoising, which is proposed by Baudes in 2005, this algorithm uses redundant information prevalent in natural images to denoising. The principle of the first denoising method was quite simple: replacing the colour of a pixel with an average of the colours of nearby pixels. [16] Different from the commonly used bilinear filter and median filter, which use the local information of the image to filter, it uses the whole image to de-noise, finds the similar areas in the image block as the unit, and then averages these areas, which can better remove the Gaussian noise in the image. The formula for filtering process of NL-Means is shown below:

$$NL\left[v\right]\left(i\right) = \sum_{j \in I} w\left(i,j\right) v\left(j\right) \tag{2}$$

Where discrete noisy image

$$\{v = \{v\left(i\right) | \, i \in I\} \tag{3}$$

, the estimated value

$$NL\left[v\right]\left(i\right) \tag{4}$$

, for a pixel i, is computed as a weighted average of all the pixels in the image.

$$w(i,j) \tag{5}$$

is the Euclidean distance between image patches centered respectively at i and j.

C. Model implementation

In this project, we implemented two different models.

- **Faster R-CNN**

In object detection tasks, R-CNN series models are very popular architectures. As R-CNN and Fast R-CNN have been released for object detection, these two models have played a significant role in computer vision research, which can reach very real-time rates. However, there is a test-time issue, that's the reason for the appearance of Faster R-CNN. As figure Fig 5. shown below, faster R-CNN can be divided into four main components. [17]

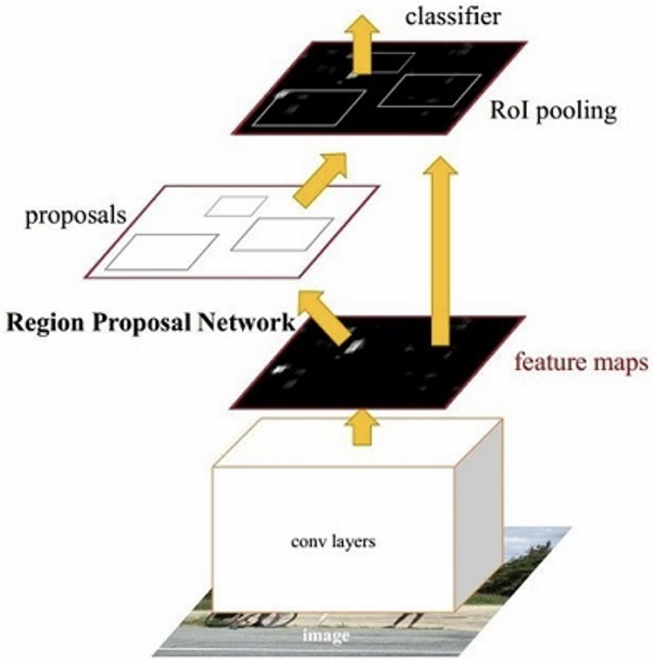

Fig. 5. The basic structure of Faster R-CNN

1. **Conv layers.** As a CNN network target detection method, Faster RCNN first uses a set of basic conv+relu+pooling layers to extract feature maps of images. The feature maps are shared for the subsequent RPN layer and the full connection layer.

2. **Region Proposal Networks (RPN).** The RPN network is used to generate region proposals. This layer determines whether anchors are positive or negative through softmax, and then uses bounding box regression to modify anchors to obtain accurate proposals.

3. **Roi Pooling.** This layer collects input feature maps and proposals, extracts proposal feature maps after synthesizing this information, and sends them to the subsequent full connection layer to determine the target category.

4. **Classification.** The proposal feature maps were used to calculate the category of the proposal, and the final exact position of the detection box was obtained again with the bounding box regression.

Fig 6. gives the network structure of RPN. As classic detection methods are time-consuming to generate detection boxes.

However, Faster RCNN abandons the traditional sliding window and SS method and directly uses RPN to generate detection boxes, which is also a great advantage of Faster R-CNN and can greatly improve the generation speed of detection boxes.

The RPN network is divided into two lines. The top one uses softmax classification anchors to obtain positive and negative classification, and the bottom one is used to calculate the bounding box regression offset to anchors. To get an exact proposal. The final Proposal layer is responsible for synthesizing positive anchors and corresponding bounding box regression offsets to obtain proposals while weeding out proposals that are too small or beyond the boundary. In fact, when the whole network reaches the Proposal Layer, it completes the function equivalent to target positioning.

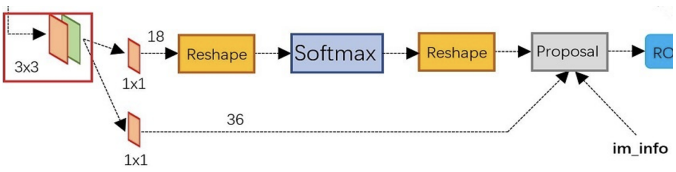

Fig. 6. RPN Network structure

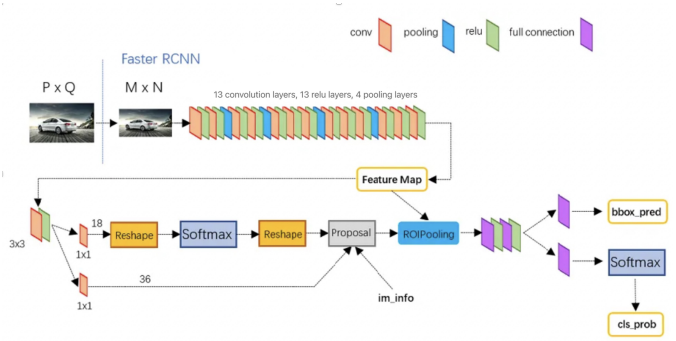

Fig. 7. Faster_rcnn_test.pt Network structure

Fig 7. shows the network structure network in VGG16 model.

The network for an image of any size P*Q can be clearly seen:

First, scale to a fixed size M*N, and then send the M*N image into the network.

The Conv layers include 13 conv layers, 13 relu layers and 4 pooling layers. RPN network first goes through 3*3 convolution, then generates positive anchors and corresponding bounding box regression offset respectively, and then calculates proposals.

The Roi Pooling layer extracts the proposal feature from feature maps through proposals and sends it into the subsequent full-connection and softmax network for classification (i.e. what object is the proposal).

- **YOLO V5**

YOLO, an acronym for "You only look once", is an object detection algorithm that divides images into a grid system. YOLO is one of the most famous targets detection algorithms for its speed and accuracy. The YOLOv5 network is one of the results of the YOLO family of algorithms. Shortly after the release of YOLOv4, Glenn Jocher released YOLOv5 using the Pytorch framework. Its network structure is very similar to the YOLOv4 network, but it converges faster and runs faster than YOLOv4.

The YOLOv5 network structure consists of three main parts: backbone: New CSP-Darknet53; neck: SPPF, New CSP-PAN; head: YOLOv3 Head. The network structure is shown below, there are five specific pre-training models, YOLOV5n, OLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5n model has the smallest structure, the shallowest depth, the fastest running speed and the lowest accuracy. The other four network structures gradually become deeper and more complex, with increasing accuracy, but slower processing speed.

YOLOv5 uses mosaic data enhancement, adaptive anchor box operation and random affine (rotation, scaling, panning, and clipping) to enhance the input image. The backbone parts are the Focus structure and the CSP structure, respectively. focus improves the training speed by slicing the input image, thus reducing floating point operations. yolov5 uses two CSP (Wang et al. 2020) structures: CSP1 for the backbone layer and CSP2 for the neck. The necks are SPP-net and FPN+PAN structures, respectively, which enhance the feature fusion effect of the network. In addition to the improvements described above for YOLOv5 and the previous four versions, YOLOv5 versions are constantly being updated. In this paper, we use YOLOv5s v6.0 version, which has some minor changes compared with the previous versions: 1. Replace the Focus structure with 6x6 Conv2d (more efficient); 2. Replace the SPP structure with SPPF (more than double the speed).

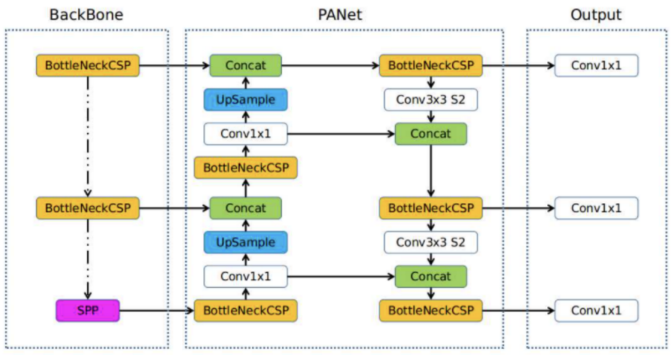

Fig. 8. Overview of YOLO V5

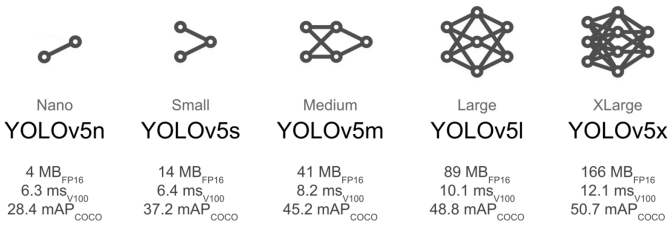

Fig. 9. Five specific pre-training models of YOLO V5

In addition, YOLOv5 has made some improvements in the following areas.

1. Calculation loss

YOLOv5 loss consists of three components: category loss (BCE loss), object loss (BCE loss) and position loss (CIoU loss), which are calculated as follows.

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \tag{6}$$

2. Calculate the balance loss

The three prediction layers (P3, P4, P5) have different weights for objectivity loss. The weights are [4.0, 1.0, 0.4], respectively, and their calculation equations are as follows.

$$L_{obj} = 4.0 \times L_{obj}^{small} + 1.0 \times L_{obj}^{medium} + 0.4 \times L_{obj}^{large} \tag{7}$$

3. Eliminating grid sensitivity

In YOLOv2 and YOLOv3, the predicted objective information is calculated as

$$b_x = \sigma(t_x) + c_x \tag{8}$$

$$b_y = \sigma(t_y) + c_y \tag{9}$$

$$b_w = p_w \times e^{t_w} \tag{10}$$

$$b_h = p_h \times e^{t_h} \tag{11}$$

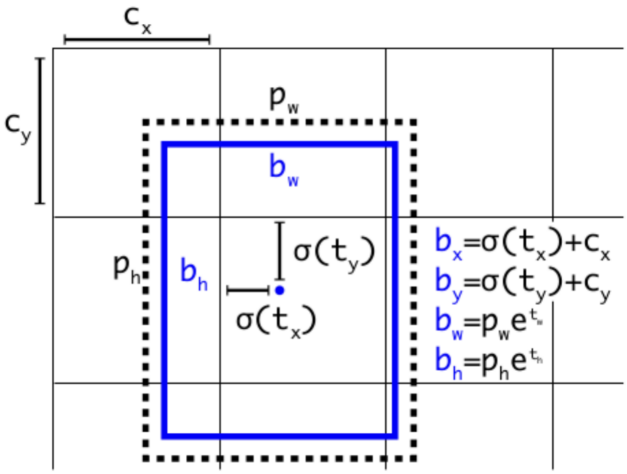

Fig. 10. predicted objective information calculation

In YOLOv5, the predicted target information is calculated as

$$b_x = (2 \times \sigma(t_x) - 0.5) + c_x \tag{12}$$

$$b_y = (2 \times \sigma(t_y) - 0.5) + c_y \tag{13}$$

$$b_w = p_w \times (2 \times \sigma(t_w))^2 \tag{14}$$

$$b_h = p_h \times (2 \times \sigma(t_h))^2 \tag{15}$$

Comparing the center point offset before and after scaling. The centroid offset is scaled from (0, 1) to (-0.5, 1.5). Therefore, offset is easily obtained as 0 or 1.

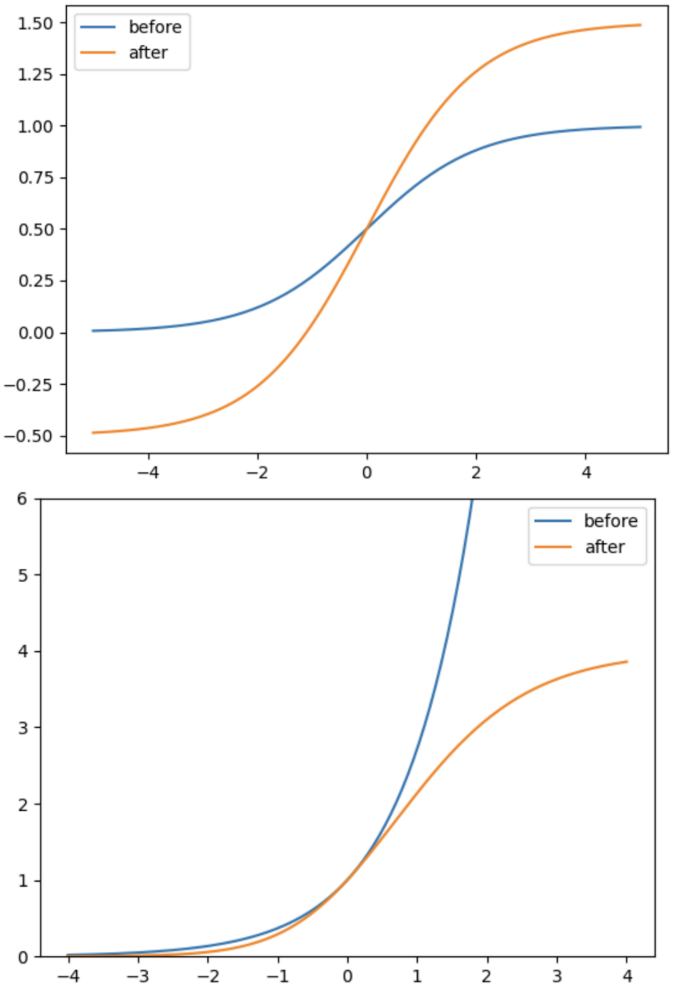

Fig. 11. Comparing the center point offset before and after scaling

Comparing the height and width scaling (relative to the anchor point) before and after the adjustment. The original Yolo/darknet box equation has a serious flaw. The width and height are completely unconstrained because they are just out=exp(in), which is dangerous because it leads to runaway gradients, instability, NaN loss and eventually complete loss of training. YOLOv3 suffers from this problem as does YOLOv4.

For YOLOv5, it can be ensured that this bug is fixed by sigmoidizing all model outputs, while also ensuring that the centroid remains constant 1=fcn(0) so that the nominal zero output of the model will result in the use of a nominal anchor size. The current eqn limits the anchor multiplier from a minimum value of 0 to a maximum value of 4, and the anchor-target matching has been updated to be based on the width-height multiplier with a nominal upper threshold hyperparameter of 4.0.

IV. Experimental Results

A. Experimental setup

The experimental codes of Faster R-CNN were running on Kaggle online Jupyter Notebook platform with Python version 3.7.12, Pyyaml version 5.1, Torch version 1.9 and Torchvision version 0.10. The model was built, trained, and tested using the external library Detectron2 which can be downloaded from https://dl.fbaipublicfiles.com/detectron2/wheels/cu102/torch1.9/index.html The experimental codes of YOLO V5 were running on Google Colab with Python version 3.7.15, torch version 1.12.1, Pandas version 1.3.5 and Numpy version 1.21.6. The model was built, trained, and tested using yolov5.

B. Results

This section tends to discuss the performance achieved by two models and with preprocessing. In this project, we use F2 scores to evaluate the performance of the 2 developed models. The algorithm is based on the evaluation protocol specified in the paper [6]. Corresponding True Positives (TP), False Positives (FP), and False Negatives (FN) are gathered per image level, and the F2 score is calculated for every image for all different IoU thresholds. The formula of F2 will be shown below (3). For each IoU threshold, the final F2 score is the average

$$F2 = 5 \cdot \frac{precision \cdot recall}{4 \cdot precision + recall} \tag{16}$$

From the formula, it can conclude that tackling false negative (FN) is much more important false positive (FP). Furthermore, there exists zero F2 score situation, so we should not calculate the F2 of each image, as those images that have no COTs will result in zero TP, in this situation, no matter the value of FP, the final result of F2 will be zero.

- **Faster R-CNN**

Apart from measuring the F2 score of the 2 models presented in this project, the F2 score for the Faster R-CNN model with Sharpen kernel for image pre-processing and without image pre-processing have also been recorded. From figure 13, it is found that the model using Fast Non-local Mean for image pre-processing has the highest average F2 score.

For IoU thresholds 0.7, 0.75 and 0.8, the F2 score of the model using Fast Non-local Mean is lower than that of model using Sharpen kernel. It is believed that this is due to the 0 F2 score for making predictions in images without COTS. According to the algorithm, the F2 score will be 0 if the model makes any numbers of predicted COTS in an image without COTS, which then leads to a lower average F2 score for that IoU threshold. To prove and mitigate the impact of

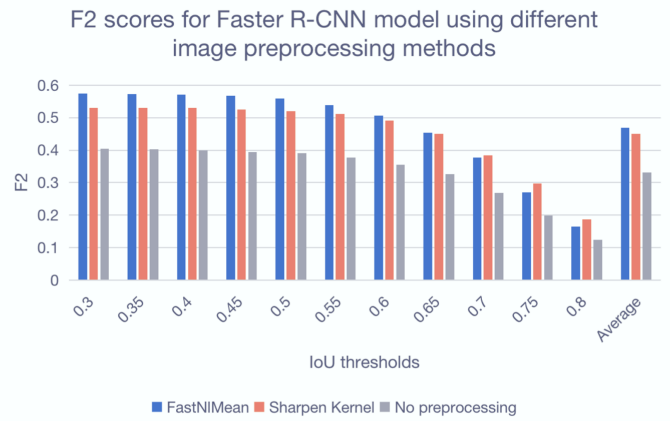

Fig. 12. F2 scores for Faster R-CNN model using different image preprocessing methods

this situation, we proposed a slightly different algorithm to calculate the F2 score for each IoU threshold. Instead of calculating the F2 score per image, TP, FP and FN accumulate for all images and calculate the final F2 score of an IoU threshold using the accumulated variables. In this case, the 0 F2 scores in empty images are eliminated. In figure 13, the updated F2 scores for the model with Fast Non-local Mean when IoU=0.7 becomes the highest among the three numbers. The difference between the Fast Non-local Mean model and the Sharpen kernel model has been reduced.

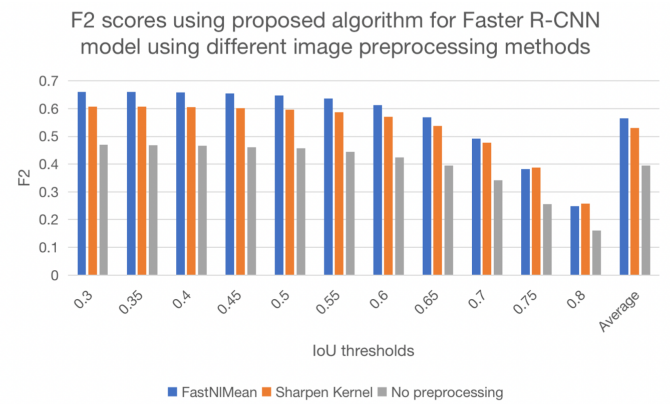

Fig. 13. F2 scores using proposed algorithm for Faster R-CNN model using different image preprocessing methods

- **YOLO V5**

The yolov5 model needs both validation and training datasets for the training. The training data set is for the model training part, while the validation dataset is to avoid the overfitting problem.

Firstly, the class distribution is shown below. Fig 14

We can see that the x-y graph is almost filled with the whole area, which means the cots are located almost at every position. The score, loss and epochs are illustrated below. (Fig 15.)

We can see that the predicted results are quite good. As the epoch increases, all the loss are decreasing except the cls_loss,

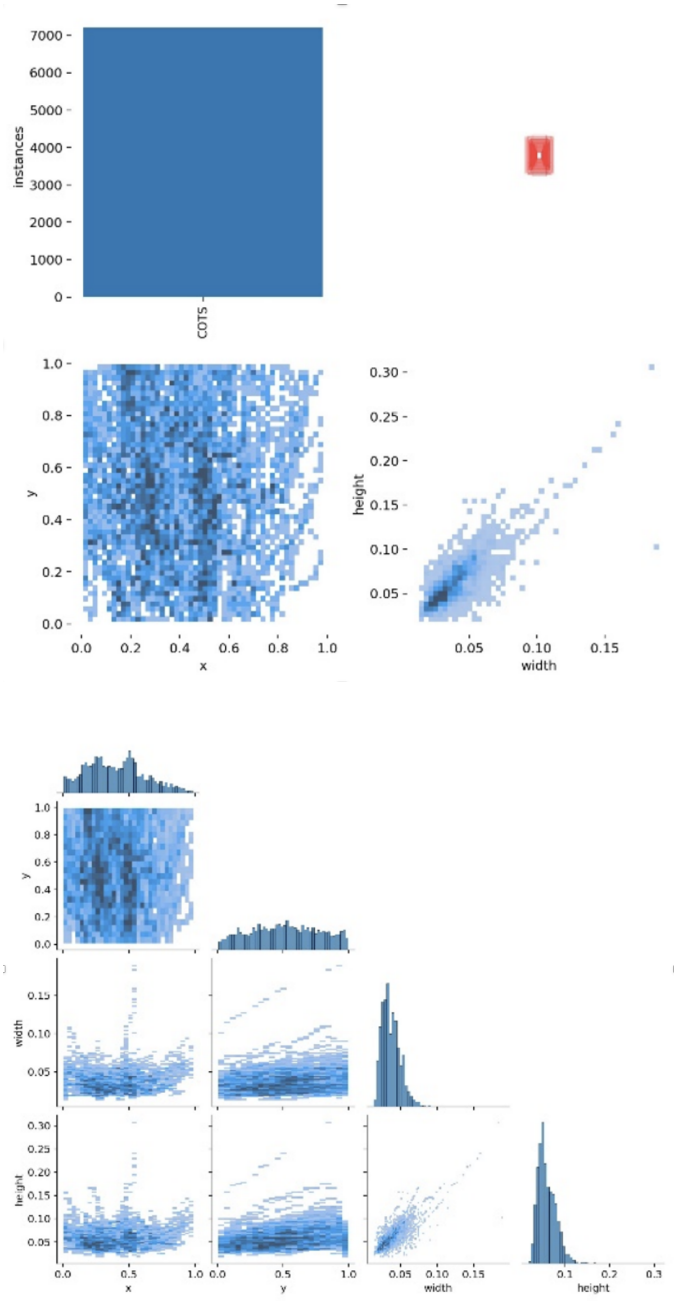

Fig. 14. class distribution

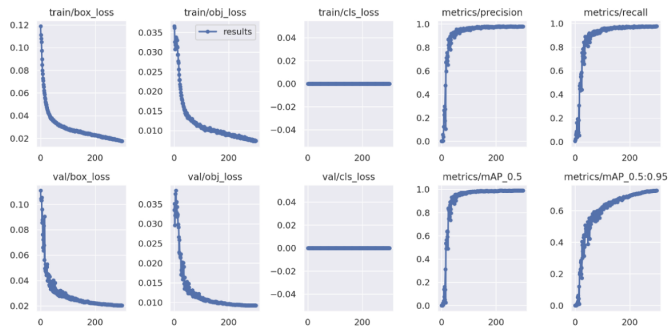

Fig. 15. score, loss and epochs

which has no change. Besides, the result scores are increasing rapidly during the first 50 epochs, which increase smoothly and slowly in the last 200 epochs. The following (Fig 16.) are some figures of different metrics.

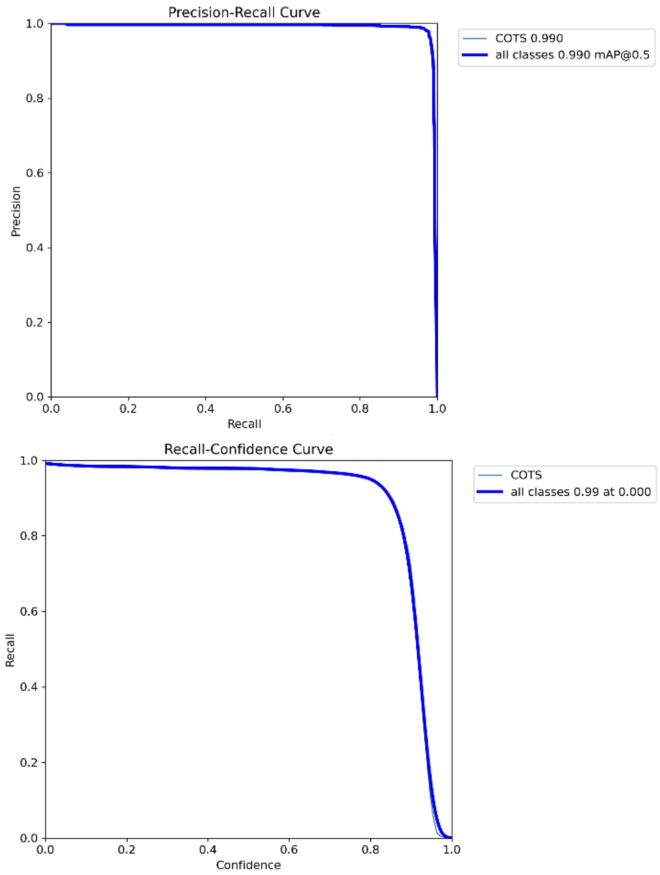

Fig. 16. Metric: precision-recall & recall-confidence

Finally, the F2 score is about 0.755. Fig 17. is the processing of F2 score calculation.

C. Output Images

Fig 18. and Fig 19. are the labels and predictions of images of YOLO V5

Fig 20. and Fig 21. show the prediction of the images of Faster R-CNN

V. DISCUSSION

A. Comparison of F2 scores between the two best model

According to the F2 scores, the best average score of Faster R-CNN is using Fast Nl Mean, which is about 0.576, while the best score of YOLO V5, without any preprocessing, is 0.755.

This is because, unlike Faster R-CNN, which should use Region Proposal Network to detect possible regions of interest and then perform recognition on those specific regions, YOLO can do all predictions with a single fully connected layer. [18]

With different image enhancement methods, the processed image is much clearer than the original one, as shown below, Fig 20. to Fig 22.

```
num_gt:10  num_pred:9   tp:99  fp:0   fn:11
num_gt:2   num_pred:2   tp:22  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:2   num_pred:2   tp:22  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:11  num_pred:5   tp:55  fp:0   fn:66
num_gt:1   num_pred:0   tp:0   fp:0   fn:11
num_gt:1   num_pred:0   tp:0   fp:0   fn:11
num_gt:2   num_pred:1   tp:11  fp:0   fn:11
num_gt:1   num_pred:0   tp:0   fp:0   fn:11
num_gt:4   num_pred:3   tp:33  fp:0   fn:11
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:3   num_pred:2   tp:22  fp:0   fn:11
num_gt:2   num_pred:2   tp:22  fp:0   fn:0
num_gt:2   num_pred:2   tp:22  fp:0   fn:0
num_gt:3   num_pred:3   tp:33  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:6   num_pred:4   tp:44  fp:0   fn:22
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:2   num_pred:2   tp:22  fp:0   fn:0
num_gt:2   num_pred:1   tp:11  fp:0   fn:11
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:2   num_pred:1   tp:11  fp:0   fn:11
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
num_gt:1   num_pred:1   tp:11  fp:0   fn:0
```

Fig. 17. The processing of the F2 score calculation

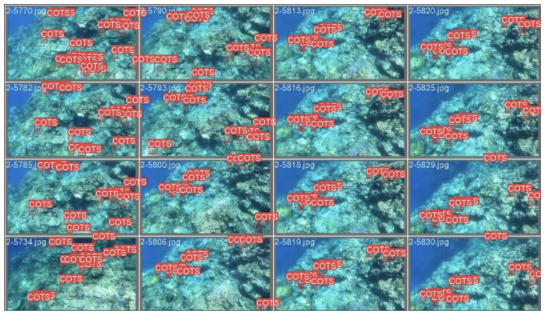

Fig. 18. Val_batch2_labels

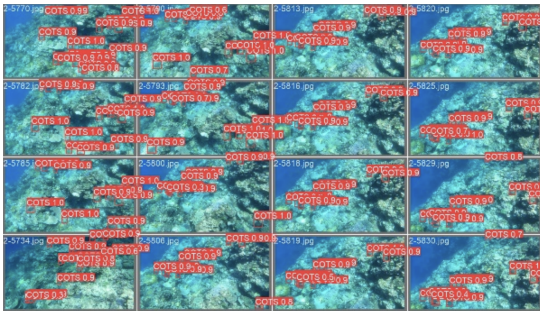

Fig. 19. Val_batch2_pred

Fig. 20. video_0/4259.jpg

Fig. 21. video_1/9355.jpg

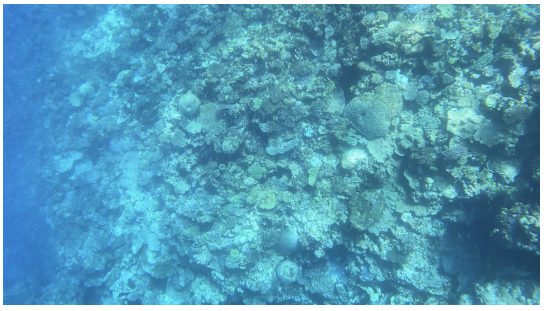

Fig. 22. image without preprocessing

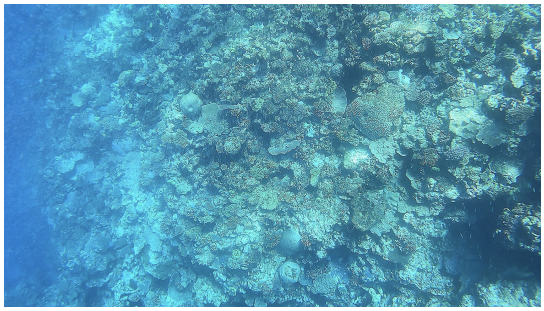

Fig. 23. image with sharpen kernel

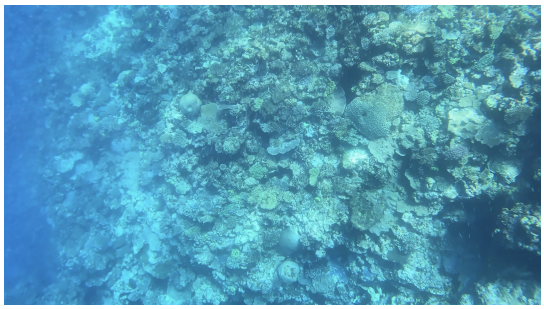

Fig. 24. image with Fast NI

B. Limitation

However, it is difficult for YOLO to detect small objects that appear in groups because each network can only detect a single object. That's the limitation of YOLO.

To make improvements, we can try other methods such as ICM or sea-thru, however, due to the hardware limitation, although these two methods can work better, it needs too much computation, that's the reason why we fail to implement our model with ICM or sea-thru. Moreover, if the camera can provide the water depth and target distance, sea-thru may work much better than ICM.

C. Performance of results

Fig 25. shows the number of predictions made during the testing of Faster R-CNN.

For YOLO V5, we totally used 984 images for detection, while that confidence bigger than 90% will be kept. Finally, there are 858 in 984 images kept.

VI. CONCLUSION

A. Summarize

In this paper, we have discussed and compared two different image enhancement methods: sharpen kernel and Fast Nl Mean, and two developed models: Faster R-CNN and YOLO V5.

Although with some image preprocessing, YOLO V5 still perform much better than Faster R-CNN, because YOLO can do all predictions with a single fully connected layer.

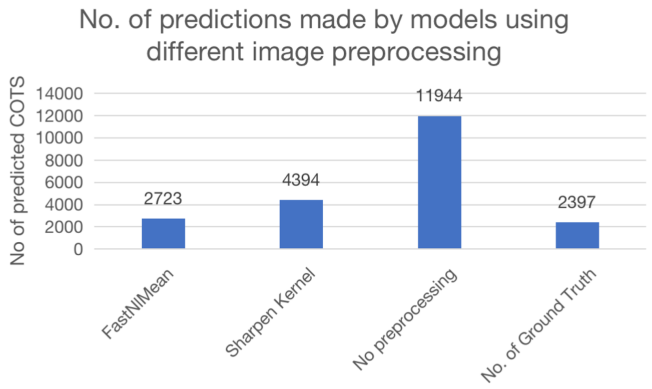

Fig. 25. No. of predictions made by models using different image preprocessing (FR-CNN)

B. Recommend future work

Hardware Improvement:

enhance the hardware, especially the number of GPUs.

Method enhancement:

The main challenge of this task is to remove or eliminate the colour of the sea water and the reflection of the light at the different levels under the sea level. Research work should be focused on this topic in future will not only help to solve this task but also benefits other areas in marine science.

REFERENCES

[1] Mondal, A., Lipps, P., Jawahar, C.V. (2020). IIIT-AR-13K: A New Dataset for Graphical Object Detection in Documents. In: Bai, X., Karatzas, D., Lopresti, D. (eds) Document Analysis Systems. DAS 2020. Lecture Notes in Computer Science(), vol 12116. Springer, Cham.

[2] Kaggle. TensorFlow – Help Protect the Great Barrier Reef: Detect crown-of-thorns starfish in underwater image data. 2021-2022. https://www.kaggle.com/competitions/tensorflow-great-barrier-reef/

[3] N. M. A. Mohamed, L. Lin, W. Chen and H. Wei, "Underwater Image Quality: Enhancement and Evaluation," 2020 Cross Strait Radio Science & Wireless Technology Conference (CSRSWTC), 2020, pp. 1-3.

[4] W. Zhang, P. Zhuang, H. -H. Sun, G. Li, S. Kwong and C. Li, "Underwater Image Enhancement via Minimal Color Loss and Locally Adaptive Contrast Enhancement," in IEEE Transactions on Image Processing, vol. 31, pp. 3997-4010, 2022.

[5] C. Li, S. Anwar, J. Hou, R. Cong, C. Guo, and W. Ren, "Underwater image enhancement via medium transmission-guided multi-color space embedding," IEEE Trans. Image Process., vol. 30, pp. 4985–5000, 2021.

[6] L. Jing, X. Liu and G. Yin, "The research and implementation of the method of pretreating the face images based on OpenCV machine visual library," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, pp. 2719-2721.

[7] K. Das and A. K. Baruah, "A Study of Data Processing for Object Recognition in Scene Image using FRCNN: A Smart Grid Technology," 2021 Innovations in Energy Management and Renewable Resources (52042), 2021, pp. 1-5

[8] G. B. Loganathan, T. H. Fatah, E. T. Yasin and N. I. Hamadamen, "To Develop Multi-Object Detection and Recognition Using Improved GP-FRCNN Method," 2022 8th International Conference on Smart Structures and Systems (ICSSS), 2022, pp. 1-7.

[9] N. Aburaed, M. Al-Saad, M. Chendeb El Rai, S. Al Mansoori, H. Al-Ahmad and S. Marshall, "Autonomous Object Detection in Satellite Images Using Wfrcnn," 2020 IEEE India Geoscience and Remote Sensing Symposium (InGARSS), 2020, pp. 106-109.

[10] W. Zhao, B. Zheng and H. Li, "FRCNN-Based DL Model for Multiview Object Recognition and Pose Estimation," 2018 37th Chinese Control Conference (CCC), 2018, pp. 9487-9494.

[11] C. Wang, A. Bochkovskiy and H. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detetctors", 2022.

[12] T. -H. Wu, T. -W. Wang and Y. -Q. Liu, "Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network," 2021 3rd World Symposium on Artificial Intelligence (WSAI), 2021, pp. 24-28.

[13] W. Fan, X. Guo, L. Teng and Y. Wu, "Research on Abnormal Target Detection Method in Chest Radiograph Based on YOLO v5 Algorithm," 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), 2021, pp. 125-128.

[14] S. Li, B. Pan, Y. Cheng, X. Yan, C. Wang and C. Yang, "Underwater Fish Object Detection based on Attention Mechanism improved Ghost-YOLOv5," 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), 2022, pp. 599-603.

[15] Kernels in Image Processing. [Online] Available: https://www.naturefocused.com/articles/photography-image-processing-kernel.html [Accessed 12-November-2022]

[16] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, Non-Local Means Denoising, Image Processing On Line, 1 (2011), pp. 208–212. https://doi.org/10.5201/ipol.2011.bcm_nlm

[17] S.Ren, K.He, R.Girshick, and J.Sun, "Fasterr-cnn:Towardsreal-timeobject detectionwithregionproposalnetworks,"2016.

[18] YOLO: Real-Time Object Detection Explained [Online] Available: https://www.v7labs.com/blog/yolo-object-detection [Accessed 12-November-2022]