COMP9517 Group Project Report

Crown-of-Thorns Starfish in Underwater Image Data

Tianwei Yang z5135067 | Zhichao Qiu z5268997 | Ziting Wang z5274518 | Chenxiao Wu z5291495 | Haoran Hu z5366098

Abstract—This document is a report for project of implementing object detection algorithm. In this project, a traditional method, HOG & SVM, and a deep learning method, YOLOv5, are used separately to detect the target in the image, Crown-of-Thorns Starfish (COTS). For the obtained detection results, they are processed using Non-maximum Suppression (NMS) and evaluated by using F2_score.

Index Terms—Deep learning, HOG, SVM, YOLO, NMS, F2_score, COTS

I. Introduction

To protect Australia's Great Barrier Reef from the threat of overpopulation of coral-eating Crown-of-Thorns Starfish (COTS), underwater cameras and automated COTS detection methods are used for large-scale video surveillance.

The current task is to implement a target detection algorithm to accurately identify starfish in real time by building an object detection model trained in underwater video of the reef.

The basic task of target detection is to locate the target of interest in the image, correctly define the class of the detected target based on the feature information, and also need to use rectangular bounding boxes to establish the location and size of the target and give the corresponding confidence level to complete the detection task [1] [2].

Target detection is a fundamental research direction under the field of computer vision technology, deep learning and image recognition, and it is a technical prerequisite for more complex high-level computer vision tasks, such as prediction of object behaviour in images after detection tasks; target tracking after object recognition [1], and so on.

The dataset used for this task was downloaded from Kaggle. There are three videos taken by divers using a GoPro Hero9 camera underwater [3]. The videos were segmented frame by frame into a total of 23501 images. The file size of all these images is about 700K, with 720×1280 pixels. Figure 1 shows one of the images in the dataset, and the other images are similar to it.

Based on the information in the .csv file, we can see that some images have annotations and some do not. Annotations indicate the location of the starfish in the image, and if there is no annotation, it means that there is no starfish in the image. Figure 2 shows the location of the starfish in the figure circled in red according to the annotation of one of the images in the .csv file.

Fig. 1. Image in the Dataset

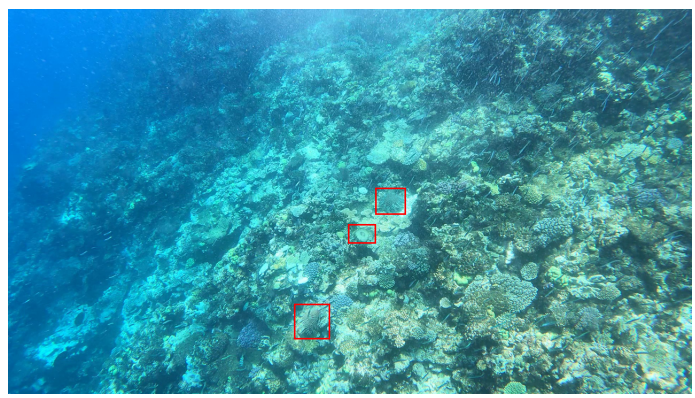


Fig. 2. Example of an Image with Annotations

The dataset in this case has these following characteristics that are different from those in several previous labs and assignments, which bring challenges to the completion of the task [3] [4]:

- Because the source of the material is video, the background in the image is moving and changing, and it is more difficult to exclude the background of the environment in which the detection target is located.
- The same COTS may appear in more than one image.
- Multiple COTS may exist in the same image, and these COTS may be overlapped or hidden, resulting

in the COTS seen in the image is not complete.

Therefore, in order to be able to meet these challenges and obtain a better result, YOLOv5, and Histogram of Orientation Gradient (HOG) and Support Vector Machine (SVM) will be applied in this task.

II. Literature Review

In computer vision, there are two main approaches to achieve object detection: based on motion information, and based on feature information [5].

The methods of detection based on motion information include optical flow method [6], inter-frame difference method [7], background difference method [8], and Gaussian mixture model method [9]. However, these methods have the unavoidable drawback of being susceptible to external conditions. For example, when the camera is in a dithering condition, the lighting of the environment changes, and the object is obscured [10], it can cause errors in the detection results, and it is more difficult to exclude the background of the environment in which the target is detected.

Feature-based information is a hot topic and focus of research in the field of video target detection at this stage. For example, for vehicle detection, among various vehicle features, features of HOG and Haar-like are used more frequently because of their stable effect in vehicle detection [11] [12].

The traditional object detection algorithm is based on region selection by sliding window traversal, then feature extraction of image blocks within the sliding window using features such as HOG, and SIFT, and finally classification of the extracted features using classifiers such as SVM and AdaBoost [13].

AdaBoost [14] consists of strong classifiers by selecting multiple optimal weak classifiers, and multiple strong classifiers cascade into a final classifier. AdaBoost algorithm, in the early stage of training, the selection of optimal weak classifiers is relatively fast, and in the later stage the selection of optimal weak classifiers becomes increasingly difficult, requires a large number of training samples, takes longer and longer to train, and is prone to over-fitting [15].

SVM introduces the concept of kernel function to compute high-dimensional data in low dimensions, and the classification effect of SVM is related to the selection of kernel function and the parameters of kernel function, which are variable (obtained by training) and the selection of parameters is related to the convergence speed and over-fitting of the model during the training process.

In the practical application of video vehicle detection, based on Haar-like features with Adaboost algorithm [16], the detection effect is better in simple scenes, and almost all vehicles in the scene can be detected more accurately. However, in the driving environment with more complex lighting and background, the detection effect is poor and the false detection rate is high [4]. Compared with other target detection methods, HOG features have good anti-interference ability for image light intensity changes and strong robustness for target description [17], and SVM algorithm shows advantages such as simple system structure, global optimality, and short training time [18].

Due to the high computational complexity of traditional methods and limited improvement in detection accuracy. Therefore, in recent years, deep learning-based object detection algorithms have gradually become mainstream, which are divided into two categories according to whether the candidate region is picked first: two-stage and single-stage [13]:

The two-stage algorithm is an object detection algorithm based on candidate regions, in which candidate regions are first selected in the image, and then target classification and position refinement are performed on the candidate regions. The advantage of this type of detection algorithm is that the detailed features of the image can be fully extracted, so the algorithm has better detection accuracy, but the algorithm is divided into two steps, which leads to a slower running model [1].

The single-stage algorithm is based on global regression classification, which is a regression problem for image classification and localization conversion, and can obtain the prediction of bounding box and category probability from the image after one operation, thus directly generating the location and category of the target object [1]. The single-stage algorithm is more real-time because of its simpler model construction, but there is a loss in detection accuracy [13].

Table I and Table II list and summarize the common models of these two types of deep learning-based object detection algorithms, respectively, and their respective advantages and disadvantages.

III. Methods

When an image is obtained, if object detection is to be performed, it needs to be understood in three steps:

- Step 1: Classification. The structure of the image transformed into information is described with a previously determined category or instance ID.
- Step 2: Detection. While the previous step describes the content of the whole image, this step requires selecting an object target for detection and obtaining information about the location and category of the object.
- Step 3: Segmentation. This step needs to segment the semantics and instances and conclude which target object or which scene the pixel belongs to.

A. YOLOv5

YOLO, which stands for YOU ONLY LOOK ONCE is quick and simple for object detection, it is also a one stage detector. From the high-level object detection architecture, YOLOv5 comprises of input, backbone, neck, and head. Backbone refers to a convolutional neural network that forms image features at different levels. Neck is a series

TABLE I
Advantages and Disadvantages of Two-stage Algorithms

Model	Advantages	Disadvantages
R-CNN [19] [20]	For the first time, CNN is combined with the method of recommending candidate frames, and the effect is greatly improved compared with the traditional algorithm	High time cost and large storage space required for extracting features from the generated candidate regions
SPP-Net [21] Fast R-CNN [22]	Only one convolution operation is done for an image, which speeds up training, shortens training time, and saves storage space	The generation of candidate regions is computationally complex, resulting in slower detection
Faster R-CNN [23]	RPN network replaces Selective Search for true end-to-end training	Complex model process, inaccurate positioning frame, and poor detection of small targets
R-FCN [24] Mask R-CNN [25] Cascade R-CNN [26] TridentNet [27]	• Higher positioning accuracy • Higher detection accuracy for instance segmentation • Train multiple detectors using different IOU thresholds • Multi-scale detection by 3 different expansion convolutions	• Complex model process, no improvement in detection speed • Instance splitting is too costly • Slow detection speed, not real-time detection

of network layers that combine image features, it can pass image features to the prediction layer. Head is dense prediction for YOLO.It is mainly to make a prediction to image features and it can generate bounding boxes and predicts categories.

The basic components and network of YOLOv5 are shown in Figure 3.

In YOLOv5, its uses CKPDarknet53 for its backbone and SPP block to separate the critical features without sacrificing the network computational speed. PANet is utilised for its neck as parameter aggregation from various backbone levels. It uses YOLOv3 for its head.

In the Help protect the Great Barrier Reef project, the purpose is to identify the starfish. Utilization of YOLOv5s6 network can quickly train and test the model by using two steps, training, and inference [29].

In training process, images with coco bounding box are used for training. Before training it, the coco format bounding box has to be converted to YOLO format and label all the bounding boxes with a group number, for

TABLE II
Advantages and Disadvantages of Single-stage Algorithms [28]

Model	Advantages	Disadvantages
YOLOv1	Simpler models, faster detection	Poor positioning accuracy, poor detection of small targets and multiple targets
SSD	Fast detection speed, multi-scale detection	The model is difficult to converge, and the small target detection effect is not significantly improved
YOLOv2 YOLO9000	Compared with v1 to improve detection speed and detection accuracy, YOLO9000 can detect 9000 types of targets	Inaccurate positioning of the bounding box, poor detection of small targets
RetinaNet	Propose Focal Loss to optimize the ratio of positive and negative samples to improve detection accuracy and detection speed	Detection speed fails to meet real-time monitoring needs
YOLOv3	Enables more accurate and faster target detection, and multi-scale detection	Unsatisfactory detection of medium and large objects
CornerNet	Detection of diagonal points instead of detection of targets, with improved accuracy	Slow detection process, no use of object feature information
CenterNet	Object detection is achieved by predicting the location of the object's center point and the length and width of the corresponding object	High model computation ,slow detection speed
EfficientDet	Simultaneous speed and accuracy of detection, with the highest detection speed available at a low scale	Detection accuracy and detection speed can be further improved
YOLOv4	Optimal trade-off between detection accuracy and detection speed of the model	The detection accuracy of the model can be further improved
YOLOv5	The model is lighter, with the best detection speed and accuracy at the time	The detection accuracy of the model can be further improved

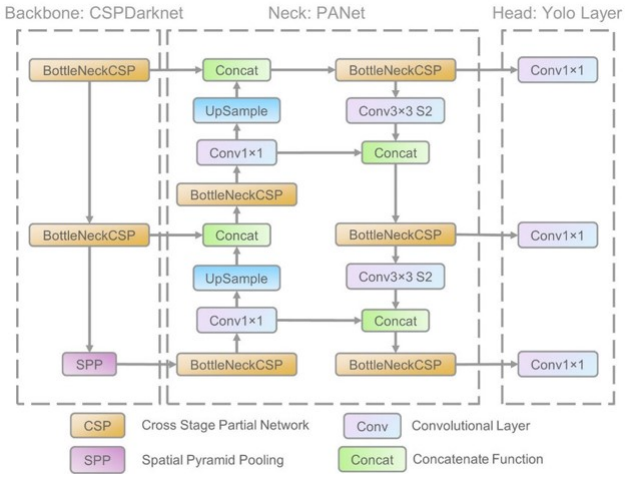


Fig. 3. YOLO Components

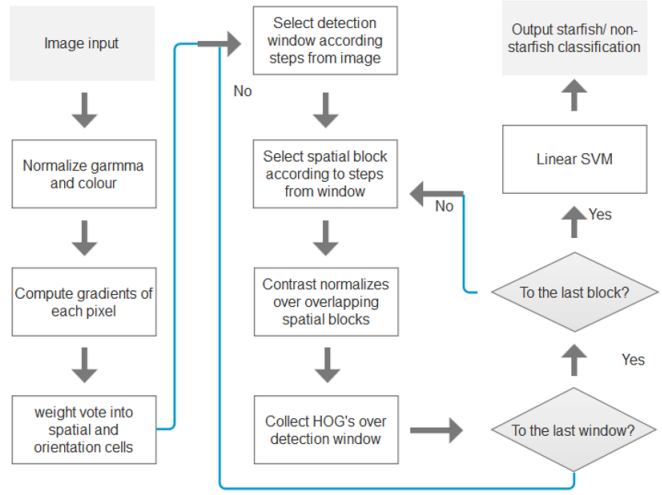


Fig. 4. Steps of HOG & SVM

instance, 0. The YOLOv5 requires an image having labels on it or a txt file with bounding box coordination in YOLO format. Thus, numerical number can be used as a group name for classification.

After that, the default parameters are quite good with 12 epochs, batch size of 4, and image size of 1280. Image mosaic, mix-up change to 0.25 and random image rotation change to 90 degrees. It can obtain a relevant stable increment in accuracy, but it requires more epochs to train and show an expected results.

In inference process, the training model (best.pt from train procedure) is imported, the format of bounding box is COCO, and then keep predicting the bounding boxes to compare with the correct boxes. The format of bounding box from [x1, y1, width, height] need to convert to coco format which is [x_mid, y_mid, width, height]. Once again, the format of bounding box is critical for predicting process.

B. HOG and SVM

The steps of HOG & SVM are shown in Figure 4.

For a single image, Hog will make statistics on the gradient information and generate the final feature description, the feature will be generated as vector, then the vector information will be sent to SVM (Support Vector Machine). Window traversal method is used to achieve the object detection task in Hog method, a window is the smallest feature extraction unit, so the window will move a set step in the entire image to extract feature, after each slide Hog will extract the windows feature then sent to SVM. In order to achieve the goal, there are some steps:

1) Positive and negative sample collection: HOG needs to extract and identify COTS area as positive samples during training, and Hog also needs negative samples to make distinctions, positive samples needs to be avoid in the image while extracting negative samples which is the ground-true. When extracting negative samples. Differentiation and completeness of negative samples are important. To ensure differentiation, for each window contains COTS, we sample four images which surrounding this window and adjacent to it, the goal of this operation is to select some regions containing some ground-true pixels when extracting negative samples, these negative samples can better distinguish the boundaries of positive and negative samples. For completeness, the scene that appears in the dataset should be included as much as possible when extracting negative samples. So two windows were randomly sampled as negative samples in images where COTS did not appear. Differentiation and completeness will make a great performance in this binary classification scenario. Finally, the ratio of positive and negative samples is set to 1/3.
2) Data preprocessing: After extracting positive and negative samples, all samples are randomly classified, splitting all data into 80% train data and 20% test data.
3) Detector Window and Context: HOG and SVM need to adjust the appropriate detection window size, since SVM can only accept vectors of the same size, so our detection window should be the smallest size that can cover COTS, and the window sliding step and image size and jointly decide that several detection windows will be selected, after trying in this project, the size of the detection window is selected as 64×64.
4) After detector window extract features, the hog algorithm is to compute the gradient and make it into vector. The first step is to compute the horizontal and vertical gradients of each pixel.
The horizontal gradients and vertical gradients func-

tion are:

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \quad (1)$$

$$G_y(x,y) = H(x,y+1) - H(x,y-1) \quad (2)$$

H(x,y) is the pixel value, For example, in the Figure 5, the center pixel is 85, then horizontal gradients $G_x(x,y) = 8978 = 11$, vertical gradients $G_y((x,y) = 68 - 56 = 8$.

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Fig. 5. Example of pixel value

After getting horizontal gradients and vertical gradients, we need to compute the gradient amplitude and gradient direction of the center pixel, gradient amplitude, and gradient direction are computed by function 3 and 4.

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (3)$$

$$\theta(x,y) = arctan\frac{G_y(x,y)}{G_x(x,y)} \quad (4)$$

So far, there is each pixels' total gradient and direction value, histogram can be generated by using the total gradient and direction value, which is shown in Figure 6.

C. Non-maximum Suppression (NMS)

After SVM processing, the processed model makes detection of the dataset, and many overlapping boxes will appear, so we will non-maximum suppression by sorting them according to the confidence score, selecting the highest confidence ratio bounding box to add to the final output list, removing it from the bounding box list, calculating the area of all bounding boxes to calculate the IoU of the bounding box with the highest confidence and other candidate boxes. Repeat the above process to delete the bounding box whose IoU is greater than the threshold until the bounding box list is empty until we get the output (Figure 7).

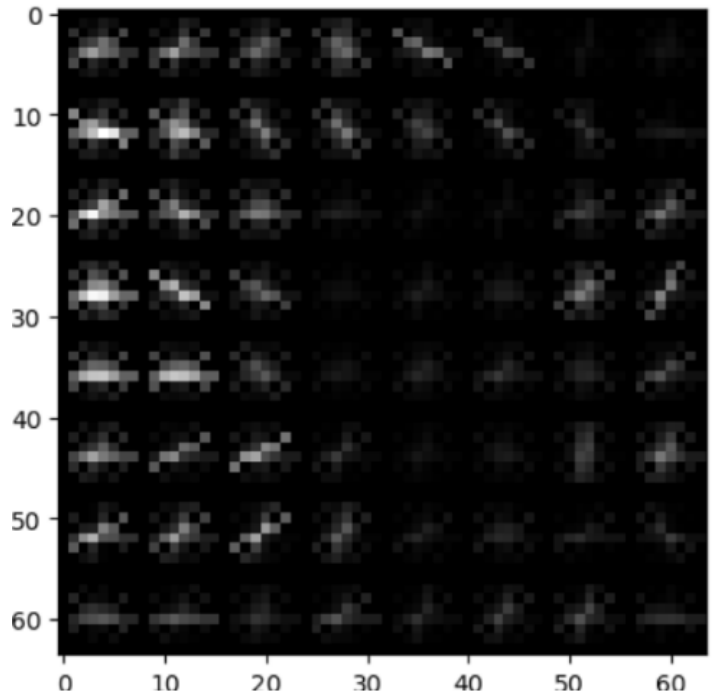


Fig. 6. Histogram

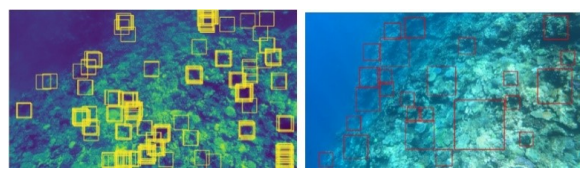


Fig. 7. Non-maximum suppression operation

IV. Experimental Results

A. Settings

1) Dataset: Some basic information of the dataset is shown in Table III.

TABLE III
Dataset information

Methods / Number of items	HOG & SVM	YOLOv5
Training images	18801	18801
Testing images	4700	4700
Annotation images	4919	4919
Non-annotation images	18582	18582

2) Operation System: The running environment of HOG & SVM is shown in Table IV. YOLOv5 is executed on Kaggle. A GPU Kernel will provide Tesla P100 16GB VRAM as GPU, with 13GB RAM + 2-core of Intel Xeon as CPU.

TABLE IV
Dataset information

Methods / Number of items	YOLOv5
Computer System	Win 10
Version number	21H2
CPU Version	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.60 GHz
Memory Size	16G
Running Time	43min

B. Metrics

Recall and precision are two important indicators in evaluating models. Recall is calculated through function 5, and Precision is calculated through function 6.

$$R = \frac{TP}{TP + FN} \tag{5}$$

$$P = \frac{TP}{TP + FP} \tag{6}$$

In general, precision and recall are inversely correlated, therefore, we need to use another metric to evaluate binary classify model, f-score is used to Weigh these two indicators comprehensively. In this project, in order to make sure less COTS are missed out, Recall is more important than Precision. Therefore, F2_score is used to evaluate the performance. In the F2_score metric, false negatives are more important than false positives. Since, the F2_score metric allow some false positive to make sure less COTS are missed out.

As shown in Figure 8, for a each single image, IoU (Intersection over Union) of a proposed set of object pixels and a set of true object pixels is calculated as the area of overlap divide to the area of union.

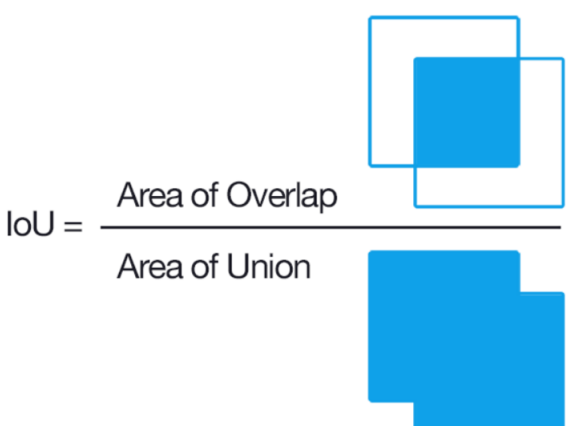


Fig. 8. IoU

Each detection window should be determined as True Positives (TP), False Positives (FP), and False Negatives (FN) window. And the F_score function shows below is calculated based on the number of Positives (TP), False Positives(FP), and False Negatives (FN). Because it is a binary classification scenario, so if the detection window can detect COTS and COTS is truly in the window, it will be counted as TP, if there are no COTS in window, it will be counted as FP, if the COTS exits but window could not detect COTS, it will be counted as FN. Beta is set to 2. F2_score is based on the number of Positives (TP), False Positives(FP), and False Negatives (FN).

$$F_{\beta}^{IoU}(\mathbb{S}) = \frac{(1+\beta^2) \cdot TP(\mathbb{S})}{(1+\beta^2) \cdot TP(\mathbb{S}) + \beta^2 \cdot FN(\mathbb{S}) + FP(\mathbb{S})} \tag{7}$$

C. Results

1) YOLOv5: The results of the SVM training are shown in Figure 9-10.

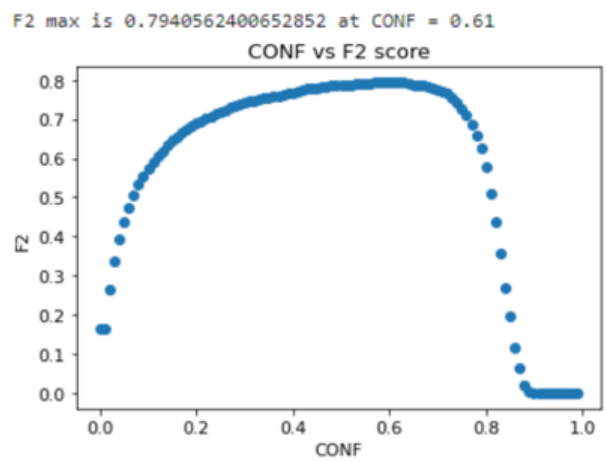


Fig. 9. F2_score of YOLO

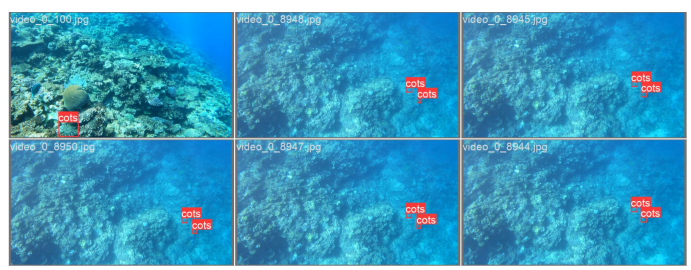


Fig. 10. Testing image of YOLO

2) HOG & SVM: The results of the SVM training are shown in Figure 11-14.

There is also a specific image shown in Figure 12.

While in the real test image, the metrics and performance are poor, and the results are shown in Figure 7.

There are other images that were used for the test and the results are shown in Figure 15(a)-15(d).

	precision	recall	f1-score	support
0	0.95	0.95	0.95	2544
1	0.94	0.94	0.94	1899
accuracy			0.95	4443
macro avg	0.95	0.95	0.95	4443
weighted avg	0.95	0.95	0.95	4443

Fig. 11. Matrics after SVM training

Fig. 12. Image of SVM training

V. Discussion

We can see that the F2_score of HOG&SVM model is very low, it is only 0.19%. The better result is only about 7-8%. It may because the collection of the negative sample are not comprehensive enough, when testing, the sliding window tends to treat background as the actual targets, so the number of false positive gets really high.

As for YOLO model, to the final output image(Figure 13), most of COTS can be detected accurately, and a decent F2_score has been achieved even though there is still room for improvement, as we can see from the Figure 14, when the confidence level gets above 90%, the F2_score drops to almost 0, which means the model cannot get any true values. As one of the testing examples shown on the right, there are windows indicating the detection of the target, but many of them are false positive examples.

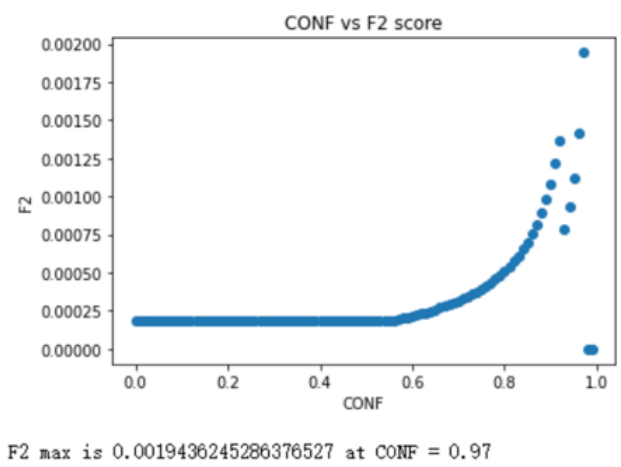


Fig. 13. F2_score of HOG & SVM

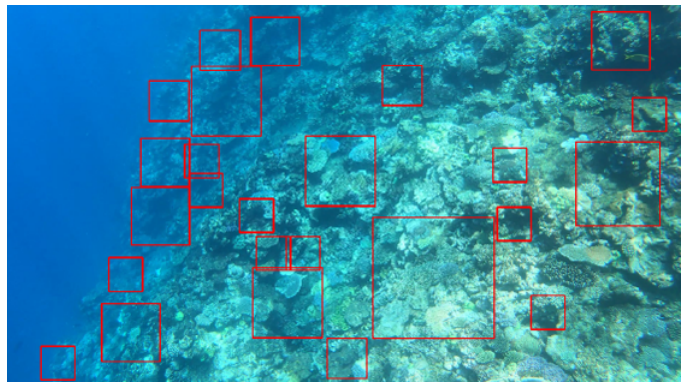
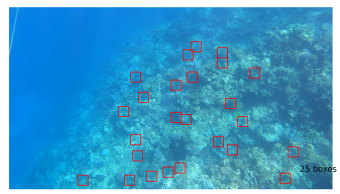
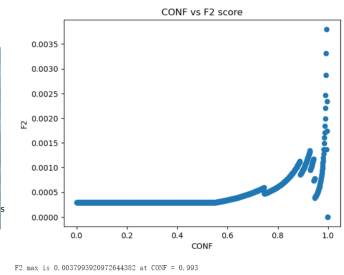


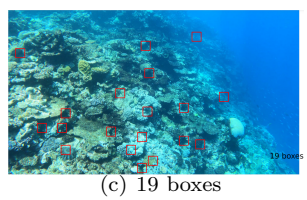
Fig. 14. Testing image of HOG & SVM



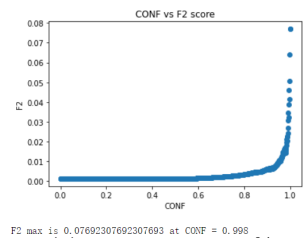
(a) 25 boxes



(b) F2_score = 3.80%



(c) 19 boxes



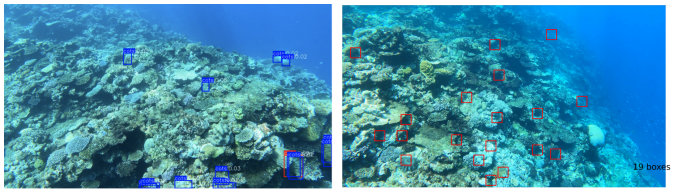
(d) F2_score = 7.69%

Fig. 15. Examples of testing results

Couples of things can be conducted for the future works, for example, we can try different basic parameters such as the batch size and the size of the pre-processed image, or customize some hyper-parameters for the models to see if we can get a better training accuracy. Also, we should do a better job gathering positive and negative samples, try more possible IoU thresholds and we can adjust the size of the sliding window for the SVM model so that it can detect the target more accurately. Overall, both methods worked, however, the results vary a lot for two models and we conclude that the YOLOv5 model works better for this project.

VI. Conclusion

Overall, both methods worked, however, the results vary a lot for two models and we conclude that the YOLOv5 model works better for this project since it achieves better training accuracy and F2_score. The Figure 16(a) and 16(b) (YOLOv5 on the left, HOG+SVM on the right) indicate the statements.



(a) YOLOv5 (b) HOG & SVM

Fig. 16. Testing examples

A. Potential Future Works

To get better results, couples of things can be conducted. Firstly, we can try different basic parameters such as the number of epochs, batch size and the size of the pre-processed image, and customize some hyper-parameters for the models to see if we can get a better training accuracy.

Further, we should do a better job gathering positive and negative samples, try more possible IoU thresholds so that the model can be more discriminatory and more cognizable features can be extracted, consequently, the testing results would be more accurate.

Besides, the size of the sliding window for the SVM model can be adjusted correspondingly for different sizes for targets, in this case, the starfishes. As a result, we may get more accurate outputting windows.

Lastly, for this project, only those with annotations are used for training, adding more data has a great chance of boosting the training accuracy and enhance the clearness of the images can help the model extract the features more sophisticatedly.

References

[1] S. Su, J. Chen, and Z. Zhang, "Survey of Vehicle Object Detection Algorithms Based on Deep Learning," Auto-motive Digest, no. 8, pp. 14–23, 2022.

[2] F. Xie and D. Zhu, "Survey on Deep Learning Object Detection," Computer Systems & Applications, vol. 31, no. 2, pp. 1–12, 2022.

[3] J. Liu et al. The CSIRO Crown-of-Thorn Starfish Detection Dataset. arXiv:2111.14311, November 2021. https://arxiv.org/abs/2111.14311

[4] G. Tan, C. Sun, and J. Wang, "Design of video vehicle detection system based on HOG feature and SVM," Journal of Guangxi University of Science and Technology, vol. 32, no. 1, pp. 19–23, Mar. 2021.

[5] N. Nourani-Vatani, P. V. Borges, J. M. Roberts, and M. V. Srinivasan, "On the use of optical flow for scene change detection and description," Journal of Intelligent & Robotic Systems, vol. 74, no. 3-4, pp. 817–846, 2013.

[6] Q. Geng, "Study on the Key Technology of Intelligent Transportation System based on the Theory of Image Recognition," dissertation, 2016.

[7] Y. Zhang and C. Li, "Improved threshold segmentation algorithm in video vehicle detection," Journal of Guangxi University of Science and Technology, vol. 28, no. 2, pp. 61–66-78, Jun. 2017.

[8] T. Mao, L. Ren, F. Yuan, C. Li, L. Zhang, M. Zhang, and Y. Chen, "Defect recognition method based on Hog and SVM for drone inspection images of Power Transmission Line," 2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS), 2019.

[9] S. Xue, Q. Cai, H. Yang, and X. Xu, "The algorithm of train ticket identification based on OpenCV," Journal of Guangxi University of Science and Technology, vol. 27, no. 2, Jun. 2016.

[10] S. Mane and S. Mangale, "Moving object detection and tracking using convolutional neural networks," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018.

[11] C. Cui, "Research on Vehicle Detection Algorithm Based on Monocular Vision," dissertation, 2017.

[12] W. Mo, Y. Gao, and Q. Zhao, "Confusable vehicle feature extraction and recognition based on cascaded SVM," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), 2017.

[13] P. Li, F. Yuan, and H. Li, "Brief overview of target detection methods," Technology Wind, p. 157, Jun. 2020.

[14] P. Viola and M. Jones, "Robust real-time face detection," International Journal on Computer Vision, vol. 57, no. 2, pp. 137–154, May 2004.

[15] Z. Li, A. Hu, and X. Wang, "Survey of Vision Based Object Detection Methods," Computer Engineering and Applications, vol. 56, no. 8, pp. 1–9, 2020.

[16] Z. Zhu and J. Qiao, "Research of preceding vehicle identification based on Haar-like features and Adaboost algorithm," Electronic Measurement Technology, vol. 40, no. 5, pp. 186–190, May 2017.

[17] S. Han, Y. Han, and H. Hahn, "Vehicle Detection Method using Haar-like Feature on Real Time System," International Journal of Electrical and Computer Engineering, vol. 3, no. 11, pp. 1957–1961, 2009.

[18] Y. Zhou, "Research on Lane Recognition Based on Vision," dissertation, 2019.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, 2014.

[20] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1030–1037, 2009.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.

[22] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448, 2015.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[24] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," 30th Conference on Neural Information Processing Systems (NIPS 2016), pp. 379–387, 2016.

[25] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2961–2969, 2017.

[26] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6154–6162, 2018.

[27] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware Trident Networks for object detection," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6054–6063, 2019.

[28] H. Luo and H. Chen, “A Review of Deep Learning-based Target Detection Research,” ACTA ELECTRONICA SINICA, vol. 48, no. 6, pp. 192–201, Jun. 2020.

[29] S. Gutta, ”Object Detection Algorithm — YOLO v5 Architecture,” Analytics Vidhya, 02 08 2021. [Online]. Available: https://medium.com/analytics-vidhya/object-detection-algorithm-YOLO-v5-architecture-89e0a35472ef