Realization and evaluation of object detection model on underwater video of coral reefs

<z5331336 Junqiu Chen><z5351499 Ruosheng Zhang><z5334986 Yifan Guo>
<z5325987 Wanqing Yang><z5265647 Jie Deng>

I. Introduction

Crown of thorns starfish have a major impact on reef growth and cover. According to Morgan S. Pratchett's survey data on the Great Barrier Reef between 1962-2012 [1], as the density of crown-of-thorns starfish increased, the disease rate and coral bleaching rate of the coral reef increased, and the coral cover decreased to half of what it was at the beginning of the survey . Therefore, the crown of thorns starfish is considered a major difficulty and challenge for coral reef conservation.

Therefore, in order to prevent damage to the Great Barrier Reef, researchers need to monitor the starfish for a long time. At present, the means of monitoring starfish basically rely on artificial diving, and because the seawater itself contains a large amount of suspended matter in reality, the seawater is turbid. However, due to errors in human eye observation, the monitoring accuracy and area are very limited, and the efficiency is low.

Therefore, applying the image detection algorithm to the underwater fishing robot can make it identify the location and type of the object to be fished, and complete intelligent automatic fishing. In this way, the efficiency of fishing success can be greatly improved.

The dataset in this test consists of underwater images taken at various times and locations around the Great Barrier Reef. Because the propagation of light in the water will produce different degrees of attenuation according to the length of the wavelength, the seabed image is usually accompanied by noise, blur, low contrast and other quality problems. In addition, there are other marine organisms on the coral reefs, which greatly increases the difficulty of target recognition tasks [2].

Currently, there are a variety of target detection and recognition algorithms. The project uses yolo v7 and hog+svm algorithms, analyzes and compares the differences between the two, and selects the optimal recognition algorithm. And use the algorithm to perform a high-accuracy starfish recognition system on the given dataset.

Figure1. Image in dataset

II. Literature review

2.1. Target detection

The task of object detection is to accurately and efficiently identify and locate a large number of object instances of predefined categories from images. With the wide application of deep learning, the accuracy and efficiency of target detection have been greatly improved, but the target detection based on deep learning still faces the challenges of improving and optimizing the performance of mainstream target detection algorithms, improving the detection accuracy of small target objects, and realizing multiple The challenges of key technologies such as category object detection and lightweight detection models [3]. The methods used for pedestrian detection in this project are: object detection of YOLO V7 and OpenCV HOG+SVM. After training on the dataset, test whether the two algorithms can overcome the interference from background objects, and calculate their recognition accuracy.

2.2. Underwater object detection

A challenging and attractive task in computer vision is underwater object detection. Although object detection techniques have achieved good performance on general datasets, problems with low visibility and color bias in complex underwater environments lead to generally poor image quality; in addition, problems with small objects and object aggregation lead to Less information can be extracted, and it is difficult to achieve satisfactory results [4].

2.3. K-Fold Cross Validation Concept

In the process of machine learning modeling, it is common practice to divide the data into training and test sets. The test set is data independent from the training and does not participate in the training at all, and is used for the evaluation of the final model. In the training process, the problem of overfitting often occurs, that is, the model can match the training data well, but cannot predict the data outside the training set well. If the test data is used to adjust the model parameters at this time, it is equivalent to knowing some information of the test data during training, which will affect the accuracy of the final evaluation result. The usual practice is to divide a part of the training data as validation data to evaluate the training effect of the model.

The validation data is taken from the training data, but does not participate in the training, so that the matching degree of the model to the data outside the training set can be relatively objectively evaluated. Cross-validation, also known as loop validation, is commonly used to evaluate models on validation data. It divides the original data into K groups (K-Fold), uses each subset data as a validation set, and uses the remaining K-1 sets of subset data as a training set, so that K models will be obtained. The K models are evaluated in the validation set respectively, and the final error MSE (Mean Squared Error) is added and averaged to obtain the cross-validation error. Cross-validation effectively utilizes limited data, and the evaluation results can be as close as possible to the performance of the model on the test set, which can be used as an indicator for model optimization [5].

2.4.Yolov7

YOLO, the full name is "You Only Look Once", is a popular real-time object detection algorithm. The original YOLO object detector was first released in 2016. It was created by Joseph Redmon, Ali Farhadi and Santosh Divvala. At the time of publication, this architecture was significantly faster than other object detectors and became the state-of-the-art for real-time computer vision applications.

The official version of YOLO is only YOLO v1, YOLO v2, YOLO v3, YOLO v4 and the latest version of YOLO v7. Each version of YOLO improves performance and efficiency over the previous version.

YOLOv7 provides a faster and stronger network architecture, providing more efficient feature integration methods, more accurate object detection performance, more robust loss function, and higher label assignment and model training efficiency. YOLOv7 greatly improves real-time object detection accuracy without increasing inference cost. The official version of YOLOv7 has higher accuracy and 120% faster speed (FPS) than YOLOv5 under the same volume [6]. In the range of 5FPS to 160FPS, both speed and accuracy, YOLOv7 surpasses currently known detectors, and tested on GPU V100, the model with an accuracy of 56.8% AP can reach 30 FPS (batch=1) The above detection rate, at the same time, this is the only detector that can still exceed 30FPS with such high accuracy.

2.5. What is OpenCV?

OpenCV [OpenCV] is an open source computer vision library. OpenCV is designed for computational efficiency with a focus on real-time applications. OpenCV is written in optimized C language and can take advantage of multi-core processors.

One of the goals of OpenCV is to provide an easy-to-use computer vision infrastructure that helps people quickly build fairly complex vision applications. The OpenCV library contains over 500 functions covering many domains of vision and is a general enough computer vision library for any machine learning problem [7].

2.6.HOG

Orientation gradient histogram is used to characterize detected objects in image processing. In an image, a local object can be described by its edge gradient and directional density, and HOG is used to characterize such features. Due to its good local geometric invariance and optical invariance, HOG features are widely used in the field of pedestrian detection and have achieved good application results [8].

2.7. SVM

Support vector machine is a supervised classification and discrimination algorithm with a solid statistical theoretical foundation, which divides the data through the maximum margin hyperplane. It combines the concept of VC dimension, the theory of structural risk minimization and the kernel function method on the basis of linear classifier, which can solve the linear inseparability problem of complex classification models [9].

III. Methods

3.1 Yolo

Algorithms for target detection are divided into two categories, namely one-stage algorithms and two-stage algorithms.

YOLO, the full name is You Only Look Once, which vividly illustrates the characteristics of its first-stage algorithm: fast speed, but lower accuracy than the second stage. The generation stage of the candidate area is abandoned, the category probability and position of the object are directly calculated, and the final result can be obtained through one detection. The YOLO series algorithm is a series of one-stage algorithms. From v1 to the current v7, there are a total of 7 versions. The YOLO algorithm is also constantly improving and producing new versions. Therefore, if you want to understand YOLO v7, you need to learn from the previous version. . Since the first three versions are too long ago, this section will focus on the v4 version.

1. The basic process of YOLO algorithm is as follows:

1) Divide an image into S×S grids (also called grid cells), if a manually marked object

The center of the object falls in the grid, then the grid is responsible for predicting the object.

2) Each grid needs to predict B bouding boxes, that is, bounding boxes, and each bounding box needs to predict

In addition to measuring the location (x, y, w, h), it is also necessary to predict a value called Confidence (taking the confidence value),

Each grid also needs to predict score scores for C classes. The final number of predicted values E should be

$$E = S \times S \times [B \times (4 + 1) + C] \qquad (3\text{-}1)$$

The meaning of the Confidence value is equivalent to the IoU mentioned above, that is, the intersection ratio between the predicted frame and the real frame. The actual form is Pr (Object) × IoU. When there is an object in the predicted frame, Pr is 1, and the entire value is Equal to IoU, when there is no object in the prediction box, Pr is directly 0. The calculation method of the final prediction accuracy is the prediction score of a certain category × IoU, that is, the final prediction value can reflect both the accuracy of the category and the accuracy of the detection position.

3.1.1 YOLO v4

YOLO v4 adds many more practical structures to the YOLO v3 algorithm, and modifies the most advanced methods to make it more effective and more suitable for single GPU training, including CBN, PAN, SAM, etc. The speed and accuracy of YOLO v4 have been greatly improved. The main improvement ideas are as follows:

Input: This stage includes an image preprocessing stage, which is to scale the input image to the input size of the network (608*608), and perform normalization and other operations. In the network training phase, YOLO v4 uses data enhancement operations (Mosaic) to improve the training speed of the model and the accuracy of the network, and at the same time uses cross-small batch normalization (CmBN) and self-adversarial training (SAT) to improve the generalization performance of the network;

BackBone：An excellent classification model is not necessarily suitable for target detection, because target detection requires greater network input resolution, deeper network structure, and more parameters to improve accuracy. In YOLO v4, the author compared multiple detection algorithms, and finally used CSPDarknet53 as the benchmark network, which contained 5 CSP modules; used the Mish activation function to replace the original RELU activation function; and used Dropblock in this module to replace the traditional dropout , to prevent overfitting.

Neck intermediate layer: In order to better extract fusion features and improve the diversity and robustness of features, the Neck intermediate layer is usually inserted between the benchmark network and the head network. The Neck middle layer of YOLOv4 adds SPP-block to fuse feature maps of different sizes; and uses the FPN+PAN structure to improve the feature extraction capability of the network.

Head：The output layer mechanism of Yolo v4 is basically the same as that of YOLO v3. The main improvement is that YOLO v4 uses the CIOU_Loss loss function for training, and uses DIOU_nms to replace the traditional NMS operation. The target detection in dense scenes has obtained more accurate detection results, further improving the detection accuracy of the algorithm.

3.1.2. YOLO v5

Shortly after the appearance of YOLO v4, YOLOv5 was born. The algorithm adds some new improvement ideas on the basis of YOLOv4, which greatly improves its speed and accuracy. The main improvement ideas focus on the input layer, backbone and neck layers:

Input: In the model training phase, some improvement ideas are proposed, mainly including adaptive anchor frame calculation and adaptive image scaling; YOLOv5 newly adopts adaptive anchor frame calculation. In the YOLO series of algorithms, for different data sets, it is necessary to set anchor boxes of specific length and width. In the YOLOv3 and YOLOv4 detection algorithms, a separate program run is required to obtain the initial anchor box. In YOLOv5, this function is embedded into the code, and the best anchor box can be adaptively calculated every time it is trained without additional operations. At the same time, adaptive image scaling has been added. For different target detection algorithms, we usually need to perform image scaling operations. However, there are some problems with the original scaling method in the YOLO series of algorithms. Therefore, after scaling and filling, a large amount of information redundancy will be caused, which will affect the reasoning speed of the entire algorithm. . In order to further improve the reasoning speed of the YOLOv5 algorithm, the algorithm proposes a method that can adaptively add the least black border to the zoomed picture. The specific implementation steps are as follows:

(1) Calculate the zoom ratio based on the size of the original image and the size of the image input to the network. Assuming that the image size is 800*600 at this time, you can get 416/800=0.52, 416/600=0.69, and take 0.52 as the zoom ratio;

(2) Calculate the scaled picture size according to the original picture size and scaling ratio, 800*0.52=416, 600*0.52=312, the scaled picture size is 416*312;

(3) Calculate the filling value of the black border: first, subtract the two sides after scaling to obtain the length of the black border that needs to be filled 416-312=104; then perform a remainder operation on the value, that is, 104%32=8, use 32 It is because the entire YOLOv5 network has performed 5 downsampling operations, 2^{5} =32; finally, divide the remainder result by 2, that is, the filled area is spread to both sides, and each side is filled with 8/2=4. In this way, the 416*416 size picture is reduced to 416*320 (312+4+4) size, which greatly improves the inference speed of the algorithm.

Backbone: Integrates some new ideas in other detection algorithms, mainly including two structures: Focus and CSP;

The main purpose of the Focus structure is to crop the input image. The size of the original input image is 608*608*3, and a feature map of 304*304*12 is output after cropping; then a convolution layer with 32 channels is output to output a feature map of size 304*304*32.

CSP structure: In the YOLOv4 network structure, the design idea of CSPNet is borrowed, but the CSP structure is only used in the Backbone backbone network. In YOLOv5, two CSP structures are designed, the CSP1_X structure is applied to the Backbone backbone network, and another CSP2_X structure is applied to the Neck network.

Neck network: The Neck network of YOLOv5 still uses the FPN+PAN structure, but some improvements have been made on its basis. In the Neck structure of YOLOv4, ordinary convolution operations are used. In the Neck network of YOLOv5, the CSP2_X structure mentioned above is used to replace part of the CBL module, thereby enhancing the network feature fusion ability

Head output layer: The anchor frame mechanism of the output layer is the same as that of YOLOv4. The main improvement is the loss function GIOU_Loss during training and DIOU_nms for prediction frame screening.

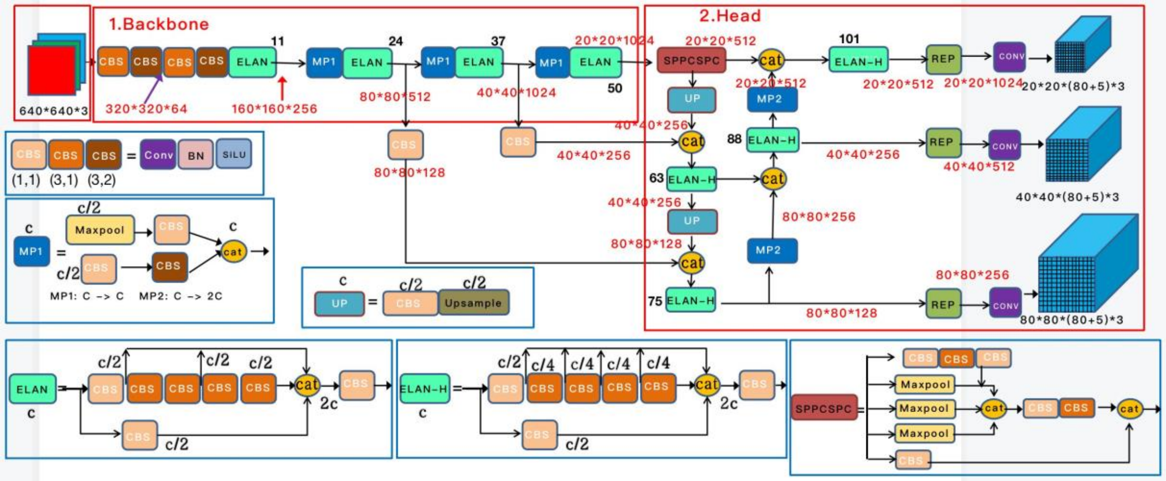

Figure 2. yolov7 structure

3.1.3. YOLO v7

YOLO v7 is similar to YOLOV5 as a whole, and has not undergone major updates or changes. Instead, internal components of a better network structure are replaced. In the range of 5FPS to 160FPS, YOLOv7 exceeds the currently known detectors in terms of speed and accuracy.

Input: The code and preprocessing of the input end are consistent with Yolo v5. First, resize the input image to a standard size (the official usually uses a large size such as 640*640) and input it into the backbone network. Then the head layer network is output to three layers of feature maps of different sizes, and the prediction results are output through Rep and convolution. The final result is the number of anchors * output category * feature map size.

Backbone: The backbone of YOLOV7 has a total of 51 layers, and the main part uses ELAN and MP structures. After the image is processed by input, it first passes through 4 layers of CBS results. Each CBS is composed of a convolution layer + BN (Batch normalization layer) and a Silu activation function. After 4 CBSs, the feature map becomes 160*160*128 in size. An ELAN module is then passed. ELAN consists of multiple CBSs whose input and output feature sizes remain constant. The number of channels in the first two CBSs (convolution kernel and step size are both 1, used to modify the number of channels) will become 1/2 of the previous ones, and the next few CBSs are used for feature extraction. After the number of channels is combined, Then pass through the last CBS (same as the one at the entrance) to output the desired channel. Then there is the combination of three MP + ELAN. Each MP has 5 layers and contains two branches for downsampling. Since each MP has 5 layers and ELAN has 8 layers, the number of layers of the entire backbone is 4 + 8 + 13 * 3 = 51 layers. If starting from 0, the last layer is the 50th layer.

Head output layer: The overall structure of the Head output layer is similar to that of YOLOV5, and still uses the FPN+PAN structure. First, each MP + ELAN combination of the backbone corresponds to an output (C3/C4/C5), and the corresponding output sizes are 80 * 80 * 512, 40 * 40 * 1024, 20 * 20 * 1024.

The first is the 32-fold downsampling feature map C5 that is finally output. After SPPCSP, the number of channels is changed from 1024 to 512. First merge with C4 and C3 according to top down to get P3, P4 and P5; then merge P3 with P4 and P5 according to bottom-up. Then pass the fused result through some kind of deformed ELAN module. The difference between it and the ELAN in Backbone is the number of cats. At the same time downsampling becomes MP2 layer. Afterwards, the sum of the three branches during training will be output, and the parameters of the branches will be reparameterized to the main branch during deployment. After that, get the final result

So far, we have introduced the basic structure of YOLOV7

3.1.4. F2-Score

F-Score is a classification indicator for a more comprehensive evaluation of the balance of accuracy and recall. When classifying a sample set (including positive samples and negative samples), the following situations will occur:

TruePositive: Classify what was originally a positive sample into a positive sample;

FalsePositive: classify the original negative samples into positive samples;

FalseNegative: The classification that was originally a positive sample is a negative sample;

TrueNegative: The classification that was originally a negative sample is a negative sample.

The definitions of precision and recall are as follows

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \qquad (3\text{-}2)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \qquad (3\text{-}3)$$

F-Score definition:

$$F_{\beta} = (1+\beta^2)\frac{PrecisionxRecall}{\beta^2 \times Precision + Recall} \qquad (3\text{-}4)$$

Therefore, F2-Score is the F-Score when β=2, that is, the recall rate at this time is twice as important as the accuracy rate.

3.2 HOG (Histogram of Oriented Gradients)

3.2.1 Background

The HOG (Histogram of Oriented Gradients) is a feature descriptor similar to the Canny Edge Detector and SIFT. It is widely used in the field of computer vision and image detection. It is often combined with SVM and is used in pedestrian detection in movies and videos, as well as vehicles and common animals in still images. A HOG focuses on the structure or shape of an object. It is better than any edge descriptor because it uses the Angle of magnitude and gradient to calculate features. For areas of the image, it generates histograms using the magnitude and direction of the gradient. The main purpose of HOG algorithm is to calculate the gradient of the image and calculate the gradient direction and gradient size of the image. The edge and gradient features extracted by him can well capture the features of local shapes. Moreover, due to Gamma correction and cell normalization of the image, they have good invariance to geometric and optical changes, and transformation or rotation has little impact on sufficiently small areas.

3.2.2 The main steps of the HOG algorithm

Image preprocessing

Preprocessing includes grayscale and Gamma transformation. Grayscale processing is optional, as both grayscale and color images can be used to compute gradient maps. For a color image, first calculate the gradient for the three channel color values, and then take the one with the largest gradient value as the gradient of the pixel. Then perform gamma correction, adjust the image contrast, reduce the impact of light on the image (including uneven lighting and partial shadows), and restore the overexposed or underexposed image to normal, which is closer to the image seen by the human eye.

Gradient calculation

In order to obtain the gradient histogram, it is first necessary to calculate the horizontal and vertical gradients Gx and Gy of the image. Generally, a specific convolution kernel is used to implement image filtering. The available convolution templates are: sobel operator, Prewitt operator, Roberts template, etc. The same result can be obtained using the sobel operator with a kernel size of 1, as does OpenCV. Use sobel horizontal and vertical operators to convolute with the input image to further obtain the magnitude of the image gradient. The direction of the image gradient: θM=arctan(dy/dx)

Calculate the gradient histogram

At this point, each pixel has two values: gradient magnitude and gradient direction.

In this step, the image is divided into several 8×8 Cells, as shown in the figure below, for example, if we resize the image to a size of 64x128, then the image is divided into 8x16 8x8 Cell units, and each The 8×8 Cell calculates the gradient histogram.

Intra-block normalization

Combining cells into larger blocks, in order to further reduce the impact of illumination on gradient features, intra-block normalization should be performed. At this time, the gradient strength is normalized, and the obtained vector becomes the HOG descriptor.

Then put this vector into the trained classification (SVM), and then you can detect the target.

3.3 Advantage and Disadvantage of HOG Algorithm

Advantage

The core idea is that the shape of the local object detected can be described by the distribution of gradient or edge direction. HOG can capture the local shape information well, and has good invariance to geometric and optical changes. HOG is obtained from the densely sampled image block, and the spatial position relationship between the block and the detection window is implied in the calculated HOG feature vector.

Disadvantage

The acquisition process of feature descriptor is complex and the dimension is high, which leads to poor real-time performance. It is difficult to deal with the problem of occlusion, and it is also difficult to detect the excessive range of human posture and movement or the change of object direction. Compared with SIFT, HOG does not select the main direction, nor does it have the rotation gradient direction histogram, so it does not have rotation invariance itself, and its rotation invariance is realized by using training samples with different rotation directions; Compared with SIFT, HOG itself does not have scale invariance, which is realized by scaling the size of the image in the detection window. Due to the nature of gradient, HOG is very sensitive to noise. In practical applications, after the division of block and Cell, sometimes Gaussian smoothing will be performed to remove noise for each region.

3.4 SVM

support vector machine (SVM) is a binary classification model, which maps the feature vector of an instance into some points in the space. The purpose of SVM is to draw a line to "best" distinguish the two types of points, so that if there are new points in the future, this line can also make a good classification. SVM is suitable for small and medium-sized data samples, nonlinear, high-dimensional classification problems. SVM was first proposed by Vladimir N. Vapnik and Alexey Ya.Chervonenkis in 1963, the current version (soft margin) by Corinna Cortes and Vapnik in 1993, It was published in 1995. Before the emergence of deep learning (2012), SVM was regarded as the most successful and best-performing algorithm in machine learning in recent decades.

SVM maps the feature vector of the instance (taking two-dimensional as an example) into some points in space, such as solid points and hollow points in the figure below, which belong to two different categories. The purpose of SVM is to draw a line that best separates the two types of points, so that if there are new points in the future, this line will also be a good classification.

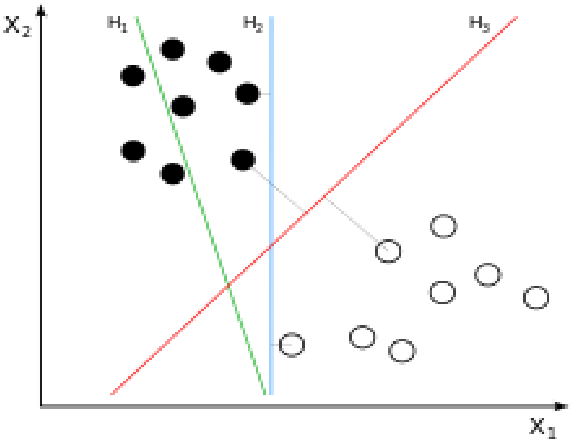

Figure 3. SVM

3.4.1 Non-Maximum Suppression

At present, non-maximum suppression is an essential component of the commonly used target detection algorithms, whether it is SSD series algorithm of One-stage, YOLO series algorithm or RCNN series algorithm of Two-stage. In the existing Anchor-based target detection algorithm, a large number of candidate rectangular boxes are generated, many of which point to the same target, so there are a large number of redundant candidate rectangular boxes. The purpose of non-maximum suppression algorithm is to eliminate the superfluous boxes and find the best object detection position.

3.4.2 HOG+SVM

In our experiment, the above two algorithms are combined to form the HOG+SVM model, which is used for underwater starfish identification. The basic idea is to divide the image into many small connected areas, namely Cell units, and then vote the gradient amplitude and direction of the Cell to form a histogram based on gradient characteristics. Normalized the histogram in the Block. Normalized block descriptors are called HOG feature Descriptors. Combine the HOG description sub-of all blocks in the detection window into the final feature vector. Then, SVM classifier is used for binary detection of target and non-target.

IV. Experimental Results

Coral Reef underwater video target detection model was developed on the anaconda platform. Primarily using pycharm and kaggle as development platforms. python3.10.6 and opencv4.6.0 are used. python libraries used during development include sklearn, skimage, and joblib. kaggle's great barrier reef dataset was adopted, which contained three underwater video image frames, and there were 23,501 pictures in total after consolidation.

4.1 Yolov7

The yolov7 model uses the "GPU T4 x2" of kaggle's computing platform as an accelerator. yolov7 is based on deep learning. Since the yolov7 model has certain requirements for input files, images and labels folders should be generated before training to represent training pictures and the location information of starfish in them, and three yaml files represent training parameters. Use the bbox.utils library to process the starfish coordinates in train.csv into txt file, representing the starfish position information in the current picture. Finally, GroupKFlod was used in the training to segment the data set to improve the accuracy and reduce the overfitting phenomenon. The training time was 4h13min. In terms of data evaluation, yolov7 uses the wandb tool for tracking the training process, which can display more abundant data types, including training curves, pictures, tables, etc. According to the chart generated by wandb, the f2 score of the model calculated by the formula is about 0.6906.

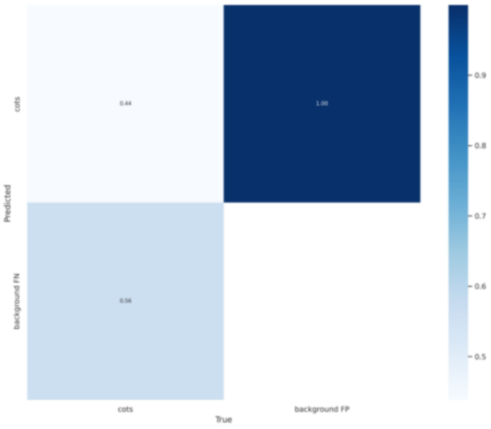

Figure 4. Confusion Matrix of yolov7

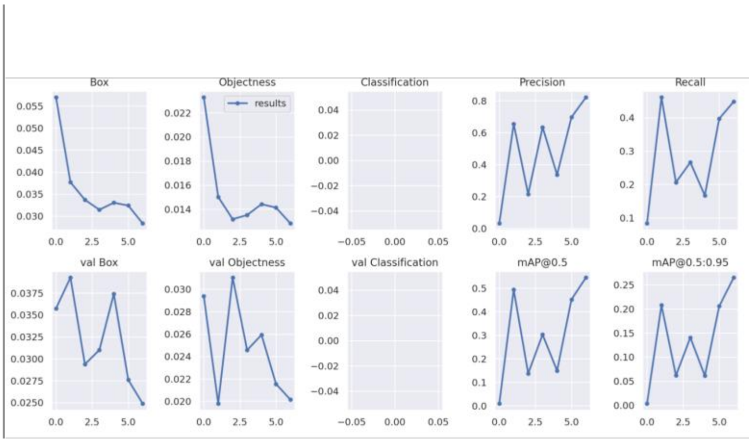

Figure 5. Result of yolov7

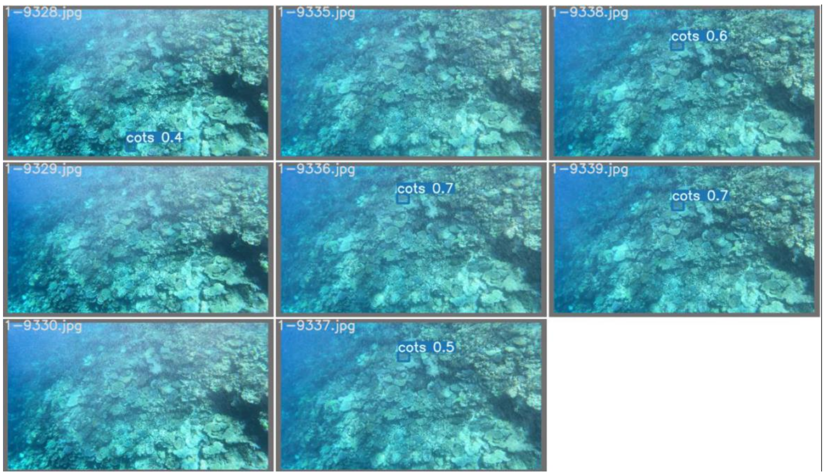

Figure 6. Output images of Yolov7

4.2 hog_svm

The hog_svm model was developed using jupyter notebook. Hog_svm model is based on the traditional image recognition method, which converts the image into hog feature vector, and then uses the binary classification algorithm in machine learning for training. During the training, train_test_split in model_selection was used to split the data set. The training set accounted for 80% and the test set accounted for 20%. The training time was 1 minute and 23 seconds. Since hog_svm model needs to divide the picture into positive and negative, we divided the starfish (positive sample) and non-starfish (negative sample) in the image and placed them in different paths. Then, for each sample image, the corresponding hog vector is generated, and the svm model is used for training. When detecting starfish, we use slip window to detect whether each window contains starfish. Non-Maximum Suppression is also used to optimize images to avoid overlapping of multiple frames. The hog_svm model uses confusion_matrix from the sklearn.metrics package, accuracy_score, precision_score, recall_score, classification_report and other functions were evaluated, and the precision score was about 0.9637, the recall score was about 0.9636, and the f2 score was about 0.5535.

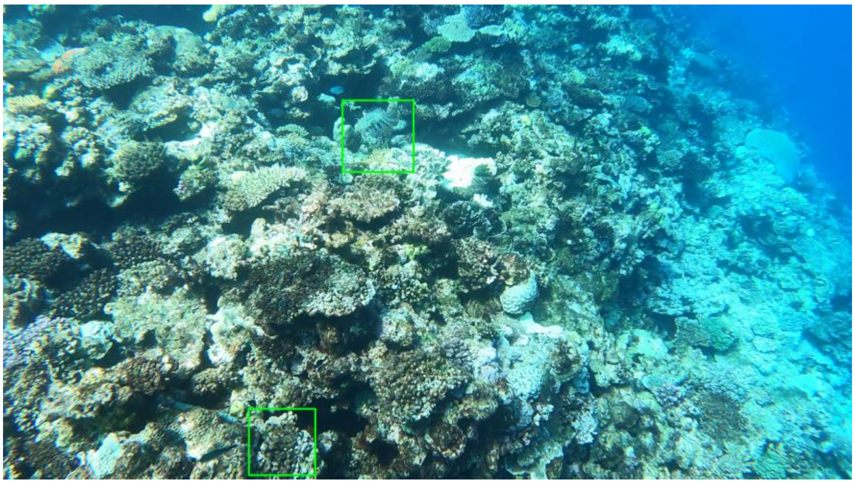

Figure 7. Output image of svm_hog

V. Comparison and Discussion

In the early years of digital image recognition, the use of HOG+SVM model is the best choice. In recent years, due to the development of neural networks and deep learning, a series of image recognition algorithms represented by yolo have appeared.

Compared with the traditional image processing plus artificial intelligence analysis software, the deep learning image recognition model like yolo has considerable advantages in processing the content of this topic. Compared with the traditional image recognition software yolo artificial intelligence processing model has the following three obvious advantages.

The first is the theoretical improvement of the computing speed. Taking the seventh generation deep learning model yolo used in our project as an example, before the input data set is pooled, multiple images will be scaled randomly at different positions and sizes to combine the pictures. In this way, the original feature points of the data set can be guaranteed and the training time can be greatly reduced. This operation also continues the memory space used by the model runtime. In this way, the number of frames can be greatly improved for the video material project that needs dynamic tracking and identification, and the real-time tracking of the dynamic image in the video material can be achieved.

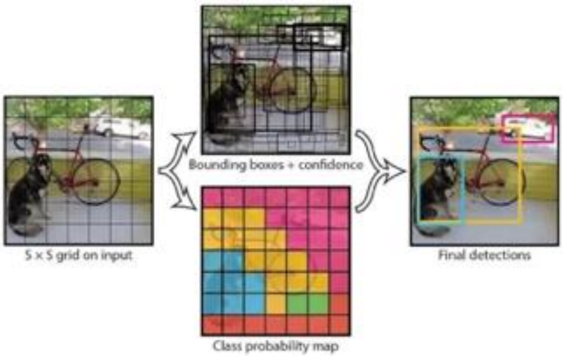

Figure 8. Yolo Predicts Object Position

Secondly, compared with the traditional image processing model function, yolo also has the advantage of stronger versatility. As you can see in the above image [Figure x],yolo automatically puts a border on the pooled data. After such operation, the yolo model will automatically try to identify the pixel center of each object in the data set to calculate whether there is any coincidence of objects. Such capability enables yolo to have incomparable advantages over traditional models in solving practical problems. Because of the same problem of object coincidence, if we want to use the combination of traditional image processing and machine learning algorithm, we are not enough, we also need to introduce other algorithms or conduct in-depth training for specific situations. Therefore, to deal with our current problems, the later development of yolo will also have more advantages. For example, in the future research, other parameters can be added to provide similar analysis such as background environment and starfish status, so as to improve the efficiency of environmental protection.

In terms of training time, almost all deep learning models use the method of gradient descent to update parameters, and many times forward and backward are needed. This means that it takes a long time for the model to be trained when the data volume is extremely large. However, for the traditional computer vision model, so many parameters are not needed, only a hog feature vector needs to be generated for binary classification. Therefore, the training time of yolo is much longer than that of HOG+SVM.

By comparing the f2 score of yolo and HOG+SVM models (table 1), we also come to the conclusion that yolo is superior to HOG+SVM in recognition accuracy.

	Yolov7	HOG+SVM
F2 score	0.6906	0.5535
Time of training	4.5 hours	5.2 minutes
Platform	Kaggle (GPU T4 x2)	Jupyter notebook (CPU)

Table 1. Comparison of the two algorithms

Overall, deep learning algorithms certainly account for the vast majority of image recognition. But in some applications that require high speed and hardware costs, such as autonomous driving, traditional algorithms can still be used to handle some simple, lower-level tasks.

VI. Conclusion

We identified starfish on the seafloor by using HOG+SVM and YOLO7 with kaggle's great barrier reef dataset. Both of our models were successful in identifying starfish and they differ in the speed of processing, the accuracy of the results, and the scope of application. The training speed of HOG+SVM is fast(5 min 23s), but due to the small number of training parameters, it cannot solve the problem of occlusion recognition, resulting in its low accuracy. Due to the problems of the model itself, it will be slow when making large, multi-vector predictions. Therefore, it is necessary to process the data, reduce the amount of data or reduce the number of support vectors to improve efficiency and accuracy, and the scope of application is relatively small. YOLO7 has been improved in many ways since it is the latest algorithm of 2022. YOLO7 has been improved in many ways since it is the latest algorithm of 2022. Compared to the previous version of YOLO, it uses Bag of Freebies to improve model performance without increasing training costs, and has improved speed and accuracy. It requires more parameters to train, more layers, and more training time(4h23min), but its accuracy will be higher, compared to hog and vsm, it can detects objects in the case of overlap as the algorithm predicts the central points of each detected objects. And our YOLO model has more advantages, not only in the drawing of image frames, as the image shows YOLO can drawing frams cut exact edge of target, but also has higher recognition compared with traditional image processing algorithms.

VII. Reference

[1] Pratchett, M, C Caballes, J Wilmes, S Matthews, C Mellin, H Sweatman, et al., ‘Thirty Years of Research on Crown-of-Thorns Starfish (1986–2016): Scientific Advances and Emerging Opportunities’.in Diversity, 9, 2017, 41, <http://dx.doi.org/10.3390/d9040041>.

[2] Wu, R, Bi, X, ‘Coral Reef Benthic Recognition Method Based on Improved YOLOv5’.i Journal of Harbin EngneeringUniversity, 2, 2022, 04, <https://kns.cnki.net/kcms/detail/23.1390.U.20220214.1147.006.html>.

[3] Zhao, Y, Rao, Y, Dong, S, Zhang, J, ‘A Survey of Deep Learning Object Detection Methods’. in JOURNAL OFIMAGE AND GRAPHICS, 4, 2016,<CNKI:SUN:ZGTB.0.2020-04-001>.

[4] Zhang, M, S Xu, W Song, Q He, & Q Wei, ‘Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion’.in Remote Sensing, 13, 2021, 4706, <http://dx.doi.org/10.3390/rs13224706>.

[5] Bian, S, Li, S, Chen, C, ‘Prediction of water quality using weighted cross validation artificial network’.in Computer Engineering and Application, 2015, 51, <https://www.cnki.com.cn/Article/CJFDTOTAL-JSGG201521048.htm>

[6] Wu, D, S Jiang, E Zhao, Y Liu, H Zhu, W Wang, et al., ‘Detection of Camellia oleifera Fruit in Complex Scenes by Using YOLOv7 and Data Augmentation’.in Applied Sciences, 12, 2022, 11318, <http://dx.doi.org/10.3390/app122211318>.

[7] Kaehler, Adrian and Gary R Bradski, Learning OpenCV 3 : Computer Vision in C++ with the OpenCV Library (O'Reilly Media, First edition., 2016)

[8] Zhang, M , Wang, W, Ren, J, Wei, D, et al., ‘Detection and Recognition Algorithm for Frequency Hopping Signals Based on HOG-SVM’.in Journal of Cyber Security, 5, 2020, 3,<http://jcs.iie.ac.cn/xxaqxb/ch/reader/view_abstract.aspx?flag=1&file_no=20200307&journal_id=xxaqxb>.

[9] Zhang, M , Wang, W, Ren, J, Wei, D, et al., ‘Detection and Recognition Algorithm for Frequency Hopping Signals Based on HOG-SVM’.in Journal of Cyber Security, 5, 2020, 3,<http://jcs.iie.ac.cn/xxaqxb/ch/reader/view_abstract.aspx?flag=1&file_no=20200307&journal_id=xxaqxb>.