# COMP9517: Computer Vision

# Help Protect the Great Barrier Reef


1st Li Song

z5331878

2nd Shinan Gao

z5291009

3rd Qifan Zhao

z5387750

4th Zheng Yang

z5342931

5th Yifei Yin

z5256051



***Abstract*—Object tracking in real-time video or time-lapse image sequences is an important and challenging computer vision task in surveillance, traffic monitoring, robotics, medical diagnosis, and biology [1]. However, in many applications, the large volume and complexity of such data make it impossible for humans to identify and analyze the relevant image information accurately, completely, efficiently and reproducibly. Therefore, it is particularly important to monitor video images through computer vision, aiming to develop methods that can analyze images accurately and efficiently.**



***Keywords—object tracking, computer vision***


## I. Introduction

Australia's Great Barrier Reef, the largest coral reef in the world, is under threat, in part because of an overabundance of cots that feed on coral. Therefore, large-scale video surveillance is needed to detect cots outbreaks, in an attempt to manage COTS populations to ecologically sustainable levels. In this project, our task is to apply object detection to develop methods that can accurately and efficiently analyze cots in each image. As the cornerstone of image understanding and computer vision, object detection is undoubtedly the basis for solving complex vision tasks, and its importance is self-evident.

The data set of this project comes from Kaggle. The training set consists of three videos, which contain tens of thousands of images with the size of 1280 * 720 and the corresponding manual annotations.

When training data using traditional methods and machine learning methods, the running speed is slow because the data set is too large. So we scale the images for different methods. In addition, the initial effect of traditional methods is not ideal, and we try to solve this problem by adjusting the window size and reducing the number of negative samples. For yolov7, we reduced the batch size and the number of epochs to make training faster, but this made training so bad that we had to try increasing the batch size and epochs.

Current object detection can be divided into two categories: traditional methods and deep learning methods. Traditional image target detection algorithms can be divided into three main steps: candidate region selection, feature extraction, and classifier. The algorithm is mainly based on Cascade + Harr / SVM + HOG / DPM and its improvement and optimization algorithms. Traditional target detection algorithms are only suitable for situations with obvious features and simple backgrounds, while deep learning can extract rich features of the same target to complete target detection, and is more suitable for practical scenarios. In recent years, deep learning models have gradually replaced traditional machine vision methods and become the mainstream algorithms in the field of object detection.

In this paper, we propose to take HOG features with Support Vector Machine (SVM) and Convolutional Neural Network method (CNN) as examples to compare and analyze the performance of traditional machine learning and deep learning on image classification algorithms. This paper mainly mentions two deep learning algorithms, Fast R-CNN and Yolov7.

## II. Literature Review

### A. Segmentation

1. Traditional segmentation method

Image segmentation is the first step of image analysis and the foundation of computer vision [2]. Image segmentation is to divide an image into several disjoint regions, so that these features show consistency or similarity in the same region, and show obvious differences between different regions [2]. We summarize the various image segmentation methods currently in use:

(1) Threshold segmentation

The thresholding method is especially suitable for images where the object and background occupy different gray levels. The principle is to classify the image pixels into several classes. The threshold segmentation method is actually the following transformation:

$$g(i,j) = \begin{cases} 1 & f(i,j) \geq T \\ 0 & f(i,j) < T \end{cases}, \quad (1)$$

where the input image f to the output image g, T is the threshold, the image element g(i, j)=1 for the object, and the image element g(i, j)=0 for the background. Therefore, the key of the threshold method is to determine an appropriate threshold. The advantages of this method are simple calculation, high efficiency and high speed. The disadvantage is that it is sensitive to noise and not robust [3].

(2) Region segmentation

Region segmentation is to find regions directly. Region growing and split and merge are two typical serial region techniques [4].

1) Region growing

Region growing starts from a single pixel and gradually merges the eligible pixels in the neighborhood of the pixel to form the desired segmentation region [5].

Split-and-merge starts with splitting the whole image to get each sub-region, and then merges the foreground region. The key to this class of methods is the design of the split-merge criterion. The advantage is that the effect of segmenting complex images is better, and the disadvantage is that the algorithm is more complex, the calculation is large, and the region boundary may be destroyed [5].

2) Watershed algorithm

The watershed segmentation method is based on mathematical morphology of topological theory and considers the segmentation of the image according to the composition of the watershed. The advantage is that it has a good response to weak edges and can obtain closed and continuous edges [6]. The disadvantage is that it may be over-segmented by noise.

(3) Edge segmentation

Edge detection solves the segmentation problem by detecting the places where the gray level or structure has abrupt changes. Its advantages are accurate edge location and fast speed. The disadvantage is that the edge continuity and closure cannot be guaranteed [7].

(4) Histogram method

The histogram is calculated by counting the pixels in the image, and the peaks and troughs in the graph are used to locate the clusters in the image [8].

2. Segmentation based on deep learning

VGGNet [9] and ResNet are based on feature encoding but take up a lot of memory and require a long training time.

Region-based selection is a very common algorithm, which detects the region to be detected by detecting the color space and similarity matrix, and then classifies and predicts according to the detection results. Among them, Fast R-CNN is an improved version of R-CNN [10], which is converted from CNN to a large feature map and mapped to the corresponding position. Then, the RoI Pooling Layer is used to extract the features corresponding to each RoI on the feature map, and the FC is used to classify and correct the bounding box. The advantage is that it saves the time of serial feature extraction, but the disadvantage is that the time-consuming selective search algorithm still exists.

*B. HOG+SVM*

The method of pedestrian detection by HOG+SVM was proposed by French researcher Dalal at CVPR in 2005. The HOG+SVM algorithm concentrates on the contrast of silhouette contours against the background [11] [12]. The size of the starfish will vary in the pictures, but their contours are similar. HOG is a descriptor used for object detection in computer vision and image processing. Features are constructed by computing and counting the gradient direction histograms of local regions. Hog feature combined with SVM classifier has been widely used in image recognition. It scans the image at different scales and at each scale examines all the subimages. In each subimage, a 3780-dimensional HOG feature vector is extracted and SVM classifier [13], then used to make a binary decision.

*C. CNN (Convolutional Neural Network)*

CNN is a feedforward neural network that can perform convolutional computations and has deep deconstruction. It is one of the representative algorithms of deep learning. Classification-based CNN can also become a two-stage detection algorithm. The convolutional neural network itself has feature extraction and feature selection and feature classification functions. Then, the convolutional neural network can be used to directly classify the candidate area generated by each sliding window to determine whether it is the target to be detected.

*D. Anchor-based*

The design and use of anchor boxes is an essential part of high-precision object detection. Anchor-based is the current mainstream object detection algorithm, which has two-stage and one-stage. anchor refers to the rectangular boxes with different sizes and aspect ratios set in advance before training, which represent the length, width and height of the main distribution of targets in the data set. These boxes can cover the whole image, and the purpose of this practice is to prevent missed detection. In the process of model training, the length, width and position of anchors are further regressed according to the IoU loss of anchors and ground truth. At the same time, the category of anchors is predicted, and finally these regression classified anchors are output. The number of anchors to be screened and optimized in the two-stage method is far more than that in the one-stage method, and the screening step is more

rigorous, so it is more time-consuming than the one-stage method[14].

## III. METHOD

### A. Hog + SVM Object Detection

HOG is a histogram feature of a statistical target image gradient map. Dalal and Triggs proposed and applied it in human detection, and the detection effect is excellent [12]. Histogram of Oriented Gradient (HOG) feature is a feature descriptor used for object detection in computer vision and image processing. It constructs features by computing and counting the gradient direction histograms of local regions of the image. Hog features combined with SVM classifiers have been widely used in image recognition. Although many object detection algorithms have been proposed continuously, they are basically based on the idea of HOG+SVM. Fig. 1 shows the steps of HOG and SVM classifiation.

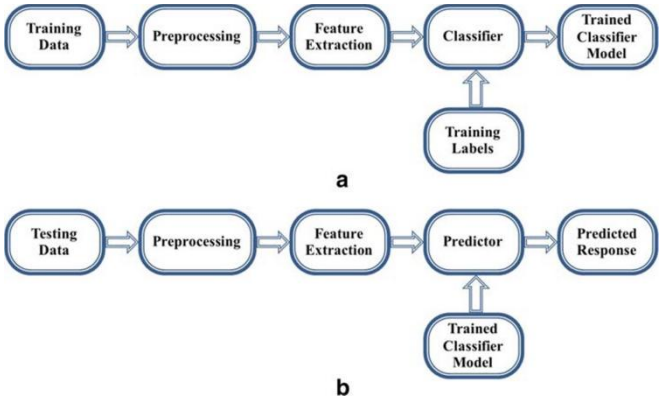



Fig. 1. (a) Training stage (b) Testing stage

1. HOG feature

Gradient orientation histogram (Histogram of Oriented Gradient, the idea of HOG algorithm is to represent the outline of the image target through the distribution of edge directions. The specific method is to divide the recognized image into several fixed-size regions, and obtain the image pixel gradient of the region and use Perform feature calculation to accumulate gradient features, so as to obtain a gradient direction histogram of a certain dimension, as shown in Fig. 2, which is completed by the following steps.

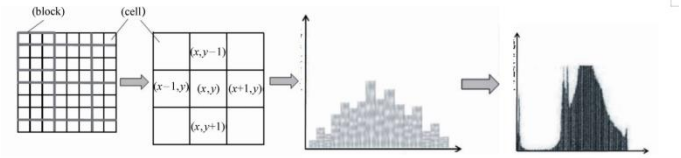



Fig. 2. HOG feature extraction process

1.1 image area division

The image is divided into two layers, the first layer is connected to the Cell units, several Cell form a Block block, and each Block can overlap.

1.2 Gradient calculation

The gradient magnitude and gradient direction of the point are obtained by calculating the gradient of the coordinate direction of the pixel point (x, y). The specific calculation formulas are shown in formulas (2), (3) and (4).

$$\begin{cases} \mathbf{Gx}(\mathbf{x},\mathbf{y}) = \mathbf{H}(\mathbf{x}+\mathbf{1},\mathbf{y}) - \mathbf{H}(\mathbf{x}-\mathbf{1},\mathbf{y}) \\ \mathbf{Gy}(\mathbf{x},\mathbf{y}) = \mathbf{H}(\mathbf{x},\mathbf{y}+\mathbf{1}) - \mathbf{H}(\mathbf{x},\mathbf{y}-\mathbf{1}) \end{cases}, (\mathbf{2})$$

In the formula, Gx(x, y), Gy(x, y), H(x, y) respectively represent the gradient of the x-axis and the y-axis direction and the pixel value of the pixel point in the two-dimensional plane vertical coordinate system. The calculation formula of the gradient magnitude and gradient direction at the pixel point is:

$$\mathbf{G}(\mathbf{x},\mathbf{y}) = \sqrt{\mathbf{Gx}(\mathbf{x},\mathbf{y})^2 + \mathbf{Gy}(\mathbf{x},\mathbf{y})^2}, (\mathbf{3})$$

$$\mathbf{D}(\mathbf{x},\mathbf{y}) = \mathbf{arctan}[\frac{\mathbf{Gy}(\mathbf{x},\mathbf{y})}{\mathbf{Gx}(\mathbf{x},\mathbf{y})}], (\mathbf{4})$$

1.3 Gradient Histogram

Divide the cell's gradient direction 180 degrees into n directional blocks called Bin, and accumulate the n-dimensional gradient magnitude of each Cell.

1.4 Normalization within the block

Combine multiple Cell units into Block blocks for contrast normalization.

1.5 Collect HOG features

Collect the HOG features of all overlapping Blocks in the detection window.

2. SVM Classifiation

SVM (Support Vector Machine) is a research system of high-dimensional feature space using linear functions as hypothesis space. It is trained arithmetically from research derived from optimization theory. As proposed by Vapnik, the algorithm has been successfully applied to face detection and recognition, and has other wide-ranging applications in the recognition of characters, sounds, and others [14].

In SVM classification, the extracted features and SVM coefficients are multiplied and accumulated until the operation reaches the window level. Then, the accumulated result is compared with the SVM threshold to determine whether the window contains the target object.

### B. Fast R-CNN

The Fast R-CNN algorithm based on VGG16 is faster than RCNN algorithm and SPPnet algorithm in training speed and testing speed.

Fig. 3 is the architecture of Fast R-CNN. Fast R-CNN reads the whole picture and ROI set as input, and then extracts features from it to obtain the feature map. The pooling layer extracts the fixed-size feature factor from the feature map and maps it to the corresponding position (softmax probability and bbox regressor) through the FC layer.

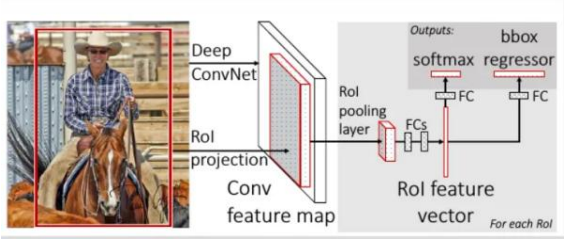



Fig. 3. Architecture of Fast R-CNN

As shown in Fig. 4, RoI pooling is more concise than SPP. Different from the multi-scale pooling operation of SPP, RoI pooling only selects one of the scales, which makes the gradient backpropagation more convenient and is conducive to the implementation of Fast R-CNN and end-to-end training.

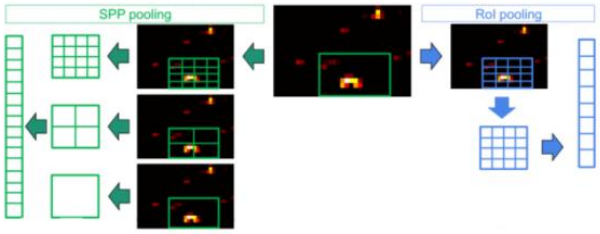



Fig. 4. The difference between SPP and RoI pooling

Furthermore, ROI Max pooling divides the RoI window of h*w into a grid of pooling Windows of h*w, and each pooling window size is about h/w vs. As shown in FIG. 1, 4*4 RoI pooling is performed, that is, a 4*4 size feature map is obtained, and then the feature map is stretched to 16*1 to input the fully connected layer.

### C. YOLOv7

YOLOv7 is obvious from Fig. 5 that YOLOV7 is the most advanced algorithm in the YOLO series. YOLOv7 outperforms multiple object detectors such as YOLOR, YOLOX, Scaled-YOLOv4, YOLOv5, and DETR in terms of speed and accuracy. For this reason, we chose to use YOLOv7.

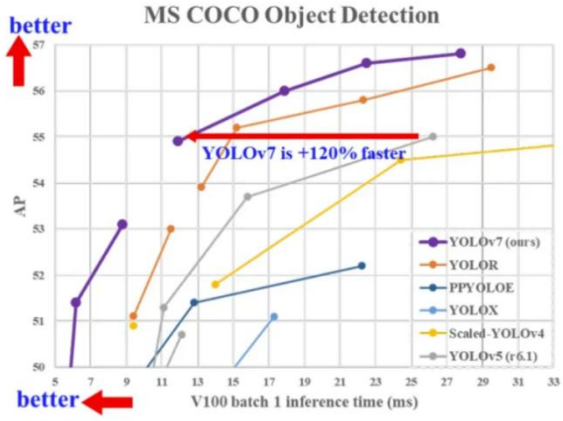



Fig. 5. comparison with others

YOLOv7 has high speed and accuracy in the range of 5 FPS to 160 FPS, and achieves the highest accuracy of 56.8% AP of real-time object detector at 30 FPS on GPU V100. YOLOv7 is trained from scratch on the MS COCO dataset without using any other dataset or pre-trained weights.

YOLO algorithm is different from Fast R-CNN two-stage detection algorithm, which is the most typical representative of one-stage object detection algorithm. It is based on deep neural network for object recognition and location, which has the advantage of fast running speed and can be used in real-time systems.

The image is reshaped to 640*640 and input into the backbone network. The backbone of YOLOv7 has 50 layers, and its structure is shown in the Fig. 6. First, it needs to go through the convolutional layer, CBS is mainly composed of a Conv layer, a BN layer and a SiLU layer, where the formula of silu activation function is, after four CBS, the feature map size becomes 160 * 160 * 128, and then through ELAN module, it learns more features by controlling the gradient path. Then the three MPS and ELAN outputs.

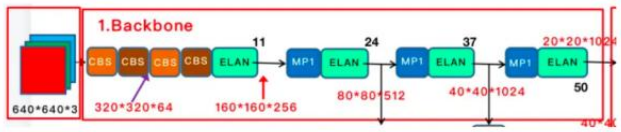



Fig. 6. The structure of backbone

First, we need to import the libraries. Then read train.csv and look at the data type. The dataset was divided into training and validation with a 2:8 ratio and assigned labels. Create multiple folders for the training and validation sets, and the labels for the training and validation sets. Define the convert function to set the size of the box. Get the image size, 1280 x 720. The yolo annotation file is *.txt, each object is in class [x_center, y_center, width, height] format, so the markup box must be standardized xywh format, note that the x_center and width values are divided by the image width, Divide y_center and height by the image height to get a value in the range 0 to 1. Also, the format of this project is coco, so we want to convert it to YOLO format.

Since the detection object of this project is only cots, the number of categories in nc dataset is set to 1, and the category label of names dataset is modified to 'starfish'. In addition to that, set the path to be the path of the data store, respectively.

Train the dataset, we are using a GPU, so the workers are set to 8, the device is set to 0, the batch size for each training is 32, and the image size is 640 x 640. Then, make predictions on the dataset.

## IV. Experimental Setup

### A. Experimental Environment

*Dataset:*
*https://www.kaggle.com/competitions/tensorflow-great-barrier-reef/*

Running Equipment:

For HOG + SVM: 3.4GHz AMD Ryzen 5 2600, 32G DDR4 RAM, NVIDA GeForce RTX 3070Ti 8G

For machine learning: MacBook Pro with 2.4 GHz Quad-Core Intel Core i5 and with 8 GB 2133 MHz LPDDR3

In addition, code update on colab after demo submission,

please try to run on colab if it is possible.
Environment: Python 3.7
Directory:

Figure 7 for HOG + SVM:

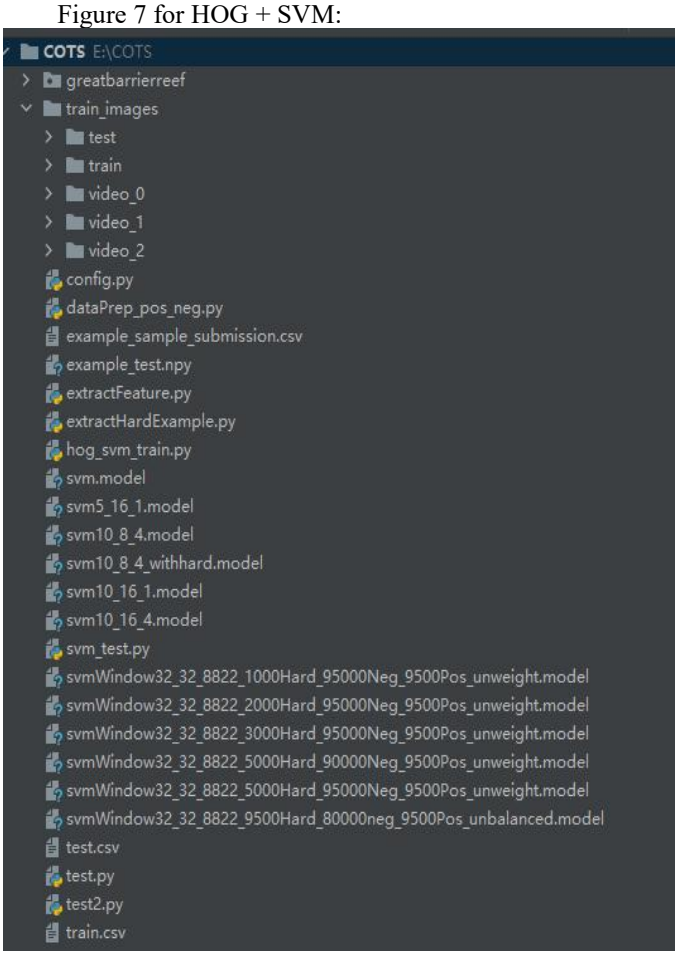



Fig. 7. directory for HOG+SVM

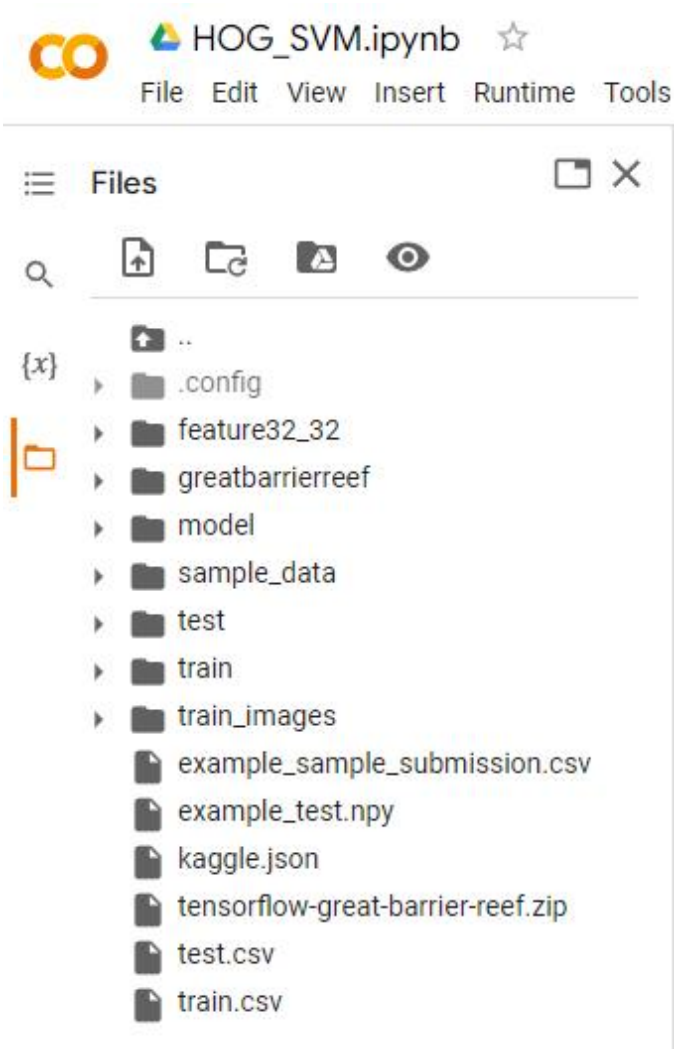



Fig.8. directory for machine learning methods

*B. Experimental setup*

1) HOG + SVM

First, we set the images with annotations in the dataset as positive samples, and the images without annotations as negative samples, and then divide the dataset into training set and test set. Afterwards, we use sklearn.feature.hog to perform the feature extraction operation of the image. When extracting hog features, the selected window size decisively affects the generation of features. A size larger than (128,128) takes too long to extract features, so we finally set the extraction size to (64,128). At the same time, if all Positive Image Resize is used for feature extraction, the obtained data has no reference value, so We decided to cut out the area with bounding box in the positive image to make a new image, and the resize is (64,128). For each negative image, ten size of (64,128) images are randomly cut and features are extracted.

We send the extracted features into linearSVC for training. If the max_iteration of linearSVC is too small, the model may fail to converge successfully. We set the ratio of the positive negative feature to 1:10, and we set the linearSVC balanced weighted.

2) Fast R-CNN

The first step is to install the detectron2 library, the required dependencies and some public libraries we need. Than we define some auxiliary functions to convert our dataset to a specific format. These functions return a list of dictionaries with comments. The next step is to register these training and validation datasets by DatasetCatalog.register and MetadataCatalog methods. After registering the dataset, we can check the training data through the visualization class.

Next part is the training section. To do this, first we import DefaultTrainer from the engine module of Detectron. We define data sets and other parameters, such as number of workers, batch size, and number of classes. After that, we initialize the model with pre trained weights and further train it. According to the size of the dataset and the complexity of the task, we set the number of iterations to 1000.

3) YOLOv7

First, we read train.csv and look at the data type. 80% of the dataset is used for training, 20% for validation during training, and 10% for testing. Create multiple folders for the training and validation sets, and the labels for the training and validation sets. Define the convert function to set the size of the box. Get the image size, 1280*720. The yolo annotation file is *.txt, each object is in class [x_center, y_center, width, height] format, so the markup box must be standardized xywh format, note that the x_center and width values are divided by the image width, Divide y_center and height by the image height to get a value in the range 0 to 1. Also, the format of this project is coco, so we want to convert it to YOLO format.

Since the detection object of this project is only cots, the number of categories in nc dataset is set to 1, and the category label of names dataset is modified to 'starfish'. In addition to that, set the path to be the path of the data store, respectively.

Train the dataset, we are using a GPU, so the workers are set to 8, the device is set to 0, the batch size for each training is 16, and the image size is 640 * 640, and use this setting for training epochs of 100.

When selecting YOLOv7, YOLOv7x is selected to achieve the best effect and the shortest time consumption by the image size (640).

*C. Results*

1) HOG + SVM

After training, we randomly select images to test the effect of this method. The Fig. 9 shows the detection effect of starfish.

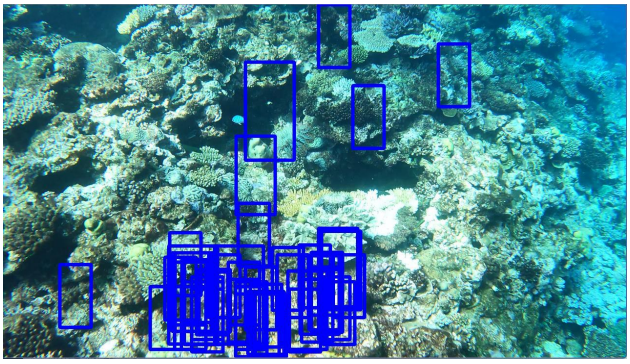


Fig. 9. COTS Detection Image

After testing with randomly selected images, we found that this method cannot effectively detect cots. We analyzed the reasons and discussed the improvement method. Due to the small number of positive samples, in order to train the model normally, we must suppress a large number of simple negative examples in some way and mine the information of all difficult examples, which is difficult example mining. the original intention. That is, during training, try to mine as many hard negatives as possible and add negative sample sets to participate in the training of the model, which will be more effective than the negative sample sets composed of easy negatives.

We decided to mine 500 negative samples, extract hard examples and retrain with positive features negative features. Through the secondary processing of the model, Fig. 10 shows the detection results after retraining the hard example on the image.

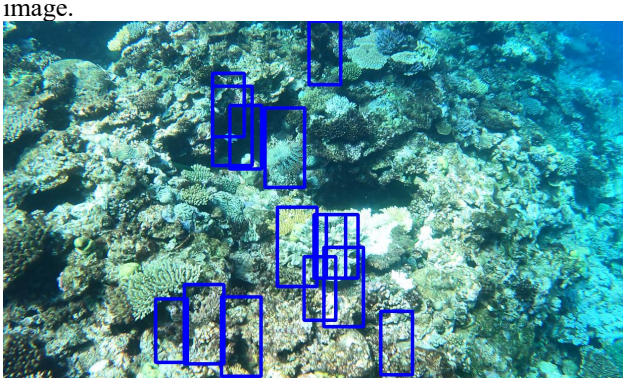


Fig. 10. COTS Detection Image

As we can see from the above, although the improved method has been significantly improved compared to the initial one, it is still unable to accurately identify cots. Through discussion, we think that more negative sample images should be used to process hard example feature extraction, then it can filter out more unreasonable boxes. Then we decided to find the correct ratio of hard feature, negative feature, and positive feature. For this data set, the increase in the number of hard features has a significant effect, and finally we decided on the window of (32,32), 5000 hard features, 9000 negative features, 9500 positive features more reasonable. Fig. 11 shows the detection result in this setting.

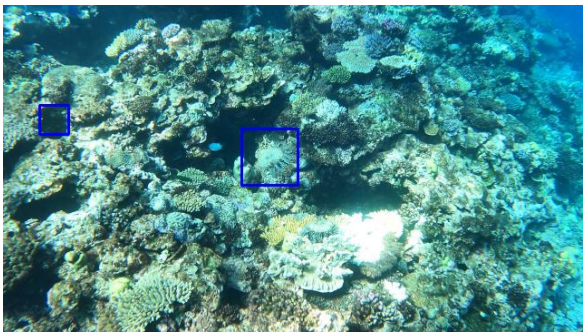


Fig. 11. COTS Detection Image

To get the performance evaluation, we use the F2 metric. The F2-score is calculated from the following formula:

$$F2\ score = \frac{(1+\beta^2)\cdot TP}{(1+\beta^2)\cdot TP + \beta^2\cdot FN + FP} \ (\beta = 2) \quad (5)$$

According to this formula 5, the F2 score is 0.1 which is the lowest value in three methods.

2) Fast R-CNN

After the training, we load the model and initialize the predictor. We take some random samples from the validation data set and pass them to the predictor. The Fig. 12 shows the detection effect of starfish.

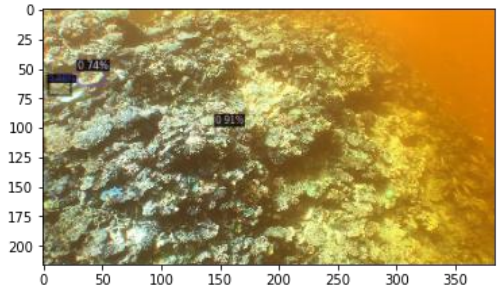



Fig. 12. COTS Detection Image from Fast R-CNN

The training data results provide a more complex table (Fig. 13). According this figure, we can calculate that the value of the F2 scores is 0.375 which is higher than HOG + SVM.

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.198
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.513
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.087
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.301
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.117
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.286
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.350
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.350
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
[11/15 04:57:20 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm  |  APl  |
|:------:|:------:|:------:|:------:|:-----:|:-----:|
| 19.826 | 51.259 | 8.697  | 30.091 |  nan  |  nan  |
```

Fig. 13. The result of Fast R-CNN

The prediction result is good mainly because Fast R-CNN has an excellent training method——the four step alternating iteration training method:

a. Train RPN, initialize the shared convolution and RPN weights with the large dataset pre training model, and use them to generate region proposals;

b. Train Fast R-CNN and use the same pre training model to initialize shared convolution;

c. Optimize RPN, use the shared convolution and RCNN trained in step 2, fix the shared convolution layer, and continue to train RPN

d. Adjust Fast R-CNN, use the shared convolution and RPN trained in step 3, and continue to fine tune the RCNN

3) YOLOv7

After the training, we got a lot of data from yolov7. The following Fig. 14 shows the actual label of the first round of validation set when epoch=100.

Fig. 14. COTS Detection Image from Yolov7

And from the confusion matrix (Fig. 15), we can know that Precision and recall are very high:

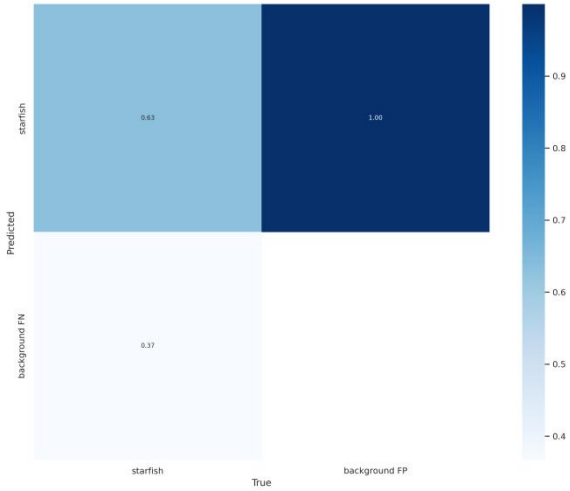



Fig. 15. Confusion matrix of Yolov7

According to the result of Yolov7 (Fig. 16), we can also get various data after training:

Box: box is the mean value of GIoU loss function. The smaller the box, the more accurate it is;

Objectness: objectness is the average loss of target detection. The smaller the object, the more accurate the target detection;

Classification: classification is the mean of classification loss, and the smaller the classification, the more accurate the classification;

Precision: Precision=TP / (TP+FP);

Recall: the true positive accuracy rate which means how many positive samples have been found;

val BOX: bounding box loss of the validation set ;

val Objectness: target detection loss mean of the

validation set;

Val Classification: classification loss mean of the validation set;

mAP@0.5: Indicates the average mAP with a threshold greater than 0.5;

mAP@0.5:0.95: Represents the average mAP over different IoU thresholds (from 0.5 to 0.95, in steps of 0.05) (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95).

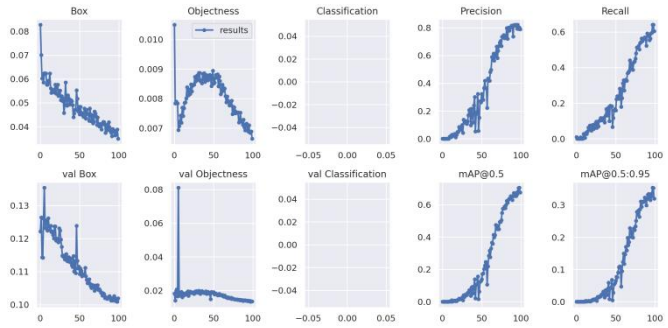



Fig. 16. The result of Yolov7

To get the performance evaluation, we can calculate that the value of F2-score is 0.5595. It is the highest value in three methods.

## V. Discussion

This section will discuss some of the results for both traditional and machine learning methods.

Hog feature combined with SVM classifier has been widely used in image recognition. HOG can describe the local shape information, which can suppress the influence of translation and rotation to a certain extent. Because the histogram is normalized in the local area, the influence of illumination change can be partially offset. In addition, the influence of illumination color on the image is ignored to a certain extent, which reduces the dimension of the representation data needed by the image. Moreover, due to the block and unit processing method of hog, the relationship between local pixels can be well characterized.

However, the process of feature descriptor acquisition is complex and the dimensionality is high, resulting in slow speed. It's hard to deal with occlusion. In addition, due to the nature of the gradient, it is sensitive to noise.

As shown in Fig. 17, at the beginning of this project, the effect obtained by hog method is not ideal. The result image is not COTS, but black area. We guess that this may be caused by too many negative samples or the dark background of the pictures in this data set. When we tried to solve this problem, we reduced the number of negative samples, but the problem was that this led to more false positives.

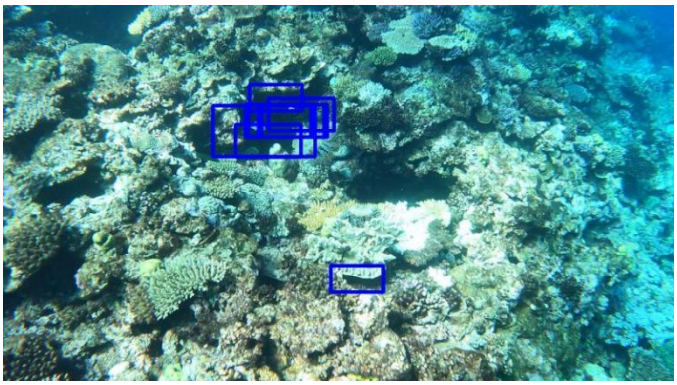
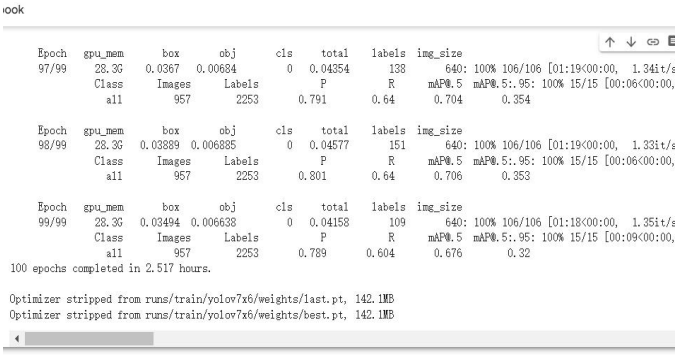


Fig. 17. Processed image

Fast R-CNN integrates the many steps of R-CNN, which effectively improves the detection speed and detection accuracy. However, the region proposal extraction uses selective search, which consumes a lot of time.

YOLOv7 greatly improves the real-time target inference speed and detection accuracy without increasing the inference cost. However, in terms of training, YOLOv7 has higher requirements for computer configuration and video memory. As can be seen from Fig. 18, 28.3G video memory is required in this project. In order to prevent thegraphics card overload, the batch size can only be reduced. We experienced blue screen or sudden disconnection of the computer during training, which may be due to graphics card overload.



```
ook

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     97/99     28.3G    0.0367   0.00684         0   0.04354       138       640: 100% 106/106 [01:19<00:00,  1.34it/s
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 15/15 [00:06<00:00,
                  all        957       2253      0.791       0.64      0.704      0.354

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     98/99     28.3G   0.03889  0.006885         0   0.04577       151       640: 100% 106/106 [01:19<00:00,  1.33it/s
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 15/15 [00:06<00:00,
                  all        957       2253      0.801       0.64      0.706      0.353

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     99/99     28.3G   0.03494  0.006638         0   0.04158       109       640: 100% 106/106 [01:18<00:00,  1.35it/s
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 15/15 [00:09<00:00,
                  all        957       2253      0.789      0.604      0.676       0.32
100 epochs completed in 2.517 hours.

Optimizer stripped from runs/train/yolov7x6/weights/last.pt, 142.1MB
Optimizer stripped from runs/train/yolov7x6/weights/best.pt, 142.1MB
```

Fig. 18. parameter

Fast R-CNN is two-stage, that is, candidate regions are generated first and then classified by CNN, and YOLO is one-stage, that is, the algorithm is directly applied to the input image and the category and corresponding location are output, which is more efficient.

## VI. Conclusion

In this project, we used three different approaches. Through the study of YOLOv7, it is found that it has better flexibility and faster detection speed, which greatly improves the accuracy of detection results. It can efficiently identify COTS in the Great Barrier Reef, Australia.

## References

[1] GP. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 743-761, April 2012, doi: 10.1109/TPAMI.2011.155.

[2] Meijering, E., Dzyubachyk, O., Smal, I. and van Cappellen, W.A., 2009, October. Tracking in cell and developmental biology. In Seminars in cell & developmental biology (Vol. 20, No. 8, pp. 894-902). Academic Press.

[3] Zimmer C, Zhang B, Dufour A, Thébaud A, Berlemont S, Meas-Yedid V, et al. On the digital trail of mobile cells. IEEE Signal Processing Magazine 2006;23:54–623.

[4] Jiang Zuyun,Sun Xiangdong,Wang Xiaochun. Image Defogging Algorithm Based on Sky Region Segmentation and Dark Channel Prior[J]. Journal of Systems Science and Information,2020,8(5).

[5] Jesús Antonio Álvarez Cedillo,Mario Aguilar Fernández,Teodoro Álvarez Sánchez,Raúl Junior Sandoval Gómez. Implementation of a parallel algorithm of image segmentation based on region growing[J]. Eastern-European Journal of Enterprise Technologies,2020,1(9).

[6] Duan Peng,Cheng Wenbo,Qian Qing,Zhang Qiang,Yang Renbing,Pan Yujun. [Overlapping Cervical Cell Image Segmentation Based on Bottleneck Detection and Watershed Algorithm].[J]. Zhongguo yi liao qi xie za zhi = Chinese journal of medical instrumentation,2020,44(1).

[7] Qi Ji,Yang HaiTao. Research on image segmentation and edge detection technology based on computer vision[J]. Journal of Physics: Conference Series,2021,1994(1)

[8] Yang Wei,Cai Lulu,Wu Fei. Image segmentation based on gray level and local relative entropy two dimensional histogram.[J]. PloS one,2020,15(3).

[9] K. Simonyan, and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Computer Science , 2014.

[10] Wang Shijie,Sun Guiling,Zheng Bowen,Du Yawen. A Crop Image Segmentation and Extraction Algorithm Based on Mask RCNN[J]. Entropy,2021,23(9).

[11] N. Dalal, Finding people in images and videos[D]. Institut National Polytechnique de Grenoble-INPG, 2006.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection." 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Vol. 1. Ieee, 2005.

[13] C. J. Burges A tutorial on support vector machines for pattern recognition[J]. Data mining and knowledge discovery, pp121-167,1998.

[14] L. Yulan, M. L. Reyes and J. D. Lee, "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines", Intelligent Transportation Systems IEEE Transactions on, vol. 8, pp. 340-350, 2007.

[15] Wang Yuting,Devji Tahira,Qasim Anila,Hao Qiukui,Wong Vanessa,Bhatt Meha,Prasad Manya,Wang Ying,Noori Atefeh,Xiao Yingqi,Ghadimi Maryam,Lozano Luis Enrique Colunga,Phillips Mark R,CarrascoLabra Alonso,King Madeleine,Terluin Berend,Terwee Caroline,Walsh Michael,Furukawa Toshi A,Guyatt Gordon H. A systematic survey identified methodological issues in studies estimating anchor-based minimal important differences in patient-reported outcomes.[J]. Journal of clinical epidemiology,2021,142.