# Penguins vs Turtles - Image classification with bounding boxes*

*Note: Sub-titles are not captured in Xplore and should not be used

Chongxiao Wang(z5342805)
*z5342805@student.unsw.edu.au*
Hanzhao Lu (z5190746)
*z5190746@student.unsw.edu.au*
Yani Li (z5237308)
*z5237308@student.unsw.edu.au*
Ziwei Han (z5194059)
*z5194059@student.unsw.edu.au*

## I. Introduction

Supervised learning has become widely applied in industries across various fields such as natural language processing, speech recognition, image processing, autonomous driving, and environmental monitoring. This project specifically focuses on the image-based classification of turtles and penguins, which poses interesting scientific challenges. The goal is to develop and evaluate methods for detecting and classifying animals in wildlife images. In this project, our focus is on penguins and turtles, which present challenges due to variations in appearance, environmental backgrounds, sensitivity to light, postures, and occlusions. The trained animal detection model involves two main stages: identifying animal features through images and classifying pictures based on these features for recognition. By training on a dataset of 572 unique images, the computer system overcomes these challenges and achieves accurate analysis of different animal postures. The dataset includes equal proportions of turtle and penguin images, each containing a single object instance. The models used a combination of HOG features and SVM, supplemented by color features and SURF features to improve accuracy. Through various experiments, we optimized the parameters of each method, resulting in significant increases in training model accuracy. LaTeX. Please observe the conference page limits.

## II. Literature Review

The main purpose of this project is to distinguish turtles from penguins in the given images, putting a bounding box around the target and write the corresponding label at its side. Classification problem has been studied by many reseachers and for a long time period. Among such various methodologies that can be utilized for this project, we decide to use Histograms of Oriented Gradients (later referred as HOG) and Support Vector Classification(later referred as SVC) as our base method to perform penguin and turtle classification. HOG is first suggested by Navneet Dalal and Bill Triggs in 2005[11], and after the publication, its outstanding performance has make it one of the most extensively used feature extraction algorithms in computer vision and image based object classification. Although it is first designed to identify human in the image, over the years, it has been proven it can be utilized for most object identification. The solid foundation of HOG feature extract algorithm is that it focus on the local object's shape and appearance, and the very first step of HOG algorithm requires all images change to gray image, which would largely reduced the influences bourght by different colors. And HOG's feature vectors captures gradient changes within a small block of the original image, the vector then can be interpreted as a local intensity gradient or edge directions. These foundenmantal features has make HOG algorithm a robust and efficient method for object detection tasks.

In HOG algorithm, it first computed image gradient to extract the outline shape features within the region, then all pixels in the region will be aissigned to an angle, and then a vector of all these angles will be computed according to the angle values. Then feature vectors will be produce by aggrageting all small regions(typically 8x8 pixels) of images. In addition to eliminate the influences caused by intensity multiplication(intensity multiplication usually happense when the image is not bright enough and researchers multiplied the intensity to have a better visual), local normalization is adopted to get a better performance. At then end, each block feature would be concatenated as one large vector to form the final feature descriptor for this particular image. And this descriptor would have captured all essential shape and appearance information about the major object that is presented in the image.

SVC is a one of the variations under Support Vector Machines(later referred as SVMs)[1], and this methodology is first proposed by Vladimir Vapnik and Corinna Cortes in "Support-vector networks", 1995. In this paper, they have demonstrate that "This paper introduces the support-vector network as a new learning machine for two-group classification problems. ".

In SVC, the essential thought is to find the hyper plane that would classify data points to different classes. The format and shape of the hyperplane is decided by te number of the dimension, if it is a 2 dimension dataset, the pane would be just a line. To decide the optimal division plane, the machine would maximize the margin and thus the capacity would defined

Identify applicable funding agency here. If none, delete this.

the plane. Another way is maximize the separation between support vectors from different classes. This optimization strategy would lead to a daily good performance on unseen data. There are also situations where data can't be linearly separate, SVC would map input training features to a higher-dimendion space, where a separating hyperplane can be identified.

Apart from the HOG and SVC, there are also several new methods that have been proposed alongside the development of computer vision. Deep feature concatenation is one of them. This technique has many variations as it would been modified for different tasks in different projects, each research team would make some slight changes to the actual method to cope with the characteristics of different datasets. However, LeCun has pave the way for this technique[8] in his paper "Deep Learning" in 2015. It has mentioned the importance of hierarchical feature extraction process which is capable of extracting data at different complexity levels.

This technique is still of great importance in recent computer vision research area. During the covid-19 pandemic, one common way to diagnose covid-19 patients is to take Chest X-rays and see if patients' lung has vast white area. During 2021, Emine Cengil and Ahmet Çınar have done a research to evaluate[9] the performance of Depp Feature Concatenation. They have a create two data sets, one set of features is produced by feature transfer method, extracted from AlexNet, Xception, NASNETLarge, and EfficientNet-B0 architectures. And the other training set is completed by concatenating all the features. The proposed pipeline is then applied to various classification algorithms with three different data source, "COVID-19 Image Dataset," "COVID-19 Pneumonia Normal Chest X-ray (PA) Dataset," and "COVID-19 Radiography Database". Among all the algorithms they've applied, each data source have achieved best accuracy of 98.8

Due to the variations of Deep Feature Concatenations among different projects, it is unwise to simply follow other research teams' work, thus we propose our own way of Deep Feature Concatenation to better fit the turtle, penguin dataset, and the method will be explained further in the coming sections.

This project also aims to draw bounding boxes around the object of interests, and Non Maximum Suppresion (NMS) is a technique that has been widely adopted in such research area. It is a prominent algorithm which designed to fix the problem of identifying multiple[5] bounding boxes wound one single object. It essentially uses measurements to evaluate the quality of each bounding boxes for a certain object and then prioritize them according to the confidence score based on some metrics, such as overlapping areas, shape and positions[6]. By removing low confidence score boxes, the interpretability and usability of object detection results have been inhenced.

Hoever, NMS also have some limitations, one of the major issue is that when multiple objects are very close to each other[2], multiple bounding boxes will have a large area overlappingm, and standard NMS would likely to remove too many bounding boxes and potentially removing the boundiung box of valid entity. Severa variations have been proposed to further improve the performance, such as Soft NMS. But considering the nature of our training dataset: each image would only contain one object. Such problem would not exist and using standard NMS would be sufficient to produce a prominent result. And the metrics we've adopted when using the sdandat NMS will be presented in the comming Methodology section.

## III. Method

### *A. Histograms of Oriented Gradients and Support Vector Classification*

For this project, we have utilize the HOG extraction function in skimage library, the HOG function in this library applies an optional global image normalisation equalisation that is designed to reduce the influence of illumination effects. And would finally produced the HOG features of the given image. As for the SVC, we choose to use SVC from sklearn library. However, it is mentioned that, the sklearn version of SVC function has a execution time which scales at least quadratically with the number of samples, and LinearSVC or SGDClassifier should be used if the training dataset is large, in this scenario, we only had 500 training images and 72 validation images, using SVC is appropriate.

*1) Data Preprocessing:* The first step in our methodology involved preparing the image data for further processing. The images were converted to grayscale as the HOG descriptor is sensitive to edges in an image rather than the color information. Images were then resized to a standard resolution to ensure consistency across the dataset. In scenarios where the object detection scale varied, an image pyramid approach was implemented. As mentioned above, HOG algorithm would require all input image to be gray image, which would only have exact one value "intensity", after reading all training images, we transformed them into grayscale for later training. And in HOG algorithm all images would have to be rescaled to a same size, however all the training image in our dataset are all of size 640x640, thus we can then feed images to the HOG function.

*2) Feature Extraction using HOG:* Once all data has been convert to gray scale images, we start to feed images to the HOG function with tune parameters(all parameters we passed into the function are carefully compared and decided) and retrieve each different images' hog features and append them into one list for later training. The HOG in skimage would divide the images to smaller cells and computed a histogram of gradient directions or edge orientations for the pixels within each cell. The resulted histogram would be the descriptor. And all desceriptor would be normalized to improve the accuracy and robustness.

*3) Classification using SVC:* The resulting HOG descriptors served as the input for the next phase of our methodology. Classification will then be performed. We trained an SVC model using the each HOG descriptor that we have generated via HOG process, and each HOG dfescriptor would have an associating class label read from the annotation file where 1 stands for penguin and 2 stands for turtle.

*4) Evaluation:* When evaluation a model's performance, it is usually quantified using standard metrics like accuracy, precision, recall, and F1-score. In order to fetch these information, we can get them via classification report function in sklearn library. This function would print out the precision, recall, F1-score and support information of our model. If further analysis is required, we can evaluate it with different set of training image datasets to access its on-field applicability.

### B. Bounding Box

*1) Find Bounding Box:* After HOG feature extraction and SVC model training, this project also aims to draw bounding boxes around the object that been identified, the bounding box is considered better if it fit the object better. We have derived a function get bounding boxes to determining the appropriate bounding box for each image, the core thoughts about this function is to iterate through different size of bounding boxes and for each windowsize, we iterate through the whole image with that certain windowsize, for each possible image segments, we use HOG function to extract the features and use the classifier that we've trained before to get the prediction score of this segment and then uses decision function of the classifer to get whether this piece is closer to peguin, turtle or none of those two. And we only consider segments with fairly good detection score.

When detecting the appropriate box size, our methods have four loops, which would likey to cause long runtime issues. Improvements can be made such as starting with a large windowsize and downsizing each side accordingly.Also, instead of going through the entire image again, we can iterate through the last bounding box size to speed up the process. After we have all corresponding bounding boxes produced by the above function, we can use functions from open CV library to draw corresponding bounding boxes around our object and put the correct label right beside the image.

*2) Bounding box evaluation:* To evaluate the bounding boxes we have derived, we decided to perform the evaluation from two dimensions, one is the distance "d" from the centre of our bounding box to the actual centre(the actual centre is the centre of the provided bounding box), another metric is given by the formula: $iou = intersectionarea/(pred_area + truearea - intersectionarea)$ This formula essentially measures how similar our predicted bounding box is comparing to the actual bounding box. The distance "d" should be zero if the bounding box is exactly the same to the actual one, but if the predicted bounding box is of the same shape but larger scale, the centre distance would also be zero. That is why we have introduced the second measure "iou", this value is always positive and is bounded within (0,1]. The higher this value approaches 1, the greater the performance it signifies. If the bounding box is exactly the same to the actual box, iou should be 1 accordingly. Thus when evaluating the boxes generated, we should use function calculate distance and iou to calculate the distance and iou for further analysis. On the basis of these evaluation measures, we also calculate the standard deviation and mean value of these two metric for all validation images. It would help us to capture the overall performance of our model in a larger scale and identify any hidden problems or any natural characteristics of our training data set.

### C. Feature concatenation

*1) Motivation:* After analyzing the hog+svm method, certain limitations were identified that could potentially impact our project. Limited feature representation Although HOG features are effective in capturing edge, shape, and texture information, their representation capability is limited. During the image converting process (bgr -¿ gray) some of the details are loss. Lack of semantic understanding HOG features mainly focus on low-level local features, such as the occurrences of gradient orientation in localized portions of an image, and may not capture the semantic meaning of objects. To address the limitations of the HOG and SVM method, the feature concatenation approach is used.

*2) Color histogram:* To improve the feature representation, we decided to use color histograms as the color feature. In this project, the dataset comprises two types of animals, penguins, and turtles. It is observed that penguins are always black or white, whereas turtles can vary in color. Given this diversity in colors, It can be beneficial to use color histograms as color features for classification. This feature contains valuable information pertaining to the color distribution within the images, allowing the classifier to differentiate between the distinctive color properties of penguins and turtles.

To obtain color features for both the training and testing datasets, the get color feature function is implemented. This function takes a list of images as input and returns a list of color histograms. The OpenCV function, cv2.calHist, is used to compute the histograms for the color features. It is important to note that the bin parameter in the cv2.calHist function plays a significant role in the results obtained. The bin size determines the number of bins or intervals into which the color values are divided. Having a small bin size may result in a loss of information. Conversely, using a large bin size may lead to overfitting. To determine a reasonable bin size, several experiments are conducted. Varying bin sizes are tested, and the outcomes are shown below. After chosen the optimal bin size, the color features then append to the hog features, the combined features are used to train the svm model with the same setting above

| ***Bin size*** | ***Accuracy*** |
|---|---|
| 8 | 0.7778[a] |
| 12 | 0.7778[a] |
| 16 | 0.7639[a] |

*3) SURF features:* In order to gain a deeper semantic understanding from the images, it is crucial to incorporate a method that focuses on extracting high-level local features. Therefore, several methods such as SIFT, ORB, SURF, among others, were carefully considered. To determine the most suitable method, a comprehensive evaluation was conducted, taking into account both the runtime efficiency and performance. After evaluating these factors, SURF was selected as the chosen method.

SURF is specifically designed to extract local features from images. It detects and describes interest points based on the presence of blob-like structures. In comparison to low-level features such as color or texture histograms, SURF captures higher-level information by extracting low-level features such as local image gradients, orientations, and scale-space extrema surrounding the interest points and denotes more semantically meaningful concepts[3].

An implemented function named ”get surf features” addresses the issue of varying sizes of SURF features across images in both the training and testing sets. To tackle this problem, a maximum feature size is set as a parameter. If the number of SURF features obtained from an image is less than the maximum feature size, the function pads the feature list with zeros. This ensures that all images have feature vectors of equal length, maintaining consistency and preserving the structure of the data. Conversely, if the number of extracted features exceeds the maximum size, the function handles this by abandoning some of the surplus features. This pruning process eliminates the excess features and maintains conformity with the predetermined maximum feature size. In order to obtain the optimal maximum feature size, several experiments are conducted, and the outcomes are shown below. Based on the results mentioned above, it is evident that the maximum, minimum, and average number of features in both the test and train datasets do not affect the accuracy. However, it is important to note that when the number of features exceed 10000 an increase in the number of features leads to a decrease in accuracy. The highest accuracy is achieved when the max length is set to approximately 5000.

Thus the maximum feature size is set tp 5000, then the SURF features are computed and appended to the combined features list.

| Maximum features size | Value | Accuracy |
|---|---|---|
| **Random picked** | 500 | 0.7778 |
| **Random picked 1000** | 1000 | 0.7917 |
| **Random picked 5000** | 5000 | 0.7917 |
| **Random picked 10000** | 10000 | 0.7778 |
| **Random picked 100000** | 100000 | 0.6806 |
| **Random picked 160000** | 160000 | 0.6944 |
| **Minimal number of features in train dataset** | 2048 | 0.7917 |
| **Average number of features in the train dataset** | 105603 | 0.6806 |
| **Minimal number of features in the test dataset** | 19968 | 0.7778 |
| **Maximal number of features in test dataset** | 282048 | 0.6528 |
| **Average number of features in test dataset** | 101415 | 0.6806 |
| | | |

*4) Evaluation:* The evaluation method remains consistent with the one described in section 3.1.4. It is evident that the accuracy improved after each method was applied.

### *D. Improved non Max Suppression algorithm*

*1) Motivation:* Several tests were conducted on the bounding boxes function, and during this process, it was observed that the bounding boxes with the highest decision score might not be the most optimal one. Therefore, a new method is required to identify the optimal bounding boxes.

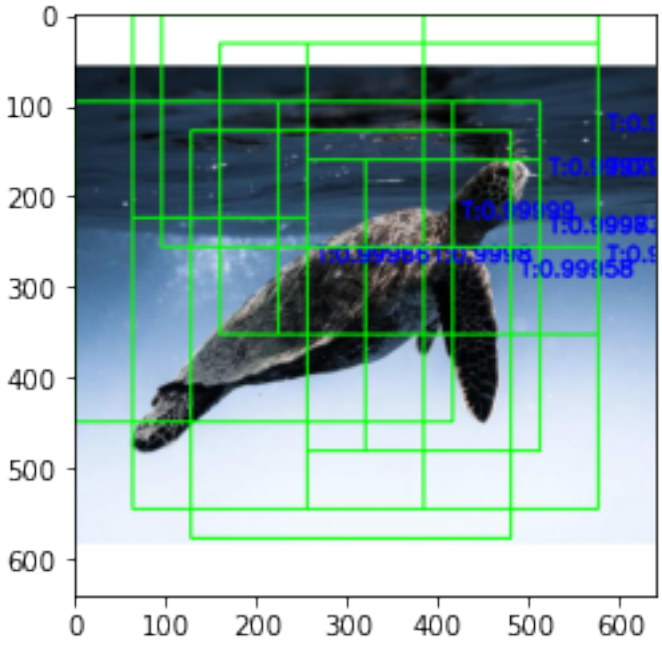


Fig. 1. A list of bounding boxes.

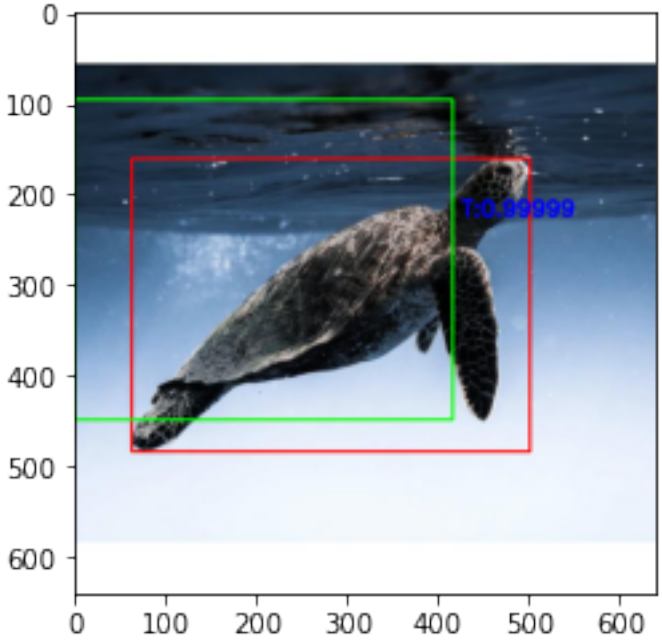


Fig. 2. The bounding box with the highest decision score

*2) Non Max Suppression algorithm:* After conducting thorough research, it was decided to write a Non Max Suppression algorithm which suitable for the project.Non-Maximum Suppression (NMS) is a technique commonly employed in computer vision to choose a solitary entity among multiple overlapping entities, such as bounding boxes in object detection tasks. The typical criterion entails discarding entities with probabilities falling below a predetermined threshold.

The 'non max suppression' function is implemented to obtain the optimal bounding boxes, the function takes in a list of bounding boxes and a specified overlap threshold as input. And perform non-maximum suppression to remove redundant bounding boxes, a simple description of the algorithm. Convert the list of boxes into a numpy array. Init a empty list called 'pick' to store the indexes of selected bounding boxes. Get The coordinates (x1, y1, x2, y2) of the bounding boxes. Compute the area of each bounding box with the formula: (x2 - x1 + 1) * (y2 - y1 + 1). Sort the indexes based on the respective decision scores values. Looping while some indexes still remain in the indexes Select the last index, and appended to the pick list. Initial a list called 'suppress' with the last index. Loop over all indexes in the indexes list. Compute the ratio of overlap between the computed bounding box and the bounding box in the area list If there is sufficient overlap, suppress the current bounding box After loop over all indexes in the indexes list, delete all indexes from the index list that are in the suppression list. Check if there is only one bounding boxes in the pick list, if not compute the average coordinations of the bounding boxes, and return the computed bounding box, and if there is only one bounding boxes in the picked list just return it. The main difference between this algorithm and the traditional non max suppression algorithm is the number of bounding boxes returned by the traditional Non Max Suppression algorithm may be more than one. To deal with this problem, a few attempts have been made, including computing the union, intersection, and averaging the resulting bounding boxes. After compare the mean and standard derivation of the result, averaging the resulting bounding boxes method has been picked.

*3) Evaluation:* The basic evaluation algorithm and method are mentioned in 3.2.1 Bounding box evaluation, in order to evaluate the new method the new method max suppression algorithm and the remove white frame method mentioned below, 2 sets of means and standard derivations are computed, one set with sliding window method and the other set computed by remove the white frame from the image first and then use the max suppression algorithm.

### *E. Remove white frame*

*1) Motivation :* It was observed that some of the bounding boxes contain white frames, that have influence on both distance and the IoU scores, so a method to remove the white frame is implemented.

*2) Remove white frame :* The remove white frame method takes in a list of image, and loop through the image pixel by pixel, to find the non white pixels, and then get the top, bottom, left, right coordinations. A improved get bounding boxes function was implemented, the basic algorithm of the new get bounding boxes function is mentioned above. The new get bounding boxes function takes in an extra parameter image shape, which computed by remove white frame function, so that the new maximum height and weight of the bounding boxes can be computed by top - bottom and right - left, and the loop to scan the whole image can start with the new image shape.

*3) Evaluation:* The evaluation method are mentioned in 3.4.4 Evaluation.

## IV. Experimental Results

### *A. Setups*

*1) Environment:* 1. OS name: Windows 11
2. Processor: Intel Core i7-10750H CPU @ 2.60 GHz, 2592MHz, 6 Cores
3. Dependencies and Packages:
opencv: 3.4.2
Numpy: 1.19.2
Pandas: 1.1.5
Scikit-learn:0.24.2
Scikit-image: 0.17.2

*2) Dataset:* The Penguins vs Turtles image dataset was obtained from Kaggle. We utilized 500 images from the train folder as our training data and 72 images from the valid folder as our test data. The training dataset consists of 250 penguin images and 250 turtle images, while the testing dataset includes 36 penguin images and 36 turtle images.

*3) Model parameter:* For the svm model we just used the default parameter, which is C = 1, kernel='rbf', degree=3, gamma='scale'

*4) Evaluation Metrics:* We used the classification report method in sklearn.metrics, to evaluate the classification result. The classification report computed precision, recall, f1-score and support.

*5) IoU and distance :* We evaluated the image detection results using IoU scores and the distance between the predicted bounding box center and the ground truth bounding box center. To assess the overall performance, we calculated the mean and standard deviation for the distance and IoU scores of 72 images.

### *B. Results*

At first, we were trying to use neural network to judge animal identification precisely, but after attempting, we found that there would be underfitting. Therefore, we decided to use traditional algorithms (HOG features combined with SVM) rather than neural network to achieve animal recognition. Afterwards, we generate quantitative results, which are accuracy, macro average precision, weighted average precision and other metrics.

*1) HOG features and SVM :* After extracting HOG features, we use classification report function in sklearn llibrary to get the standard metrics like accuracy, precision, recall, f1-score and support information. The results are shown in the

following table: We can observe quantitative results from the table above. The accuracy of this model to identify animal species is 0.76. Macro avg and weighted avg refers to the mean average precision, it represents the performance of the model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.83 | 0.67 | 0.74 | 36 |
| 2 | 0.72 | 0.86 | 0.78 | 36 |
| accuracy | | | 0.76 | 72 |
| macro avg | 0.77 | 0.76 | 0.76 | 72 |
| weighted avg | 0.77 | 0.76 | 0.76 | 72 |

*2) Sliding window :* The image detection method is performed using the sliding window function. After obtaining a list of potential bounding boxes, we select the bounding box with the highest decision score. The mean and standard deviation for the IoU scores and distances of the valid dataset are as follows: distance mean:73.706443366432, distance standard deviation: 41.30096480316264, iou mean: 0.4037986778677747, iou standard deviation: 0.26830034113177

*3) Color features :* In get color features(), the function first converts the image to the desired color space to extract the color features. Then append it to the color features list after normalizing the histogram. After the list has been generated, extract the color features from the dataset and combine it with HOG features to training the model. The result of this method is shown in following table: It can be clearly seen that the

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.83 | 0.69 | 0.76 | 36 |
| 2 | 0.74 | 0.86 | 0.79 | 36 |
| accuracy | | | 0.78 | 72 |
| macro avg | 0.79 | 0.78 | 0.78 | 72 |
| weighted avg | 0.79 | 0.78 | 0.78 | 72 |

accuracy increased from 76

*4) Surf features :* In get surf features(), the parameters includes image and max length where max length is the length of features per image. Once the number of surf features obtained from the image is less than max length, the function will pad the shorter arrays with zeros to maintain consistency and preserve the structure of the data. The final step is converting the padded arrays to a numpy array. The progress of training the model is similar as color features. From the results in the table below, it can be seen that the accuracy of training the model with the combination of surf features and HOG features is the highest among these three methods.

*5) Sliding window improvement :* The improved image detection method involves removing the white frame first. Subsequently, the bounding boxes are computed using the resulting image without the white frame. To determine

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.84 | 0.72 | 0.78 | 36 |
| 2 | 0.76 | 0.86 | 0.81 | 36 |
| accuracy | | | 0.79 | 72 |
| macro avg | 0.80 | 0.79 | 0.79 | 72 |
| weighted avg | 0.80 | 0.79 | 0.79 | 72 |

the best bounding box, we employ the non max suppression algorithm. The mean and standard deviation of the valid dataset's IoU scores and distances are as follows: distance mean:61.3076270257409, distance standard deviation: 41.97156388120474, iou mean: 0.45244532656454733, iou standard deviation: 0.36370194587106525

## V. Discussion

### A. HOG features

For image classification mentioned in 4.1, it is clear that the accuracy for penguin classification is higher than the turtle. The reason for this is that compared with penguins, there are significant differences in the frontal and lateral appearance and body shape of turtles. These disparities can lead to the model recognizing the features of turtles in the dataset incorrectly.

### B. Color features

The reason for extracts color features is that it has been observed that in general, penguins appear black and white while turtles may appear in colors ranging from yellow to green. So, adding color histograms as color features can prove beneficial for classification purposes. When using cv2.calHist to calculate the color feature histogram, we found that when the bin size is small, it will cause information loss, and when it is large, there will probably overfitting, which has a negative impact on the recognition and classification results. Therefore, after repeated experiments on the bin size, we determined a reasonable bin size. After using it and hog to train the svm model, the accuracy rate is higher than the previous method.

### C. Surf features

Compared with the first two, the accuracy of using the combination of surf features and HOG features to train the SVM model is the highest. We use the max length parameter when compiling the function so that all images have feature vectors of the same length. However, if the number of extracted features exceeds the maximum size, the function discards some redundant features to handle this. Therefore, the size of max length is very critical. After experiments, we found that when the number of features exceeds 10,000, the increase in the number of features will lead to a decrease in accuracy. And the ideal precision is achieved when max Length is set to 5000. Therefore ,we set the maximum feature size to 5000 to achieve the highest accuracy.

### D. Sliding window

In our image detection process, we utilized a trained model to predict each window and aimed to draw bounding boxes around the detected animals. After obtaining a list of bounding boxes, we selected the one with the highest decision score. We evaluated the results based on the distance between the center of the bounding box and the actual center, as well as the IoU score calculated using the aforementioned formula. The IoU score ranges from 0 to 1, and a higher score indicates better performance. However, after analyzing the results of the model mentioned in section 4.3, we found that the IoU score was unsatisfactory. This is because the bounding box with the highest decision score may not necessarily be the optimal one.

### E. Non max suppression and remove write frame

To improve the accuracy, the non-max suppression algorithm are used to pick the optimal bounding box, and for each image before compute the list of bounding box we remove the white frame first. The result is displayed in 4.3 although the mean and distance for both distance and IoU improved the result was still unsatisfactory. Some of the reasons are listed below. 1. Sliding Windows: The algorithm uses a sliding window approach to scan the image at different positions and scales. This approach may not accurately capture all instances of the object in the image, especially if the object has varying sizes or orientations. 2. Fixed Window Size: The algorithm resizes each window to a fixed size (640x640) in order to get the hog feature with the training data. This could lead to inaccuracies if the object size varies significantly in the image. 3. Single Feature Extraction: The algorithm solely relies on Histogram of Oriented Gradients (HOG) as the feature extraction technique. While this method can be effective for certain objects, it may not perform well for all types of objects or under varying conditions. In the case of turtles, there are significant variations in their frontal and lateral appearances, as well as their body shapes. The get bounding boxes() function aims to iterate through each item in the bounding boxes list and extract the one with the best performance. However, the misclassification of turtles with different perspectives can interfere with the list and subsequently distort our final results.

## VI. Conclusion

### A. Summarise

This paper has discussed the performance of HOG + SVM in image recognition and detection with penguins and turtles. It seems that the discrimination of this method is efficient in the small training data set. And this paper finds adding the features obtained from color histograms and SURF feature to train the model significantly improves the accuracy. The accuracy of this model is up to 79

Using Non-Max Suppression algorithm and removing white frame both have some effectivity to increase iou. However, IoU never achieved the same level of excellence as the penguin pictures in the testing of the turtle pictures.This is because the bounding box method has some drawbacks. The big difference in size between the top and front views pictures of turtles may be one of the reasons.

### B. Recommend future work

*1) Data Preprocessing:* In scenarios with occlusions or overlapping objects, the model might struggle to accurately detect individual objects, leading to low IOU scores. Data augmentation will increase the performance of the bounding box. Increase the diversity of the training dataset by applying data augmentation techniques such as random rotation, scaling, flipping, and brightness adjustments. This helps the model generalize better to different object sizes and orientations, leading to better IoU scores.

In addition, It will be helpful to try some underwater image enhancement processes to reduce the image detection effects of greenish-blue underwater colors. Try using balancing, gamma correction, and sharpening procedures to enhance degraded underwater images. Or try some preprocessing models such as the Revised Imaging Network-model (RIN-model) and the Visual Perception Correction-model (VPC-model)[4].

*2) Hyperparameter Tuning:* Trying the Cross-validation method might increase the performance of our model. In this project, the data with poor IoU and accuracy performance is used as the test set, and the rest of the data is used as the training set. The training set is then used to train the model and the validation set is used to evaluate the performance of the model. By constantly trying different hyperparameters and model configurations, the model parameters that perform best on the validation set can be found.

*3) Post Processing:* Apply a multi-stage post-processing approach that combines multiple post-processing steps to further improve detection accuracy. For instance, combining Weighted Boxes Fusion (WBF)[10] with NMS and removing the white frame. WBF is an integration method that combines the bounding box predictions of multiple target detection models into a single, more accurate prediction.WBF uses confidence and bounding box location information to weigh the fused bounding boxes for more accurate results.

## References


[1] Bidault, F., Despres, C., Butler, C. L. (1995). The search for competence: A struggle for coherence. Journal of Business Research, 34(3), 129-138. doi: 10.1007/bf00994018.

Boesch, G. (2022). What is Computer Vision? The Complete Technology Guide for 2022. [online] viso.ai. Available at: https://viso.ai/computer-vision/what-is-computer-vision/.

CV Low vs. High-Level Features." Baeldung, March 2021. Retrieved from https://www.baeldung.com/cs/cv-low-vs-high-level-features.

Fu, C., Liu, R., Fan, X., Chen, P., Fu, H., Yuan, W., Zhu, M., Luo, Z. (2023). Rethinking general underwater object detection: Datasets, challenges, and solutions. Neurocomputing, 517, 243–256. https://doi.org/10.1016/j.neucom.2022.10.039

Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich

feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 580-587.

Hosang, J., Omran, M., Benenson, R., Schiele, B. (2017). Learning non-maximum suppression. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5723-5732.

Jason Brownlee (2019). A Gentle Introduction to Computer Vision. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/what-is-computer-vision/.

Khan, Muhammad Afzal/(n.d.). Deep Learning. Available at: https://www.researchgate.net/publication/277411157_Deep_Learning (Accessed: 21 November 2021).

Rahman, M. M., Islam, M. Z., Bruce, N. D. (2020). A comprehensive survey on deep anomaly detection in computer vision: Theories, methodologies, and applications. IMA Journal of Management Mathematics, 32(3), 433-474. Available at: https://onlinelibrary.wiley.com/doi/10.1002/ima.22659 (Accessed: 21 November 2021).

Solovyev, R., Wang, W., Gabruseva, T. (2021). Weighted boxes fusion: Ensembling boxes from different object detection models. Image and Vision Computing, 107. https://doi.org/10.1016/j.imavis.2021.104117

Villas-Boas, A. S., Agogino, A. M. (2005). New qualitative evidence on product teams and innovation. IEEE Transactions on Engineering Management, 52(2), 220-233. doi: 10.1109/TEM.2005.847339.