# COMP9517 Group Project Report

Yanjie Zhang, Haijia Wang, Yiping Tang, Hao Wang, Tianyi Bai
*School of Computer Science and Engineering, the University of New South Wales*
*UNSW Sydney Australia, 2052*
{*z5326851, z5415023, z5429550, z5429625, z5430739*}*@ad.unsw.edu.au*

***Abstract*—This project aims to perform object detection and classification on the Penguins versus Turtles dataset. Faster R-CNN model is used as the basic model, supplemented by Mosaic data augmentation to expand the training dataset, and GrabCut algorithm is employed for bounding box correction. By combining all methods (Mosaic and GrabCut) along with the baseline, this project compares four models: (i) YOLOv8 (benchmark); (ii) Faster R-CNN baseline; (iii) Faster R-CNN trained on mosaiced images; and (iv) Faster R-CNN with Bounding Box Correction. The results obtained are as follows: Compared with YOLOv8, Faster R-CNN baseline performs better on most of the validation images. In terms of accuracy, precision, recall, and F1 score, all exceed 94%. In terms of Intersection Over Union (IoU) for the images, most exceed 0.9. Particularly, after using the GrabCut algorithm, there is a significant improvement in predicting the bounding boxes.**

***Index Terms*—Object Detection, Data Augmentation, Faster R-CNN, Bounding Box Correction, Computer Vision**

## I. Introduction

As an important field of artificial intelligence, computer vision has attracted great attention for its wide application in the real world. Object detection and classification in images and videos is one of the core tasks of computer vision, and its applications in surveillance, traffic monitoring, robotics, medical diagnosis, and biology provide us with infinite possibilities [4]. However, due to the huge and complex real-world image data, the accurate, complete, and efficient recognition and analysis of image information only relying on manual work has become insufficient, so a fully automated solution is urgently needed.

As rare marine organisms, penguins and turtles have received extensive attention and protection [14]. They play an irreplaceable role in maintaining the marine ecological balance and are an important part of the marine ecosystem. However, with the impact of climate change and human activity, domestic tourism is likely to be even greater, especially as there are as many as 700 million international trips each year. People like to watch wild animals, bringing new experiences, impressions and emotions, but this will cause great damage to the habitat of wild animals [2]. The number of species and the living environment are facing serious threats. Therefore, monitoring and studying the abundance, distribution and behavior patterns of other wildlife such as penguins and turtles is critical for scientists and conservation agencies. However, for the monitoring and research of these two species in the huge marine ecosystem, the traditional manual identification method faces difficult and inefficient challenges.

In this project, we are required to design and develop a method to identify and locate objects in wildlife images. The given dataset used consists of penguin and turtle images, which were pictured in different environments and published on the Kaggle website. This dataset is divided into training and validation subsets with 500 and 72 images respectively. To avoid imbalance issues, images containing two classes of animals are evenly provided. In the detail of the tasks, our method is expected to output the correct label representing the corresponding animal, and the bounding box as tight to the object as possible for each image. To simulate the real-world scenario, we are only allowed to use images and annotations in the training set to optimize or train our model and evaluate the performance on the validation set. With popular measurement metrics, the scores for both classification and detection tasks should be calculated.

Recently, in the object detection task, a number of deep learning based models are introduced and lead the board of various competitions. Due to the complexity and diversity, the models possess different advantages and shortages but no one can always reach the peak across all domains. Therefore, we analyze and investigate different methods introduced in the course or published by researchers to design a proper model for the given problem.

In this report, we summarize the recent researches that work on the related problems and analyze the features of potential solutions. Based on the data exploration, we give the explanation of why Faster R-CNN [cite] is selected as our basic model. Then, introduce the theories used in this project and the progress of how we develop and enhance our method iteratively. To prove the efficiency, we compare the performance of our method with the benchmark and display sample results got from different methods.

To achieve better performance, we contributed to different parts of the model. Our main work includes:

- Compare performance of Faster R-CNN and YOLOv8;
- Fine-tune the pre-trained model in our dataset;
- Expand training dataset with Mosaic data augmentation;
- Correct bounding boxes with the GrabCut algorithm and edge filtering strategy.

The main content of our report is organized in the following sections. In Section II, the related works are summarized. In Section III, we introduced the primary methods used in the implementation. In Section IV, experimental results and evaluation are illustrated. Finally, we discuss and conclude our project in Sections V and VI.

## II. Related Works

### A. Deep Learning

Deep learning is a class of machine learning methods based on artificial neural networks with multiple layers. It is used to learn knowledge representations from training data and automatically generate analytical models to solve certain challenges. In recent years, deep learning has accumulated tremendous success in a variety of application domains. This novel technique has been growing rapidly and applied to various problems. Natural language processing and computer vision are the two main domains in which deep learning performs well.

*1) Natural Language Processing (NLP):* It aims to solve the problem of communication barriers between humans and machines. Nal et al. [27] defined a new form of CNNs applied in the NLP domain, but for long language sequences, RNN [9] and its variations [5], [28] have more advantages. The Attention mechanism [29] and pre-training method [30] were introduced in 2015 and brought the whole domain to a higher level. In the next years, a large number of models were proposed to address different problems, and Large Language Models (LLM) [31]–[33] became mainstream. Recently, the "pre-train, fine-tune" procedure is gradually replaced by the "pre-train, prompt, and predict" form [34].

*2) Computer Vision (CV):* It is a field of study that designs and develops methods to help computers understand and represent visual semantics in images. Some traditional methods were used before deep learning appears, such as feature extraction [10], support vector machines [11], principal component analysis [12], and so on. Then, the appearance of deep neural networks brings infinite possibilities into this domain. ImageNet [13], VGG [14], GoogleLeNet [16], and residual networks [15] were proposed to address the classification problem and demonstrated excellent performance than humans. After that, researchers focus on investigating powerful deep-learning models that can extract and recognize complex features behind the images [21] and deploy them in various applications [22]. By designing different architectures, the models are capable of addressing various problems, such as image segmentation [17], object detection [18], face recognition [19], and so on.

From the development history of deep learning, researchers usually obtain inspiration from the methods that perform well in some domains and enhance them to fit other applications. In the NLP domain, GPT-4 [20] demonstrates its extraordinary capabilities of understanding human languages based on a large volume of the corpus. However, due to the difficulty of artificial annotations on the images, researchers pay more attention to designing better models with limited datasets.

### B. Object Detection

Object detection is a hot topic in the computer vision world, and the goal is to design and develop a method to identify and locate objects in images. Before the appearance of deep learning, traditional object detection frameworks are generally designed as three steps: (1) frame a part of the graph as a candidate region using sliding windows of different sizes; (2) extract visual features related to the candidate region; and (3) recognize objects with classifiers, e.g. Support Vector Machines (SVM) algorithm. While these conventional approaches might be useful in certain uncomplicated scenarios, they have issues with performance and robustness in more complex circumstances [6]. Since researchers realized the potential of deep learning in computer vision, more complex but accurate models are increasingly grown [26]. In recent years, most models are built on two main architectures: one-stage and two-stage.

*1) One-stage:* The one-stage detectors pass input to an end-to-end model and predict bounding boxes and class probabilities for objects directly without the need to pre-select boxes first. The advantage of this kind of object detection algorithm is that it is very fast and will be better for real-time applications. The classic model is YOLO (You Only Live Once) which predicts objects based on a convolutional neural network that uses the $S \times S$ grid system. Liu et al. [23] introduced the concepts of multi-reference and multi-resolution and developed SSD (Single Shot MultiBox Detector). In 2017, Lin et al. [24] proposed RetinaNet which uses focal loss to learn more information from misclassified samples. Law and Deng [25] defined object detection as a keypoint prediction problem and developed CornerNet using corner points to form the bounding boxes.

*2) Two-stage:* Different from the one-stage models, the two-stage detectors generate candidate frames before performing further classification operations on them. Comparatively, two-stage detectors perform better accuracy but much slower than one-stage detectors. Among them, the RCNN detector [36] is a typical model built on the one-stage architecture and performs great accuracy in various applications. End-to-end detector training on shared convolutional features is made possible by Fast R-CNN [8], which exhibits impressive accuracy and speed. To make the model completely automate, Ren et al. [50] introduced a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network and produces region candidates without manual settings. He et al. [35] proposed spatial pyramid pooling networks (SPPNet) which can generate a fixed-length representation for different sizes of the image/region of interest.

There is no one model that performs excellently in every application so researchers are still exploring potential solutions for specific problems. Generally, one-stage detectors can work better in real-time scenarios than two-stage models because of the architectural structure. In the static datasets, two-stage models have stronger capabilities in bounding box prediction.

### C. Data Augmentation

Data augmentation in computer vision is to extend the dataset by processing or manipulating original images. In certain applications with small training sets, the models, especially deep learning based, usually have poor generalization ability. In these cases, data augmentation is a powerful method

of minimizing the distance between the training and validation set [37]. According to the characteristics of the target scenario, different data augmentation techniques are used to complement the complexity and diversity of training data.

Some data augmentation techniques have been proven useful and are easy to be implemented, such as flipping, rotation, color space transformation, cropping, translation, noise injection, and so on [38]. Nonetheless, the safety issues are still worthy of attention. For example, rotating and flipping 6 and 9 in digit recognition tasks could produce incorrect training data. Apart from regular image manipulations, there are some counter-intuitive methods are introduced. Ionue [40] demonstrated that flipping and mixing average pixels of image pairs is an effective strategy for classification models. Summers and Dinneen [41] introduced a non-linear method to mixing images which results in better performance. Takahashi and Matsubara [42] investigate the feasibility of forming new images by randomly cropping and concatenating multiple images. A similar method is also applied in the implementation of the YOLO series of models [43]. In addition, CutMix [44] was proposed by researchers to effectively guide the model to focus on the less discriminative part of the object, so that the network can have better generalization ability.

Since Generative Adversarial Networks (GANs) were introduced in 2014 [45] and demonstrated great performance in various domains, researchers started investigating their effects in data augmentation [46]. Tero introduced Progressively Growing GANs [47] that have great capabilities to improve image resolutions that allow downstream applications to learn more complex representations. Zhu et al. [49] proposed CycleGAN to finish image-to-image translation. Related work on data augmentation is developed by using Auxiliary Classifier GAN (ACGAN) [1]. This method is able to incorporate class label information when generating synthetic signals, providing richer augmented data for domains such as machine fault diagnosis. All these methods give ideas for image data augmentation.

In summary, data augmentation can help deep neural networks avoid over-fitting to some extent, but can not overcome all biases in limited data applications. Basic image manipulations can generate useful training data that is human observable. GAN-based models can feed complex features to downstream models in an unintuitive way, but sometimes they could be complicated to implement. Like other techniques, data augmentation techniques are domain-dependent as well. To obtain more improvement, it is important to choose efficient techniques before appliance.

### *D. Bounding Box Correction*

In object detection applications, due to limited generalization ability, the models cannot always produce accurate results, including oversized, partial, and false bounding boxes. Bounding Box Correction aims to correct and adjust these imperfect bounding boxes. In fact, it is not a big topic in the academic world as researchers usually incorporate related work into a comprehensive problem, such as object tracking, automated annotation, and so on. In this section, we investigate effective methods that contribute to the improvement of the bounding box and analyze the potential patterns.

To correct the bounding box error, Jiang et al. [39] defined a regression problem for bounding box correction and developed a deep reinforcement learning method that is capable of moving and scaling imperfect boxes to match with correct annotations. Prannay et al. [48] divide RPN proposals into three parts and trained a cascade model to refine high-quality bounding boxes from candidates. Bishwo and Heikki [51] simulated human actions at the experiment stage and proposed an iterative strategy to produce satisfying bounding box annotations for supervised models. Tianxiang et al. [52] proposed a novel network to correct partial bounding boxes in low-shot applications and showed state-of-the-art accuracy on the PASCAL VOC benchmark. Ekaterina and Alexey [53] proposed a bounding box correction framework that can be applied in medical image segmentation to improve the tightness of bounding boxes.

By observing the above works, researchers usually focus on the bounding box correction in cases where small datasets are given. These methods are all built on neural networks which require complicated design and only suit restricted scenarios. Another shortage of deep learning-based models is lacking explainability. In this project, we only have limited resources so it is not realistic to develop a complex model for correcting the bounding box. Then, traditional image processing algorithms become potential solutions and are easier to be observed.

## III. Methods

### *A. Dataset Exploration*

The dataset is the Penguins versus Turtles dataset available from Kaggle (uploaded by ABBY MORGAN in April 2023)[1]. There are 572 images in this dataset, divided into a training folder containing 500 images and a validation folder containing 72 images. The dataset ratio of turtle and penguin images is 50:50. Each image is sized at $640 \times 640$ and contains only one object instance. The accompanying annotation information is a list of dictionaries. The bounding box coordinates are provided in Pascal VOC format. The category IDs include 1 (penguin), and 2 (turtle).

This project consists of two tasks: The first task involves detecting and locating the animal in each image, while the second task involves classifying the animal in each image.

The challenge introduced by this dataset is the variable field of view and resolution. This variability implies that the lighting conditions, perspective, and image quality differ throughout the entire dataset. The images displayed in "Fig. 1" shows these cases.

A further challenge is that the given dataset is too small. 500 training images is a relatively small dataset, especially for the classic object detection methods, which cannot reach the expected result. While for complex deep learning models such as Convolutional Neural Networks (CNNs), thousands

[1]https://www.kaggle.com/datasets/abbymorgan/penguins-vs-turtles

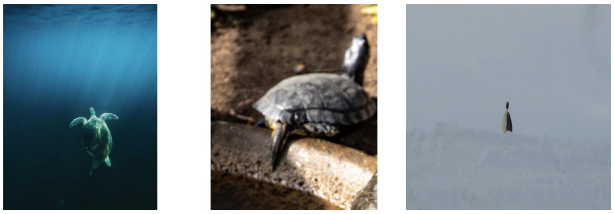

Fig. 1: Example images from the training dataset.

of images are also required as a training set. Deep-learning models have a large number of parameters that need to be optimised, and small datasets can lead to overfitting, poor generalisation, limiting the model's ability to learn, etc. This project primarily focuses on optimizing the Faster R-CNN model and uses YOLOv8 model as a benchmark for comparison. Some effective approaches have been taken to mitigate the problem of small datasets: transfer learning and data augmentation.

### *B. Faster R-CNN*

Faster R-CNN is a classic object detection algorithm proposed by Microsoft researchers in 2015 [7]. Faster R-CNN is developed on the basis of R-CNN and Fast R-CNN algorithms, aiming to solve the two main bottlenecks in traditional object detection algorithms: region extraction and slow object classification. It introduces Region Proposal Network (RPN), a neural network for fast candidate region generation and a fully convolutional network that learns to generate multiple regions of interest (ROI) . These ROI will be input as candidate regions to the subsequent object classification network [7].

"Fig. 2" shows the structure of the Faster R-CNN, the object detection process can be divided into the following steps. Firstly, the input image is passed through the convolutional neural network to get the corresponding image-conv-feature maps. These feature maps contain the semantic information and feature representation of the image. Secondly, after feature extraction, Faster R-CNN introduces RPN for generating candidate object frames. It slides windows on the feature map and performs classification and regression operations on each window to generate multiple candidate object frames. And then, the candidate object boxes generated by RPN are relative to the original image, and in order to correspond to the features on the feature map, the candidate boxes need to be projected onto the feature map to obtain the corresponding feature matrix [7]. Next, the feature matrix corresponding to each candidate frame is scaled to a fixed-size (usually 7x7) feature map through the ROI Pooling layer. The purpose of doing so is to map candidate boxes of different sizes to a fixed-size feature map, which is convenient for subsequent processing. After ROI Pooling, each candidate frame is mapped to a fixed-size feature map. Then, these feature maps are spread and subjected to object classification and bounding box regression operations through a series of fully connected layers. Finally, based on the classification scores and bounding box regression results, the candidate frames are filtered and de-weighted to obtain the final object detection results.

In order to make the final prediction results more accurate, we made adjustments to the classifier layers of the Faster R-CNN only replacing the classifiers from 91 output feature layers to 3 to adapt to our dataset, which represent penguins, turtles, and backgrounds, respectively.

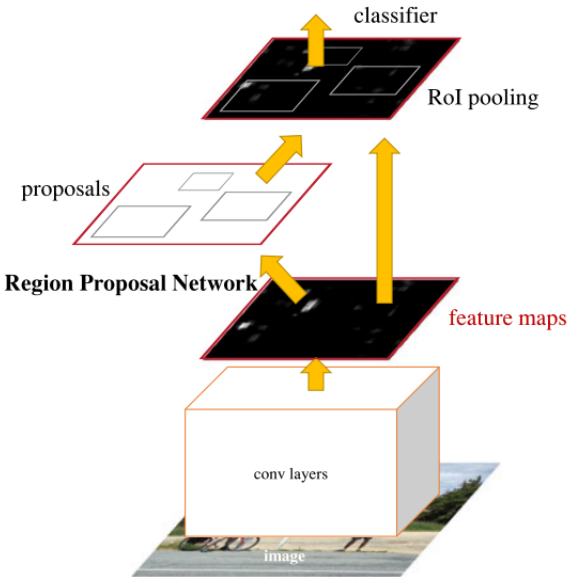


Fig. 2: Faster R-CNN network structure diagram.

### *C. YOLOv8*

The latest YOLO model, YOLOv8[2], was released in January 2023 by Ultralytics, the company that developed YOLOv5. In this project, YOLOv8 is used as a reference for comparison with the detection results of Faster R-CNN. YOLO is well-known for its significant advantage in real-time object detection and has been widely applied in various fields.

"Fig. 3" shows YOLOv8 network structure. Similar to YOLOv5, the architecture consists of a backbone, head, and neck. YOLOv8 uses a similar backbone (including the CSP-Darknet53 structure) as YOLOv5 with some changes on the CSPLayer, which is replaced by the C2f module. The C2f module uses high-level features and contextual information together to improve the accuracy of detection. The size of the convolution kernel in front of each C2f module is 3×3 with stride=2 to make downsampling. At the end of the backbone section, the SPPF (Spatial Pyramid Pooling with Fusion) module is still employed to extract features through three pooling operations.

The feature fusion method used in the neck section remains FPN (Feature Pyramid Network) combined with PAN (Path Aggregation Network). FPN is designed to enable the simultaneous detection of objects of different sizes and is further enhanced with PAN for bottom-up feature propagation. FPN

[2]https://github.com/ultralytics/ultralytics/tree/main

conveys strong semantic features from top to bottom, while PAN conveys strong localization features from bottom to top. YOLOv8 combines both methods to strengthen the feature fusion capability of the network, effectively addressing the multi-scale object detection problem.

In the head section, YOLOv8 utilizes an anchor-free model with decoupled heads, independent of anchors (transitioning from Anchor-Based to Anchor-Free), to handle objectness, classification, and regression tasks separately for bounding box prediction. This design allows each branch to focus on its specific task. With its larger feature map and improved convolutional network, the model becomes more efficient than previous versions, resulting in the overall improvement in accuracy and speed.

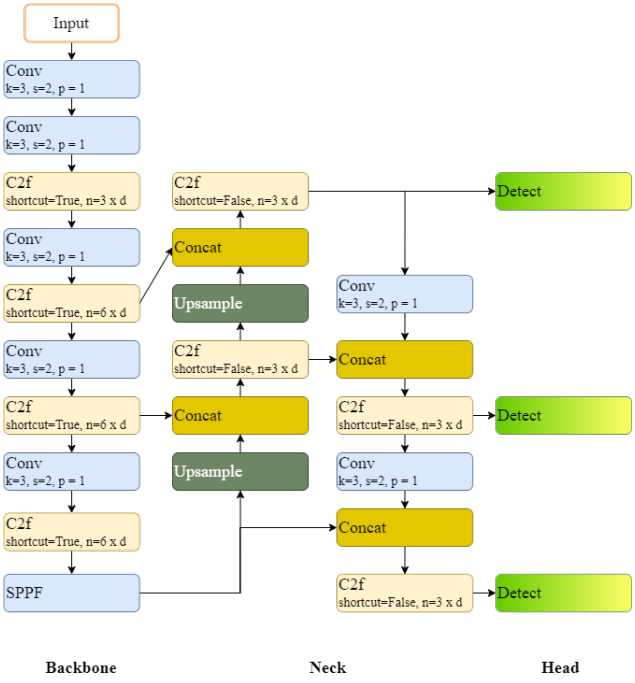


Fig. 3: YOLOv8 network structure diagram.

### D. Data Augmentation

Both Faster R-CNN and YOLOv8 are capable of simultaneously detecting multiple objects in a single image, providing corresponding bounding boxes and class labels for each detected object. In YOLOv8, mosaic data augmentation technique is applied to generate multi-object images for better performance.

The Mosaic data augmentation is an improvement of the CutMix data augmentation. The typical methods of data augmentation involve image transformations such as flipping, color variations, and scaling. However, CutMix data augmentation involves concatenating two images and directly feeding the concatenated image to the neural network for training. Mosaicing is a technique used during the training process to make multiple images into a single image, thereby increasing the diversity of objects and scenes in each training batch. This helps improve the model's generalization capability to different object sizes, aspect ratios, and contexts. This technique is also integrated into Faster R-CNN in this project to enhance the model's training effectiveness.

The images displayed in "Fig. 4" represents mosaiced dataset images. Mosaic data augmentation algorithm generates a lot of images, but mosaiced images may not show the entire appearance of the object. With mosaiced images, the model can systematically learn the object with an unknown full appearance and identify the location and type of object by part of the object's appearance.

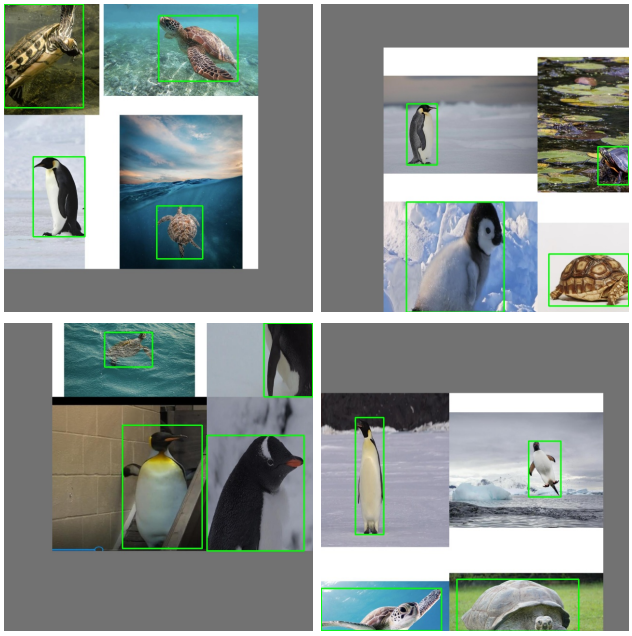

Fig. 4: Mosaiced images.

The following steps are used to create the mosaiced image, given four original images and the desired size ($640 \times 640$) of the generated image:

*1) Resize:* Resize the images to half of their original size ($640 \times 640$ to $320 \times 320$);

*2) Combination:* Make all four resized images into a single image, where each of the four resized images is placed in different corners. And the center of the mosaiced image is randomly chosen to simulate the effect of random cutout;

*3) Place bounding box:* Place the bounding boxes on the new mosaiced image in the correct regions and remove the bounding boxes that aren't in the cutout or occupy less than 25% of the original boxes, resizing any remaining bounding boxes that are cut off by the edge.

### E. Bounding Box Correction

During the training process, there are instances where the bounding boxes in the validation images are inaccurate. To improve the accuracy of predicted bounding boxes, this project aims to apply image segmentation algorithms to separate the object from the background. By using the segmentation results,

the bounding boxes will be corrected, leading to optimized detection results.

In this project, GrabCut algorithm is utilized for bounding box correction in object detection. The method consists of the following steps:

*1) Image Segmentation:* The initial predicted bounding boxes from Faster R-CNN are passed as input to the GrabCut algorithm. Utilizing the iterative graph cuts technique, the algorithm progressively refines the estimation of foreground (object) and background based on the initial input, ultimately completing the image segmentation.

*2) Optimized Bounding Box Generation:* Based on the image segmentation results, we search for the farthest points in each of the four directions from the object and extend these points to create new bounding box boundaries, thus determining the optimized bounding box position.

The images displayed in "Fig. 5" shows the progress of bounding box correction (blue - truth bbox, red - predicted bbox, green - optimized bbox).

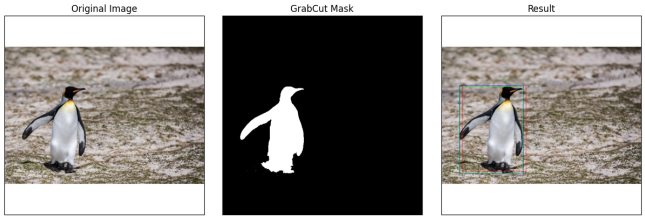


Fig. 5: Bounding Box Correction Process.

To reduce segmentation errors in the GrabCut algorithm, this project should adhere to the following criteria and propose corresponding optimization strategies:

- Good Input: To mitigate the issue of overfitting caused by excessively small bounding box sizes, bounding box expansion is introduced. At the same time, to prevent the object from being excessively surrounded, leading to erroneous segmentation results, the extent of bounding box expansion should be controlled within a reasonable range (0.01-0.09).
- Good Segmentation: GrabCut is a classic bounding box-based segmentation method proposed in 2004. For images with complex backgrounds, the GrabCut algorithm may not adapt well to the complexity of the background, leading to suboptimal segmentation results. Therefore, denoising methods are used to optimize the segmentation results. In this project, the primary approach employed is the dilation technique to fill small gaps in the main regions. The image displayed in "Fig. 6" shows how the dilation technique works: It can remove the false farthest points.
- Good Bounding Box Boundary Selection: To reduce the negative impact of segmentation errors in GrabCut on bounding box correction, an edge filtering strategy is employed for optimization. Under the following two conditions, the predicted bounding box edges should be retained, and no correction is applied:

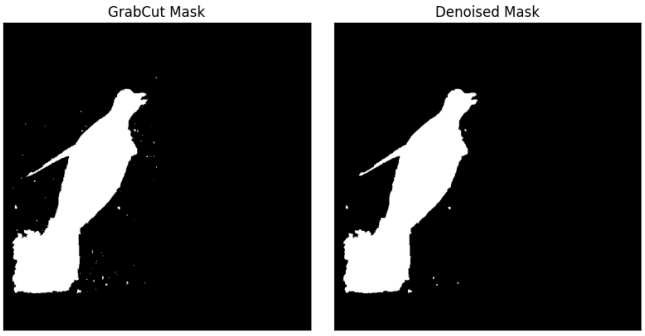


Fig. 6: Segmentation after Noise Reduction.

  - Case 1: The accumulation of the farthest points exceeds 10% of the image boundary (640 pixels), which can be treated as a straight line.
  - Case 2: The difference between the adjusted edge and the corresponding edge of the predicted bounding box exceeds 20% of the length of the predicted bounding box. The prediction results from Faster R-CNN already exhibit a high level of accuracy. Therefore, to further improve the overall accuracy, it is essential to control the correction magnitude within a certain range.

  For example, In "Fig. 7", the segmentation errors are observed on the left and top edges, which appear as nearly straight lines. Therefore, these two edges are ignored for correction, and the bounding box is kept as the predicted bounding box edges.


Fig. 7: Segmentation with Edge Filtering.

## IV. Experimental Results

We conduct the experiments on the Penguin-vs-Turtle dataset. The given dataset is split into train and validation subsets with 500 and 72 images respectively. In all project experiments, only the train subset is used in model training. To evaluate our approach, we calculate scores for classification and detection tasks with multiple popular metrics.

### *A. Implementation Details*

We implement the YOLOv8 model with official source code and configure it to work on the given dataset with only two categories. We implement the second approach with Pytorch Framework [Pytorch paper] based on the official Faster R-CNN baseline model. To accelerate the training process, all models are running on a Colab Jupyter Notebook with a V100

GPU. To demonstrate the best performance of each model, the weights causing the highest scores during training will be saved and used for validation.

**Training:** We use ResNet-50 as the model backbone in Faster R-CNN and train it with the standard configurations. We initialize the weights with the optimal model pre-trained on the COCO2017 dataset to take advantage of prior knowledge. The *learning rate* is 0.005 and *momentum* is 0.9 at the beginning. To avoid over-fitting, we train each model at most 20 epochs.

In Faster R-CNN-based models, we only take the object with the highest confidence score as the result for each image.

### B. Comparison

We compare the performance of all methods mentioned in previous sections. To demonstrate the effects of our strategies, we compare all combinations of methods as well as the baseline, i.e. (i) YOLOv8; (ii) Faster R-CNN baseline; (iii) Faster R-CNN trained on mosaiced images; and (iv) Bounding Box Correction on the model (iii).

**Evaluation Metrics:** We use the prediction results of all models to assess their performance on classification and detection tasks respectively. For the classification task, the evaluation metrics and corresponding formulas are as follows:

$$Accuracy = \frac{TP}{Total} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

where TP/FP are the numbers of True/False positive predictions and TN/FN are the numbers of True/False negative predictions.

In the detection task, the $IoU$ and the distance between the center points of the predicted and truth bounding box are the popular metrics. We calculate the mean value and standard deviation of these two scores to evaluate the overall performance. The relevant mathematical expressions are as follows:

$$IoU_i = \frac{\text{Intersaction Area of } i\text{-th Boxes}}{\text{Union Area of } i\text{-th Boxes}} \tag{5}$$

$$Mean_{IoU} = \frac{1}{N} \sum IoU_i \tag{6}$$

$$STD_{IoU} = \sqrt{\frac{1}{N} \sum (IoU_i - Mean_{IoU})^2} \tag{7}$$

$$Dist_i = \sqrt{(xc_i^{truth} - xc_i^{pred})^2 - (yc_i^{truth} - yc_i^{pred})^2} \tag{8}$$

$$Mean_{Dist} = \frac{1}{N} \sum Dist_i \tag{9}$$

$$STD_{Dist} = \sqrt{\frac{1}{N} \sum (Dist_i - Mean_{Dist})^2} \tag{10}$$

The scores of all methods are shown in TABLE I and the Confusion Matrix of our method is displayed in "Fig. 8":

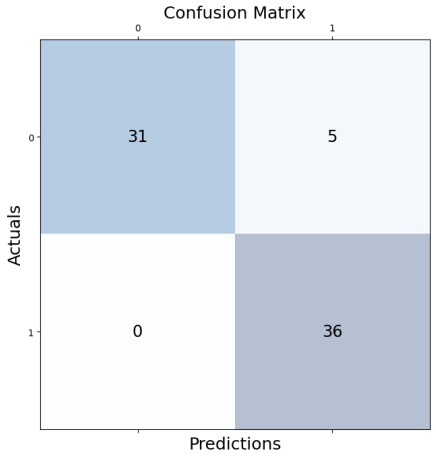


Fig. 8: Confusion Matrix

## V. Discussion

### A. Comparison of Faster R-CNN and YOLO

In terms of recognition and object extraction, in this test, Faster R-CNN demonstrated better recognition accuracy. YOLO has some advantages in image recognition speed, as the RPN method in Faster R-CNN can slow down the recognition process. Additionally, YOLO has some advantages in background discrimination during detection. However, if we apply the YOLO pre-trained model to small datasets, the model may misidentify other objects in the image that have similar features. For example, it might incorrectly identify a sharp black rock as a penguin. Moreover, YOLO's ability to recognize small objects in images is also very poor. In contrast, Faster R-CNN performs well in these aspects. Regarding training, there is not much difference in training time between the two methods, but Faster R-CNN has higher demands on computing power.

### B. Bounding Box Correction

In the set of 72 validation images, we utilized the Faster R-CNN method and produced excellent bounding boxes. In terms of Intersection Over Union (IoU), only 2 images score less than 0.7, 3 are less than 0.8, while the rest of the validation images all exceed 0.8, with the majority even surpassing 0.9. We try adding the GrabCut algorithm to optimize the bounding boxes. This method significantly improves the correction range of the boxes, but the performance on very few images is still poor. The likely reason is that it relies heavily on the initial rectangle frame. If the input box itself is very poor, it significantly affects the segmentation results, such as validation image 30. "Fig. 10" The red boxes represent the ground truth boxes, the blue boxes represent the Faster R-CNN generated boxes, and the green ones are the boxes corrected after GrabCut processing.

Furthermore, it cannot always handle complex backgrounds well, especially in cases where the object parts and the background are similar, like in validation image 24 ("Fig. 11"). However, overall, the cases where the IoU is improved after

TABLE I: Performance Comparison

| **Method Name** | **Classification** | | | | **Detection** | | | |
|---|---|---|---|---|---|---|---|---|
| | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1*** | ***Mean of IoU*** | ***STD of IoU*** | ***Mean of Distance*** | ***STD of Distance*** |
| YOLOv8 | 90.28% | 91.67% | 89.19% | 90.41% | 84.08% | 0.2648 | 47.04 | 126.04 |
| FR baseline | **95.83%** | 97.14% | **94.44%** | **95.77%** | 89.85% | 0.0717 | 10.67 | 10.76 |
| FR+ Mosaic | 93.06% | **100.0%** | 86.11% | 92.54% | 90.86% | **0.0649** | 8.97 | **8.71** |
| FR + Mosaic + Correction | - | - | - | - | **91.39%** | 0.0721 | **7.75** | 8.90 |

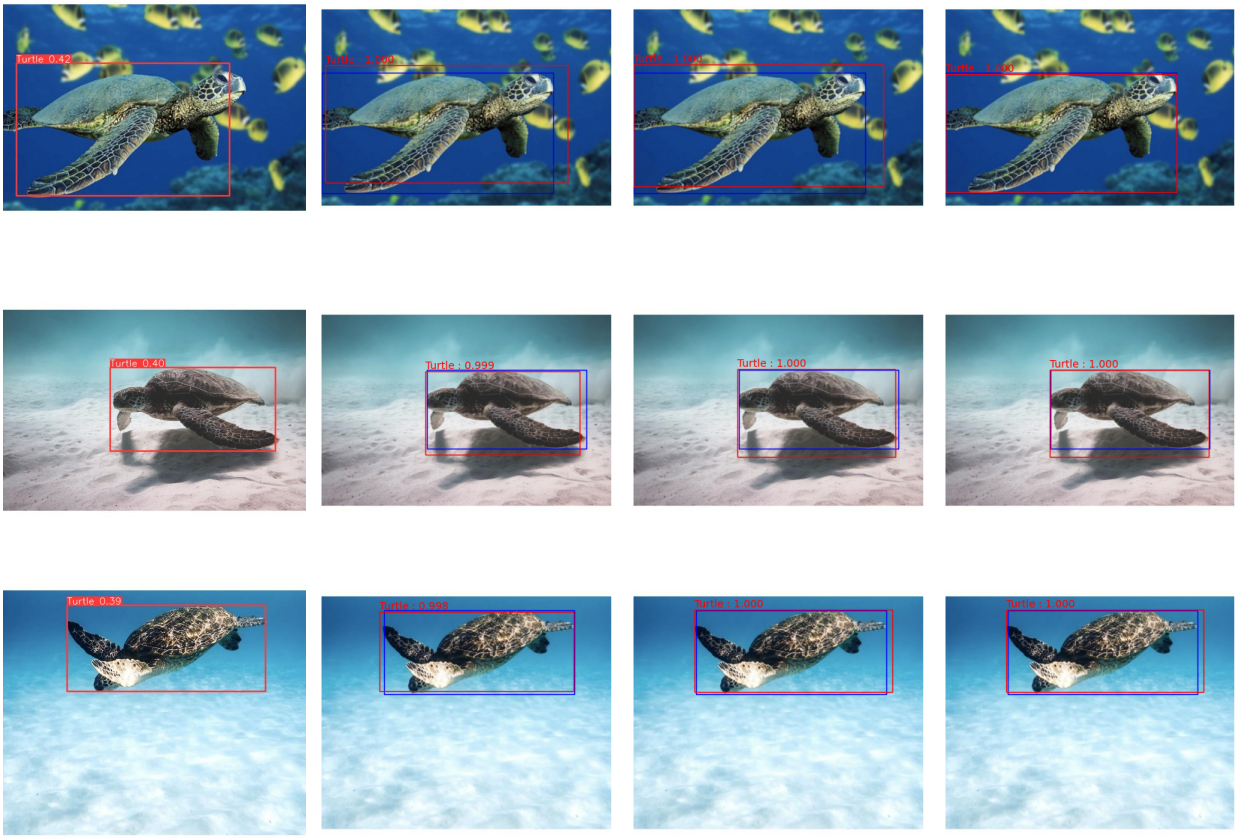


Fig. 9: Comparison of methods

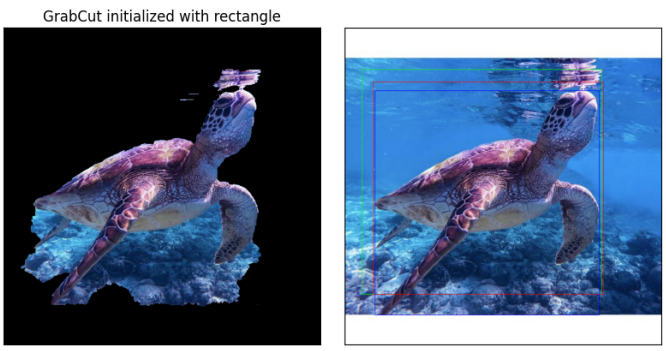


Fig. 10: valid image30

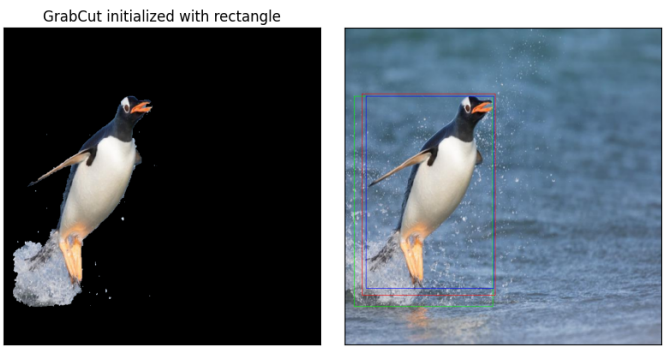


Fig. 11: valid image24

processing comprise the vast majority, such as validation image 19 (“Fig. 12”), which can be seen very intuitively.

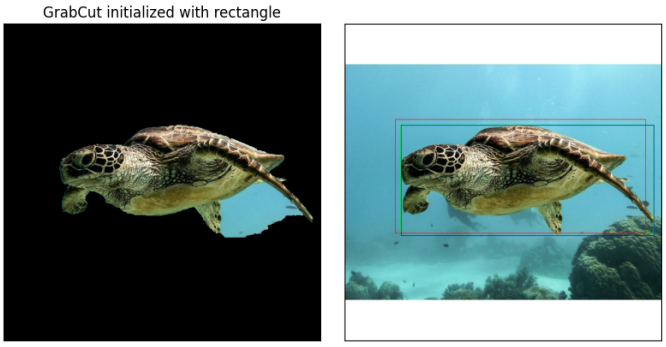


Fig. 12: valid image19

### C. Data Augmentation

Due to the small size of our training set, we shall use image augmentation to expand the training set. However, data augmentation does not always yield positive results. For example, we train the Faster R-CNN using 500 training set images and then with 500 training set images plus 1,000 mosaic images. Comparatively, the former has higher accuracy, recall, and F1 score on 72 validation set images. After applying the mosaic method, only our precision improved, while accuracy, recall, and the F1 score all decreased. Possible reasons include: 1) The abundance of similar samples in the training set may lead to overfitting, resulting in the loss of some important object feature data. 2) The limited number of samples in the training set may increase the difficulty during the training process. The use of data augmentation could potentially hinder model convergence. 3) In some object detection tasks, the orientation of the object may have different meanings, causing adverse effects when using data augmentation.

## VI. Conclusion

The diversity and survival of species are under severe threat. Therefore, monitoring and studying the richness, distribution, and behavioral patterns of various wildlife such as penguins and turtles are crucial for scientists and conservation agencies. In this project, we have adopted the state-of-the-art object detection network "Faster R-CNN" and the "YOLOv8", which perform good results in the identification of penguins and turtles in small datasets.

The experiment takes YOLOv8 as a benchmark, focusing on improving the detection and classification functions of the Faster R-CNN model. This includes but is not limited to mosaic images, the idea of the combination of image segmentation and pattern recognition ideas, and the GrabCut algorithm, to some extent improving our model and results. Looking at the results, the various measurement indicators of Faster R-CNN are very satisfying, with most of them surpassing the YOLOv8 model. In summary, our detection and classification performance is excellent on small datasets that only contain two types of object entities. Among these, due to the program's combination of image segmentation and pattern recognition and the GrabCut algorithm, this project does the best in adjusting the bounding box. However, our project still has some shortcomings inherent in YOLO and Faster R-CNN, such as computing power requirements, training time, and restrictions inherent in the algorithms themselves (e.g. YOLO's handling of small objects, blurred backgrounds and object boundaries, the training and processing time of the large Faster R-CNN model, etc.).

Moving forward, we aim to continue refining our model and exploring solutions to its current limitations, to further push the boundaries of what is possible in wildlife image recognition and analysis.

## Acknowledgment

We hereby express our heartfelt gratitude for the guidance, support, and selfless assistance throughout the process of completing this paper. First and foremost, we extend our thanks to the COMP9517 teaching team for their course instruction and meticulous guidance. Secondly, we would like to thank our tutors, Ye Lin, and Lei Fan. Your vivid explanations and patience in the help sessions helped us to gain a deeper understanding of our knowledge. During the project process, you gave us a lot of valuable advice and suggestions, which benefited me a lot. Finally, thanks to our teammates for working hard on this project.

## References

[1] Shao, S, Wang, P & Yan, R 2019, 'Generative adversarial networks for data augmentation in machine fault diagnosis', Computers in industry, vol. 106, pp. 85–93.

[2] Green, Ronda, and Melissa Giese. 'Negative effects of wildlife tourism on wildlife.' Wildlife tourism: Impacts, management and planning (2004): 81-97.

[3] Scott, R, Hodgson, DJ, Witt, MJ, Coyne, MS, Adnyana, W, Blumenthal, JM, Broderick, AC, Canbolat, AF, Catry, P, Ciccione, S, Delcroix, E, Hitipeuw, C, Luschi, P, Pet-Soede, L, Pendoley, K, Richardson, PB, Rees, AF & Godley, BJ 2012, 'Global analysis of satellite tracking data shows that adult green turtles are significantly aggregated in Marine Protected Areas', Global ecology and biogeography, vol. 21, no. 11, pp. 1053–1061.

[4] Cho, Hyunggi, et al. 'A multi-sensor fusion system for moving object detection and tracking in urban driving environments.' 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014.

[5] Buchipalli, SSR & Thiyagarajan, D 2023, 'Accurate image classification of dogs vs cats using enhanced convolutional neural network in comparison with long short term memory', in AIP Conference Proceedings.

[6] Zheng, R & Qian, W 2021, 'Realization of target detection in near-infrared remote sensing image', in Proceedings of SPIE - The International Society for Optical Engineering, SPIE, p. 117636A–117636A–6.

[7] Shaoqing Ren, Kaiming He, Girshick, R & Jian Sun 2017, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149.

[8] Girshick, Ross. 'Fast r-cnn.' Proceedings of the IEEE international conference on computer vision. 2015.

[9] Thakker, U, Dasika, G, Beu, J & Mattina, M 2019, 'Measuring scheduling efficiency of RNNs for NLP applications',.

[10] Luo, H, Yu, X, Liu, H & Ding, Q 2011, 'A method for real-time implementation of HOG feature extraction', in Proceedings of SPIE, SPIE, Bellingham WA, pp. 819302–819307.

[11] Ben-Hur, A & Weston, J 2010, 'A User's Guide to Support Vector Machines', Data Mining Techniques for the Life Sciences, vol. 609, pp. 223–239.

[12] Abdi, H & Williams, LJ 2010, 'Principal component analysis', Wiley interdisciplinary reviews. Computational statistics, vol. 2, no. 4, pp. 433–459.

[13] He, K, Zhang, X, Ren, S & Sun, J 2015, 'Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification',.

[14] Sengupta, A, Ye, Y, Wang, R, Liu, C & Roy, K 2019, 'Going Deeper in Spiking Neural Networks: VGG and Residual Architectures', Frontiers in neuroscience, vol. 13, pp. 95–95.

[15] Guan, H, Shen, X, Lim, S-H & Patton, RM 2018, Composability-Centered Convolutional Neural Network Pruning, United States.

[16] Khan, R. U., Zhang, X., & Kumar, R. (2019). 'Analysis of ResNet and GoogleNet models for malware detection', Journal of Computer Virology and Hacking Techniques, 15, 29-37.

[17] Zhang, H, Fritts, JE & Goldman, SA 2008, 'Image segmentation evaluation: A survey of unsupervised methods', Computer vision and image understanding, vol. 110, no. 2, pp. 260–280.

[18] Gidaris, S & Komodakis, N 2015, 'Object Detection via a Multi-region and Semantic Segmentation-Aware CNN Model', in 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, pp. 1134–1142.

[19] Zhao, W, Chellappa, R, Phillips, P & Rosenfeld, A 2003, 'Face recognition', ACM computing surveys, vol. 35, no. 4, pp. 399–458.

[20] Waisberg, E, Ong, J, Zaman, N, Kamran, SA, Sarker, P, Tavakkoli, A & Lee, AG 2023, 'GPT-4 for triaging ophthalmic symptoms', Eye (London).

[21] Voulodimos, A, Doulamis, N, Doulamis, A & Protopapadakis, E 2018, 'Deep Learning for Computer Vision: A Brief Review', Computational intelligence and neuroscience, vol. 2018, pp. 7068349–13.

[22] Fang, W, Ding, L, Love, PED, Luo, H, Li, H, Peña-Mora, F, Zhong, B & Zhou, C 2020, 'Computer vision applications in construction safety assurance', Automation in construction, vol. 110, p. 103013.

[23] Liu, W, Anguelov, D, Erhan, D, Szegedy, C, Reed, S, Fu, C-Y & Berg, AC 2016, 'SSD: Single Shot MultiBox Detector'.

[24] Lin, T-Y, Goyal, P, Girshick, R, He, K & Dollar, P 2020, 'Focal Loss for Dense Object Detection', IEEE transactions on pattern analysis and machine intelligence, vol. 42, no. 2, pp. 318–327.

[25] Law, H & Deng, J 2020, 'CornerNet: Detecting Objects as Paired Keypoints', International journal of computer vision, vol. 128, no. 3, pp. 642–656.

[26] Zhao, Z-Q, Zheng, P, Xu, S-T & Wu, X 2019, 'Object Detection With Deep Learning: A Review', IEEE transaction on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232.

[27] Kalchbrenner, N, Grefenstette, E & Blunsom, P 2014, 'A Convolutional Neural Network for Modelling Sentences',.

[28] Chung, J, Gulcehre, C, Cho, K & Bengio, Y 2014, 'Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling',.

[29] Chorowski, J, Bahdanau, D, Serdyuk, D, Cho, K & Bengio, Y 2015, 'Attention-Based Models for Speech Recognition',.

[30] Wang, Y, Song, D, Wang, W, Rao, S, Wang, X & Wang, M 2022, 'Self-supervised learning and semi-supervised learning for multi-sequence medical image classification', Neurocomputing (Amsterdam), vol. 513, pp. 383–394.

[31] Brown, TB, Mann, B, Ryder, N, Subbiah, M, Kaplan, J, Dhariwal, P, Neelakantan, A, Shyam, P, Sastry, G, Askell, A, Agarwal, S, Herbert-Voss, A, Krueger, G, Henighan, T, Child, R, Ramesh, A, Ziegler, DM, Wu, J, Winter, C, Hesse, C, Chen, M, Sigler, E, Litwin, M, Gray, S, Chess, B, Clark, J, Berner, C, McCandlish, S, Radford, A, Sutskever, I & Amodei, D 2020, 'Language Models are Few-Shot Learners',.

[32] Touvron, H, Lavril, T, Izacard, G, Martinet, X, Lachaux, M-A, Lacroix, T, Rozière, B, Goyal, N, Hambro, E, Azhar, F, Rodriguez, A, Joulin, A, Grave, E & Lample, G 2023, 'LLaMA: Open and Efficient Foundation Language Models',.

[33] Scao, TL, Fan, A, Gallé, M, Webson, A, Wang, T, Bekman, S, et al. 2022, 'BLOOM: A 176B-Parameter Open-Access Multilingual Language Model',.

[34] Liu, P, Yuan, W, Fu, J, Jiang, Z, Hayashi, H & Neubig, G 2023, 'Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing', ACM computing surveys, vol. 55, no. 9, pp. 1–35.

[35] He, K, Zhang, X, Ren, S & Sun, J 2015, 'Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition', IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904–1916.

[36] Uijlings, J., van de Sande, KE., Gevers, T & Smeulders, AW. 2013, 'Selective Search for Object Recognition', International journal of computer vision, vol. 104, no. 2, pp. 154–171.

[37] Shorten, C & Khoshgoftaar, TM 2019, 'A survey on Image Data Augmentation for Deep Learning', Journal of big data, vol. 6, no. 1, pp. 1–48.

[38] Shorten, C & Khoshgoftaar, TM 2019, 'A survey on Image Data Augmentation for Deep Learning', Journal of big data, vol. 6, no. 1, pp. 1–48.

[39] Jiang Y, Shin H, Ko H. Precise regression for bounding box correction for improved tracking based on deep reinforcement learning[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 1643-1647.

[40] Inoue, H 2018, 'Data Augmentation by Pairing Samples for Images Classification',.

[41] Summers, C & Dinneen, MJ 2018, 'Improved Mixed-Example Data Augmentation',.

[42] Takahashi, R, Matsubara, T & Uehara, K 2020, 'Data Augmentation Using Random Image Cropping and Patching for Deep CNNs', IEEE transactions on circuits and systems for video technology, vol. 30, no. 9, pp. 2917–2931.

[43] Nepal, U & Eslamiat, H 2022, 'Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs', Sensors (Basel, Switzerland), vol. 22, no. 2, p. 464.

[44] Yun, S, Han, D, Oh, SJ, Chun, S, Choe, J & Yoo, Y 2019, 'CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features',.

[45] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[46] Frid-Adar, M, Diamant, I, Klang, E, Amitai, M, Goldberger, J & Greenspan, H 2018, 'GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification', Neurocomputing (Amsterdam), vol. 321, pp. 321–331.

[47] Karras, T, Aila, T, Laine, S & Lehtinen, J 2017, 'Progressive Growing of GANs for Improved Quality, Stability, and Variation',.

[48] Karras, T, Aila, T, Laine, S & Lehtinen, J 2017, 'Progressive Growing of GANs for Improved Quality, Stability, and Variation',.

[49] Zhu, J-Y, Park, T, Isola, P & Efros, AA 2017, 'Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks',.

[50] Shaoqing Ren, Kaiming He, Girshick, R & Jian Sun 2017, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149.

[51] Adhikari, B & Huttunen, H 2020, 'Iterative Bounding Box Annotation for Object Detection',.

[52] Pan, T., Wang, B., Ding, G., Han, J. and Yong, J.H., 2019, August. 'Low Shot Box Correction for Weakly Supervised Object Detection'. In IJCAI, pp. 890-896.

[53] Redekop, E & Chernyavskiy, A 2021, 'Medical image segmentation with imperfect 3D bounding boxes',.