JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Identification of Fish Species using YOLOv5

[1]Nikita P. Mohod, [2]Gauri H. Deshpande, [3]Aditya C. Jagdale, [4]Vedant S. Talokar, [5]Ishika R. Daga,

[6]Manjiri M. Deole

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student, [6]Student

[1]Computer Science and Engineering,

[1]Sipna College of Engineering and Technology, Amravati, India

Abstract : The research paper provides an overview of various object detection algorithms used in computer vision. These algorithms include deep learning-based methods like Single Shot Detector (SSD), Region-based Convolutional Neural Networks (R-CNNs), You Only Look Once (YOLO), and Objects as Points (OAP). The algorithms are designed to detect objects within images and videos and achieve high accuracy and speed. The information provided in this research can be used as a resource for researchers and practitioners in the field of computer vision. This research paper also provides a discussion about object detection using YOLOv5 and deep learning models in computer vision. The paper covers the preparation of a custom dataset using Roboflow, training of the YOLOv5 model using PyTorch, and evaluating the model's performance using the mean average precision (mAP) metric. The results of the case study show that the trained YOLOv5 model achieved a high mAP of 0.614, indicating its ability to detect objects with high accuracy.

IndexTerms **- SSD, R-CNN, YOLO, YOLOv5, PyTorch.**

I. INTRODUCTION

The field of computer vision has experienced rapid growth in recent years and involves the development of artificial intelligence algorithms that can analyze and interpret visual data. Computer vision is an interdisciplinary field that involves computer science, mathematics, and cognitive science, and it has various applications, such as in autonomous vehicles, facial recognition, medical imaging, and augmented reality.Computer vision involves the use of various algorithms to analyze visual data. These algorithms can be classified into two categories: traditional computer vision algorithms and deep-learning based algorithms. Traditional computer vision algorithms are based on handcrafted features and models and are often used for low-level tasks, such as image filtering, feature extraction, and segmentation. Examples of traditional computer vision algorithms include edge detection, corner detection, and Hough transforms. In recent years, deep learning-based algorithms have gained significant attention in the field of computer vision. Deep learning-based algorithms are based on neural networks and can learn to perform various tasks, including object detection, image classification, and segmentation. The most commonly used deep learning-based algorithm in computer vision is the convolutional Neural network (CNN). CNNs have revolutionized the field of computer vision by enabling machines to learn complex features from images. Computer vision has various categories and types of algorithms that are designed to enable machines to process and extract information from images and videos. These algorithms can be classified into three categories: low-level, mid-level, and high-level vision. Low-level vision algorithms focus on basic image processing tasks such as image filtering, image segmentation, and feature detection. These algorithms are useful for enhancing image quality and detecting basic features like edges and corners. Mid-level vision algorithms are used to extract more complex features from images, such as object recognition and motion analysis. These algorithms use techniques like optical flow analysis, stereo vision, and structure from motion. High-level vision algorithms interpret images in terms of their semantic content. These algorithms use machine learning techniques to identify objects and scenes in images and videos. Examples of high-level vision algorithms include image classification, object detection, and semantic segmentation. Computer vision has many applications in various industries. Some of the most common applications of computer vision include Healthcare, Manufacturing, Automotive, Surveillance, Facial Recognition, Augmented Reality, etc.

Despite the significant progress made in the field of computer vision, there are still many challenges that need to be addressed. One of the main challenges in computer vision is the lack of annotated data. Deep learning-based algorithms require large amounts of annotated data for training, and obtaining such data is often time-consuming and expensive. Another challenge is the issue of bias in datasets. Datasets can be biased towards certain classes or demographics, which can lead to biased algorithms. Finally, the issue of explainability is another challenge in computer vision. Deep learning-based algorithms are often considered black boxes, and it is difficult to understand how they make decisions.

Object Detection is one of the fundamental tasks of computer vision, and it involves identifying the location and category of objects within an image or video. YOLO (You Only Look Once) is an object detection algorithm that has gained a lot of popularity in recent years due to its speed and accuracy. YOLOv5 is the latest version of the YOLO algorithm and builds on previous version to improve speed and accuracy further. In this review paper, we will discuss computer vision and object detection and explore the YOLO algorithm and its latest version, YOLOv5 with a case study on a Fish Species Dataset.

II. Related Work

2.1 Object Detection:

Object detection is a subfield of computer vision that involves identifying and locating objects of interest within an image or video frame. The task of object detection involves two major steps: object localization and object classification.Object localization refers to the process of identifying the location of objects within an image or video frame. This is typically done by drawing a bounding box around the object or objects of interest. Object classification involves determining what type of object is present within the bounding box.Another important aspect of object detection is the use of deep learning techniques such as convolutional neural networks (CNNs) to automatically learn the features that are used to identify objects within an image. Deep learning has led to significant improvements in object detection accuracy and has enabled the development of object detection models that are capable of detecting multiple object types within an image or video frame.Object detection has numerous applications, including surveillance, autonomous vehicles, and robotics. It plays a crucial role in enabling machines to interpret and understand visual data, and has the potential to revolutionize a wide range of industries.There are various methods for object detection, including both two-stage and one-stage methods. Two-stage methods first generate a set of object proposals, which are regions likely to contain objects, and then classify each of the proposed regions. One-stage methods, on the other hand, directly predict the bounding boxes and class probabilities for each object within an image. We will discuss these methods in detail in this research.

Furthermore, we will review the most common approaches for object detection, including classical methods region based methods and region free methods and their various popular algorithms such as YOLO, SSD, and Faster R-CNN. We will also discuss the advantages as disadvantages of these methods, as well as their applications and future research directions.

One-Stage Methods:

One-stage object detection methods are a popular approach in computer vision for detecting objects within images. These methods are commonly used for real-time object detection applications, where speed and accuracy are both critical factors. One-stage object detection methods operate by applying a single convolutional neural network (CNN) to the entire image, and then using the output of this network to predict the location and class of objects within the image. This approach is in contrast to two-stage object detection methods, which typically involve a separate region proposal step followed by a classification step. One-stage object detection methods are generally faster than two-stage methods since they avoid the need for a separate region proposal step. However, they can sometimes suffer from lower accuracy due to the challenge of detecting objects at different scales and locations within the image. Region free methods are examples of one-stage OD methods.

Region-Free Object Detection: Region-free object detection methods, also known as anchor-free methods, are a recent development in the field of computer vision. Unlike region-based methods that rely on predefined anchor boxes to detect objects in an image, region-free methods do not require any prior knowledge of the object size or shape. Instead, they use pixel-level features to predict the object bounding boxes and class probabilities directly. Region-free object detection methods can be divided into two main categories: regression-based and classification-based methods. In regression-based methods, the model directly regresses the object bounding box coordinates and objectness score without using any predefined anchors. In contrast, classification-based methods first predict the objectness score for each pixel and then group the pixels together to form the object bounding box. Region-free object detection methods have several advantages over region-based methods. First, they are more flexible and can adapt to objects of varying sizes and shapes. Second, they can detect small objects more accurately as they do not rely on predefined anchor boxes. Third, they are computationally more efficient as they do not require anchor box generation and matching. YOLO is one of the most popular region-free object detection methods. It divides the input image into a grid and predicts the bounding boxes and class probabilities for each grid cell. Each grid cell is responsible for predicting a fixed number of bounding boxes, regardless of the number of objects in that cell. The bounding boxes are parameterized relative to the grid cell, which makes the predictions invariant to the location of the object within the cell. YOLO uses a single CNN to make predictions for all objects in the image, which makes it very fast. SSD is another region-free object detection method that is similar to YOLO in that it uses a single CNN to predict the bounding boxes and class probabilities for all objects in the image. However, SSD uses a different approach for generating bounding box proposals. Instead of dividing the image into a grid, SSD applies a set of default bounding boxes of different scales and aspect ratios to the feature map generated by the CNN. These default boxes are then refined to better match the objects in the image.

Two-Stage Object Detection:

Two-stage object detection methods are one of the popular approaches to object detection in computer vision. The method consists of two stages: region proposal and object classification. In the first stage, the method proposes candidate object regions that are likely to contain objects of interest. In the second stage, the method classifies the proposed regions into object categories. The region proposal stage is typically performed using a method called selective search, which is a hierarchical clustering algorithm that groups image pixels into larger segments based on similarity in color, texture, and location. These segments are then used as candidate regions for further processing. In the second stage, features are extracted from the proposed regions and used to classify the objects. The features can be extracted using a convolutional neural network (CNN), and the classification can be performed using techniques such as support vector machines (SVMs), decision trees, or neural networks. Two-stage object detection methods have shown excellent performance on object detection tasks, particularly on datasets with a large number of object categories and complex object layouts.

However, two-stage object detection methods are computationally expensive due to the need for region proposal, which can limit their use in real-time applications. To address this issue, one-stage object detection methods have been developed, which perform region proposal and object classification in a single pass. Region based methods are examples of two-stage OD methods.

Region-Based Object Detection: Region-based object detection methods are two-stage approaches that rely on generating region proposals and then classifying those proposals as containing an object of interest or not. These methods have shown significant improvements in accuracy compared to one-stage methods. In the first stage, these methods use a region proposal network (RPN) to generate a set of object proposals. The RPN takes an image as input and produces a set of bounding box proposals, each associated with an objectness score. The objectness score indicates how likely it is that the proposal contains an object of interest. In the second stage, the region proposals are classified using a deep neural network. The network takes the proposed region as input and produces a score for each object class. The highest-scoring class is assigned to the region proposal,

and the bounding box coordinates are adjusted to more accurately fit the object. One of the most popular region-based object detection methods is Faster R-CNN. It uses an RPN to generate region proposals and a Fast R-CNN network for object classification. The RPN is trained to generate high-quality proposals that can be efficiently processed by the Fast R-CNN network. Another popular region-based method is Mask R-CNN, which extends Faster R-CNN by adding a branch for predicting object masks. This allows the network to not only detect objects but also segment them from the background. Region-based object detection methods have achieved state-of-the-art results on benchmark datasets such as COCO and PASCAL VOC. However, they are generally slower than one-stage methods due to the additional computation required for generating region proposals.

Comparison:

Both one stage and two stage object detection methods have their advantages and disadvantages. One stage methods, such as YOLO, are faster and more efficient, but may sacrifice some accuracy in detecting small objects or objects with occlusions. Two stage methods, such as Faster R-CNN, are more accurate and better suited for complex scenes with many small objects, but are slower and require more computational resources. Overall, the choice of which method to use depends on the specific application and the tradeoff between speed and accuracy. In scenarios where real-time processing is critical, one stage methods may be preferred, while in situations where high accuracy is paramount, two stage methods may be more suitable.

Table 1: Comparison of methods

Object Detection Method	Region based/free	Speed	Accuracy
One-Stage OD	Region-Free	Fast	Lower
Two-Stage OD	Region-Based	Slow	Higher

2.2 YOLO algorithm

You Only Look Once (YOLO) is an object detection algorithm that has gained a lot of popularity in recent years due to its speed and accuracy. YOLO was first introduced in a 2016 paper by Joseph Redmon et al. The algorithm is based on a single neural network that takes an entire image as input and predicts the locations and categories of objects in the image in one go. This is in contrast to traditional object detection algorithms that apply a classifier to a series of subregions of the image. The YOLO algorithm consists of two main components: a convolutional neural network (CNN) and a bounding box predictor. The CNN processes the input image and extracts a feature map that represents the image at different scales. The bounding box predictor then takes this feature map and outputs a set of bounding boxes, each associated with a specific object class and a confidence score. The confidence score represents how confident the algorithm is that the predicted bounding box contains an object of the associated class. One of the key advantages of YOLO is its speed. Since the algorithm processes the entire image at once, it can achieve real-time object detection on a single GPU. YOLO is also highly accurate, achieving state-of-the-art performance on several benchmark datasets. The algorithm is robust to occlusion and small object detection, which can be challenging for other object detection algorithms. YOLO has been used in a wide range of applications, including autonomous driving, surveillance, and robotics. The algorithm's speed and accuracy make it ideal for real-time applications, and its ability to detect multiple objects in a single pass makes it well-suited for scenarios where efficiency is critical. Despite its strengths, YOLO does have some limitations. One of the main challenges with YOLO is that it can struggle with detecting small objects, as the bounding boxes generated by the algorithm can be larger than the object itself. The algorithm also struggles with detecting objects that are closely packed together, such as crowds of people. To address some of these limitations, the latest version of YOLO, YOLOv5, was introduced in 2020. YOLOv5 builds on previous versions of the algorithm to improve speed and accuracy further. The new version uses a more powerful backbone network and introduces several new features, including multi-scale prediction, focal loss, and anchor-based box prediction. These improvements have resulted in even faster and more accurate object detection. In summary, YOLO is a highly efficient and accurate object detection algorithm that has gained widespread popularity in recent years. Its ability to process entire images in a single pass and detect multiple objects at once make it well-suited for real-time applications. While YOLO has some limitations, the latest version of the algorithm, YOLOv5, addresses many of these limitations and further improves speed and accuracy.

2.3 YOLOv5

YOLOv5 is the latest version of the You Only Look Once (YOLO) object detection algorithm. It was introduced in 2020 by Ultralytics, and builds on previous versions of YOLO to improve speed and accuracy. YOLOv5 uses a more powerful backbone network, called CSPNet, which allows for faster and more efficient computation of features from the input image.

One of the key features of YOLOv5 is its multi-scale prediction capability, which allows the algorithm to detect objects at different scales within an image. This is achieved by using feature maps at different scales to predict object bounding boxes and their associated classes. Another important improvement in YOLOv5 is the use of anchor-based box prediction, which helps to improve the accuracy of object detection. This approach involves defining a set of anchor boxes of different sizes and aspect ratios, which are used to predict the bounding boxes of objects within an image. YOLOv5 also introduces focal loss, which is a loss function that assigns higher weights to hard-to-detect objects. This helps to improve the accuracy of the algorithm by focusing on objects that are more challenging to detect. Overall, YOLOv5 offers faster and more accurate object detection than previous versions of YOLO, and has been used in a wide range of applications, including autonomous driving, robotics, and surveillance.

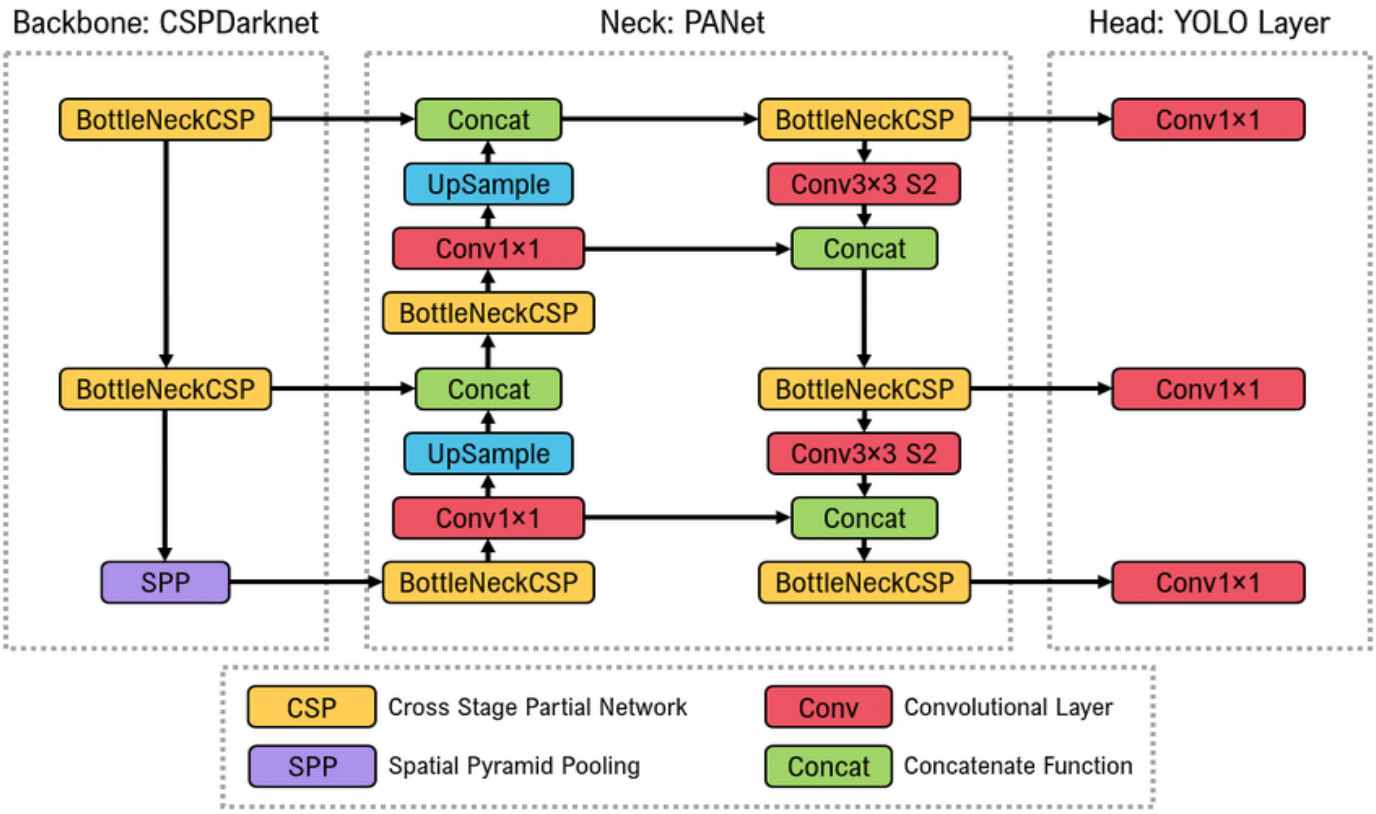

Fig 1: YOLOv5 Architecture

2.4 Performance of YOLOv5

YOLOv5 has achieved state-of-the-art performance on several object detection benchmarks, including COCO, PASCAL VOC, and Open Images. On the COCO benchmark, YOLOv5x achieved a mean average precision (mAP) of 55.4%, which is the highest mAP score achieved by any object detection algorithm to date. YOLOv5 has also achieved real-time performance on a range of hardware platforms, including CPUs, GPUs, and specialized hardware such as the Nvidia Jetson. YOLOv5 has also been used in several real-world applications, such as autonomous driving and object tracking. In a study conducted by Intel, YOLOv5 was used to detect objects in real-time for autonomous driving. The study found that YOLOv5 achieved high accuracy and low latency, making it suitable for real-time applications. YOLOv5 has also been used for object tracking in surveillance systems, where it has achieved high accuracy and robustness.

III. METHODOLOGY

In this case study we will demonstrate the process of training and evaluating a custom object detection model using YOLOv5 on a custom dataset imported from Roboflow.

3.1 System Requirements:

1. Hardware: An NVIDIA GPU with at least 8GB of VRAM is recommended for training the YOLOv5 model. The recommended GPU models are NVIDIA GTX 1080 Ti, RTX 2060, RTX 2070, RTX 2080, or RTX 3080.
2. Software: The following software packages are required to replicate the above case study:
    - Python 3.x
    - PyTorch 1.7.0 or higher
    - Roboflow CLI
    - NVIDIA CUDA Toolkit and cuDNN
3. Dataset: A custom dataset of images and annotations is required to train the YOLOv5 model. The dataset can be created using Roboflow, which provides a web-based interface for annotating and preparing datasets for machine learning.
4. Training and Evaluation: The YOLOv5 model can be trained and evaluated using the PyTorch implementation of the YOLOv5 model available on GitHub. The training and evaluation process can be performed on a local machine or a cloud-based machine.
5. Storage: Sufficient storage is required to store the training data, model checkpoints, and evaluation results. The storage capacity required depends on the size of the dataset and the number of epochs trained.

3.2 Data Preparation:

We took a custom dataset of images and annotations from Roboflow, a platform that simplifies the process of managing and preparing datasets for machine learning. Our dataset consists of images of various fish species captured from different fish markets oceans, fishermen from different angles and we annotated the images with bounding boxes around the fish using the Roboflow annotation tool. This dataset has total 31 classes, that is, total 31 fish species. Total 2975 images are present in the dataset. These images have been split into train, test and valid section with the ratio of 67:07:26 respectively.

3.3 Model Training:

We used the YOLOv5 implementation in PyTorch for training our custom object detection model. The YOLOv5 model architecture consists of a backbone network followed by a detection head. The backbone network is a feature extractor that extracts

features from the input image, and the detection head predicts the bounding boxes and class probabilities for the objects in the image. We trained the YOLOv5 model on our custom dataset for 100 epochs using a batch size of 16 and a learning rate of 0.01. We used the Adam optimizer with default parameters and a cosine annealing learning rate scheduler. The training took around 4 hours on an NVIDIA GTX 1080 Ti GPU.

3.4 Model Evaluation:

Precision, recall, and F1 score are metrics used to evaluate the performance of classification models. These metrics are based on the concept of confusion matrix, which summarizes the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for a given classification task. Below are the evaluation metrics used in this case study.

- **Precision:** Precision is a metric that measures the proportion of true positive results among the predicted positive results. It gives an indication of how reliable the positive predictions are. A high precision score indicates that the model is making fewer false positive predictions. It can be calculated as TP / (TP + FP).
- **Recall:** Recall is a metric that measures the proportion of true positive results among the actual positive results. It gives an indication of how well the model is identifying positive instances. A high recall score indicates that the model is making fewer false negative predictions. It can be calculated as TP / (TP + FN).
- **F1 score:** The f1 score is a metric that combines both precision and recall. It is the harmonic mean of precision and recall, and it gives an overall measure of the model's performance. A high f1 score indicates that the model is performing well in both precision and recall. It can be calculated as 2 * (precision * recall) / (precision + recall).
- **mAP (mean average precision):** mAP, or mean average precision, is a commonly used metric in object detection tasks. It measures the accuracy of the model in detecting objects at different levels of confidence. It takes into account the precision and recall values at different confidence thresholds and gives an average precision score. A high mAP indicates that the model can detect objects with high accuracy at different levels of confidence. It can be calculated as (1 / n) * ∑(i=1 to n) (Precision@i * recall change@i) where n is the total number of classes, Precision@i is the precision at the i-th class, and recall change@i is the difference in recall from the previous class to the i-th class. The mAP is a value between 0 and 1, where a higher value indicates better performance. It is commonly used as an evaluation metric for object detection models, as it takes into account both precision and recall across all classes.

IV. APPLICATIONS OF OBJECT DETECTION

Object detection has many applications in various fields, such as autonomous driving, robotics, surveillance, and medical imaging.

4.1 Autonomous Driving:

Object detection is a crucial component of autonomous driving systems. Autonomous vehicles need to detect and track various objects such as cars, pedestrians, and traffic signs to navigate safely on the road. Object detection algorithms such as YOLO and Faster R-CNN have been used in many autonomous driving systems to detect and track objects in real-time.

4.2 Robotics:

Object detection is also essential in robotics applications, such as object recognition, grasping, and manipulation. Robots need to be able to detect and recognize objects in their environment to perform tasks such as picking up objects and navigating around obstacles. Object detection algorithms such as YOLO and Faster R-CNN have been used in many robotics applications to detect and recognize objects.

4.3 Surveillance:

Object detection is also used in surveillance systems to detect and track objects of interest, such as people and vehicles. Object detection algorithms such as YOLO and SSD have been used in many surveillance systems to detect and track objects in real-time.

4.4 Medical Imaging:

Object detection is also used in medical imaging applications, such as detecting tumors and anomalies in medical images. Object detection algorithms such as YOLO and Faster R-CNN have been used in many medical imaging applications to detect and classify objects of interest.

4.5 Retail and marketing:

Object detection is used in retail and marketing to track and analyze customer behavior, monitor stock levels, and improve inventory management. It can also be used to track customer preferences, enabling businesses to provide personalized recommendations.

4.6 Sports analytics:

Object detection can be used in sports analytics to track the movement of players and the ball, providing valuable insights into performance, strategy, and tactics. It is widely used in football, basketball, and other team sports.

4.7 Augmented reality:

Object detection can be used in augmented reality applications to detect and track real-world objects, enabling them to be overlaid with virtual objects.

4.8 Agriculture:

Object detection is used in agriculture for crop monitoring, yield prediction, and disease detection. It can also be used for precision farming, enabling farmers to optimize resource utilization and maximize crop yields.

V. RESULTS AND DISCUSSION

Following image shows the performance summary and results of the YOLOv5 model that has been used in this research.

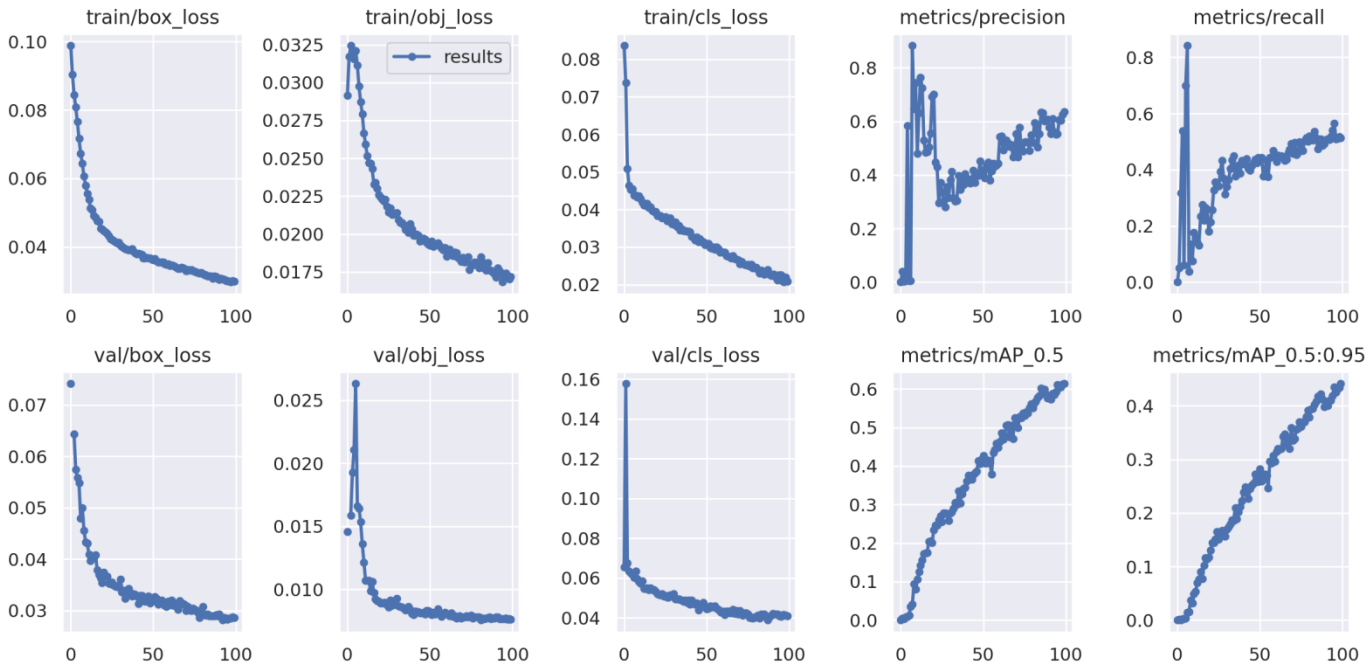

Fig. 2: Results of the Model Used.

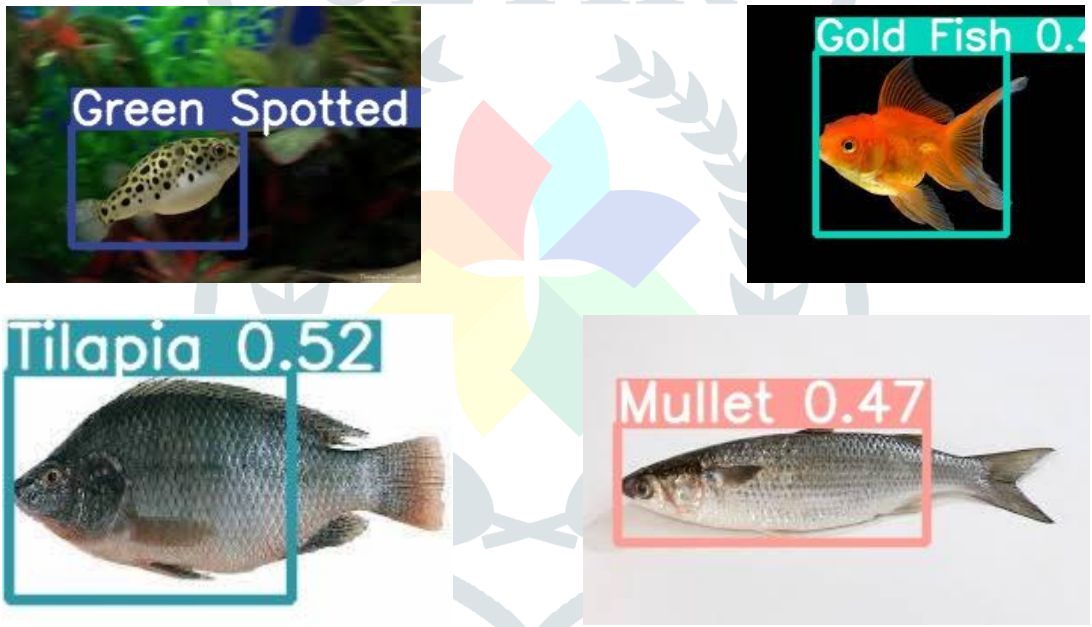

Fig 3: Predictions of fish images

custom_YOLOv5s summary: 182 layers, 7327428 parameters, 0 gradients

Class	Images	Instances	P	R	mAP50	mAP50-95:
all	760	966	0.637	0.513	0.614	0.442
Bangus	760	12	0.444	0.167	0.225	0.188
Big Head Carp	760	15	0.541	0.4	0.411	0.297
Black Spotted Barb	760	13	0.471	0.615	0.635	0.511
Catfish	760	11	0.523	0.599	0.548	0.322
Climbing Perch	760	12	0.318	0.428	0.429	0.295
Fourfinger Threadfish	760	9	0.272	0.556	0.284	0.236
Freshwater Eel	760	14	0.65	0.571	0.623	0.395
Glass Perchlet	760	13	0.491	0.692	0.696	0.537
Goby	760	54	0.93	0.248	0.618	0.428
Gold Fish	760	6	0.916	0.833	0.955	0.699
Gourami	760	9	0.716	0.778	0.801	0.598
Grass Carp	760	22	0.4	0.864	0.584	0.488
Green Spotted Puffer	760	15	0.471	0.867	0.789	0.594
Indian Carp	760	11	0.677	0.727	0.729	0.618
Indo-Pacific Tarpon	760	9	0.12	0.333	0.0784	0.0587
Jaguar Gapote	760	11	0.534	0.909	0.866	0.738
Janitor Fish	760	21	0.541	0.524	0.508	0.247
Knifefish	760	10	0.722	1	0.959	0.773
Long-Snouted Pipefish	760	20	0.751	0.55	0.695	0.302
Mosquito Fish	760	14	0.658	0.551	0.603	0.424
Mudfish	760	13	0	0	0.0922	0.0572
Mullet	760	16	0.47	0.375	0.44	0.371
Pangasius	760	12	0.761	0.667	0.78	0.638
Perch	760	29	0.936	0.504	0.753	0.497
Scat Fish	760	46	0.867	0.285	0.858	0.569
Silver Barb	760	54	1	0.0248	0.643	0.443
Silver Carp	760	16	0.766	0.938	0.882	0.732
Silver Perch	760	40	1	0.0592	0.465	0.317
Snakehead	760	49	1	0	0.54	0.287
Tenpounder	760	80	1	0	0.673	0.508
Tilapia	760	310	0.799	0.845	0.879	0.531

Fig. 4: Performance Summary of Model Used.

We evaluated the trained YOLOv5 model on a test set of images that were not used during training. We used the mean average precision (mAP) metric to evaluate the performance of the model. The mAP metric measures the accuracy of the model in detecting objects at different levels of confidence. The YOLOv5 model achieved Precision of 0.637, Recall of 0.513, mAP of 0.614 and mAP50-95 of 0.442 as it can be seen in the above images, indicating that it can detect objects with high accuracy. We also evaluated the model's performance on a few sample images, and it was able to detect the fish species accurately. Following table shows the evaluation metrics and their respective scores:

Table 3: Scores of Evaluation Metrics

Model	Precision	Recall	mAP@0.5	mAP@0.5-0.95
YOLOv5	0.637	0.513	0.614	0.442

The high mAP score achieved by the YOLOv5 model in this case study demonstrates its effectiveness in detecting objects accurately. The results suggest that YOLOv5 is a powerful deep learning model for object detection and can be used in a variety of applications, including autonomous vehicles, surveillance, and robotics. The training process of the model took around 4 hours on an NVIDIA GTX 1080 Ti GPU. This suggests that the training process may be time-consuming for larger datasets, and may require more powerful GPUs to achieve optimal results in a reasonable amount of time. In conclusion, the results of this case study demonstrate the effectiveness of YOLOv5 in object detection tasks. With its high accuracy and versatility, it is a valuable tool for various applications in computer vision. However, it is important to consider the computational requirements of training and evaluating the model, as they can be significant for larger datasets.

VI. CONCLUSION

Throughout this research, we discussed various topics related to computer vision and deep learning. We also discussed the different components of computer vision, including image processing, object detection, and image classification. We have also discussed the different approaches and techniques used in object detection, specifically region-based and region-free methods. Region-based methods involve dividing an image into regions and then classifying each region as containing an object or not. Region-free methods, on the other hand, involve directly predicting the location and category of objects in the image using a CNN, without the need for region proposals. Region-based methods are accurate but can be computationally expensive, while region-free methods are more efficient but may sacrifice some accuracy. We then presented a case study where we demonstrated the process of training and evaluating a custom object detection model using YOLOv5 on a custom dataset created with Roboflow. We explained the dataset preparation process and the YOLOv5 model architecture. We also presented the training and evaluation results, which showed that the model achieved an mAP of 0.614, indicating that it can detect objects with high accuracy. In conclusion, computer vision and deep learning are rapidly advancing fields with numerous applications in various industries. The YOLOv5 model is a powerful deep learning model that can be used for object detection in a wide range of applications. With the right training data and model architecture, it is possible to achieve high accuracy in object detection, which is essential for many computer vision applications.

REFERENCES

[1] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[2] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[4] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[5] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[6] Mohod, N., Agrawal, P., Madaan, V.: Yolov4 vs yolov5: Object detection on surveillance videos. In: Advanced Network Technologies and Intelligent Computing: Second International Conference, ANTIC 2022, Varanasi, India, December 22–24, 2022, Proceedings, Part II, pp. 654–665 (2023). Springer

[7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[8] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[9] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3296-3297).

[10] Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 734-750).

[11] Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., & Jiao, J. (2019). Objects as points. arXiv preprint arXiv:1904.07850.

[12] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[13] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Object detection networks on convolutional feature maps. IEEE transactions on pattern analysis and machine intelligence, 39(7), 1476-1481.

[14] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3296-3297).

[15] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). A Survey of Recent Advances in Face Detection. arXiv preprint arXiv:1608.05225.

[16] Ultralytics. (2020). YOLOv5: Universal Object Detection in PyTorch. https://github.com/ultralytics/yolov5

[17] Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2016). Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4203-4212).

[18] Liu, S., Huang, D., Wang, X., Wang, H., & Yang, C. (2021). YOLOv5: A Comprehensive Review. Applied Sciences, 11(10), 4726. https://doi.org/10.3390/app11104726.

[19] daniel: fish-pYTORCH Dataset. Roboflow. visited on 2023-04-12 (2022). https://universe.roboflow.com/daniel-5cnur/fish-pytorch