ABSTRACT

Identifying occluded objects is a challenging task in computer vision because real-world settings frequently involve objects that are partially hidden by different visual aspects and are often found in partial blind spots of car mirrors. This work explores the complexities of occluded object recognition using two state-of-the-art computer vision models: You Only Look Once (YOLO) and Faster RCNN. The main objective is to evaluate and contrast these models' abilities, particularly with regard to handling partial blind spot-related circumstances.

Although the two-stage Faster R-CNN framework and the one-stage YOLO algorithm have shown promising results in general object detection tasks, their ability to handle occlusion is yet unknown. This research attempts to close this gap by assessing the models' MAP (mean average precision) in identifying and locating obstacles in a variety of challenging settings.

Our proposed approach, termed YOLOv5 algorithm, incorporates the Bidirectional Feature Pyramid Network (BiFPN) into the YOLOv5 framework to tackle the hurdles associated with multi-scale and multi-level feature fusion, thereby improving detection capabilities across objects of different sizes. Moreover, we integrate the Convolutional Block Attention Module (CBAM) into the YOLOv5 architecture to augment its feature.

The findings shed light on the advantages and disadvantages of YOLO and Faster R-CNN in situations with hidden area objects. Furthermore, the research provides useful suggestions for choosing a suitable model according to the particular needs of fields where occlusion is a common problem, like robotics, autonomous driving, and surveillance.

Keywords : Neural Networks, Computer Vision, YOLO, Faster Region-Based Convolutional Neural Network (R - CNN), Accuracy, Mean Average Precision (mAP), Frames Per Second (FPS), Real-Time video

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OBJECT CLASSIFICATION :

In recent years, the field of computer vision has undergone unprecedented growth, driven by the convergence of advanced algorithms, increasing computational capabilities, and the availability of vast datasets. One of the central challenges within this domain is the accurate and efficient classification of objects within visual data, a task that forms the cornerstone of many real-world applications such as autonomous systems, surveillance, and human-computer interaction.

Fig. 1.1: Object Classification

Object classification, a fundamental aspect of computer vision, involves the identification and categorization of objects within images or video streams. This process is pivotal for systems to comprehend and interpret their visual surroundings, enabling them to make informed decisions in response to dynamic and complex environments. As technology continues to advance, the demand for robust and reliable object classification methods becomes increasingly paramount.

1.2 OBJECT DETECTION IN COMPUTER VISION :

Object detection is a fundamental task in computer vision that involves identifying and locating objects within an image or video. Over the years, the field of object detection has undergone significant advancements, with deep learning techniques playing a pivotal role in revolutionizing its capabilities. This seemingly basic task has numerous applications across diverse fields, including:

1. Autonomous vehicles: Self-driving cars rely on object detection for identifying pedestrians, vehicles, and traffic signs, ensuring safe navigation.
2. Medical diagnosis: Computer-aided diagnosis systems utilize object detection for identifying abnormalities in medical images, assisting doctors in early disease detection.
3. Surveillance and security: Object detection algorithms are employed in security cameras and surveillance systems to automatically detect suspicious activities or unauthorized individuals.
4. Robotics: Robots utilize object detection for object manipulation, obstacle avoidance, and path planning.

1.3 UNDERSTANDING CLASSIFICATION vs LOCALIZATION vs DETECTION

Table 1.1: Classification vs Localization vs Detection

Aspect	**Object Classification**	**Object Localization**	**Object Detection**
Definition	Assign a label or category to the entire image or a single object within the image.	Determine the precise location of a single object in an image, often represented by a bounding box.	Identify and locate multiple objects within an image, providing bounding boxes for each.

Aspect	Object Classification	Object Localization	Object Detection
Main Task	Identify the category or label of the entire image or a single object within the image.	Specify the coordinates (x, y) and size of a bounding box around a single object.	Detect and localize multiple objects, providing bounding boxes for each object found.
Output	A single label or category for the entire image or a detected object.	Bounding box coordinates for a single object.	Multiple bounding boxes with associated object labels.
Example	Identifying that an image contains a cat or a dog.	Locating a specific car within an image.	Detecting and locating multiple cars, pedestrians, and traffic signs in an image.
Use Case	When the focus is on recognizing the content or category of an image without specifying its location.	When the goal is to precisely locate a specific object within an image.	When the goal is to detect and locate multiple objects, useful in applications like autonomous driving and object recognition.
Complexity	Relatively simpler as it deals with assigning categories without precise localization.	Generally simpler as it focuses on a single object.	More complex as it involves handling multiple objects and their interactions.

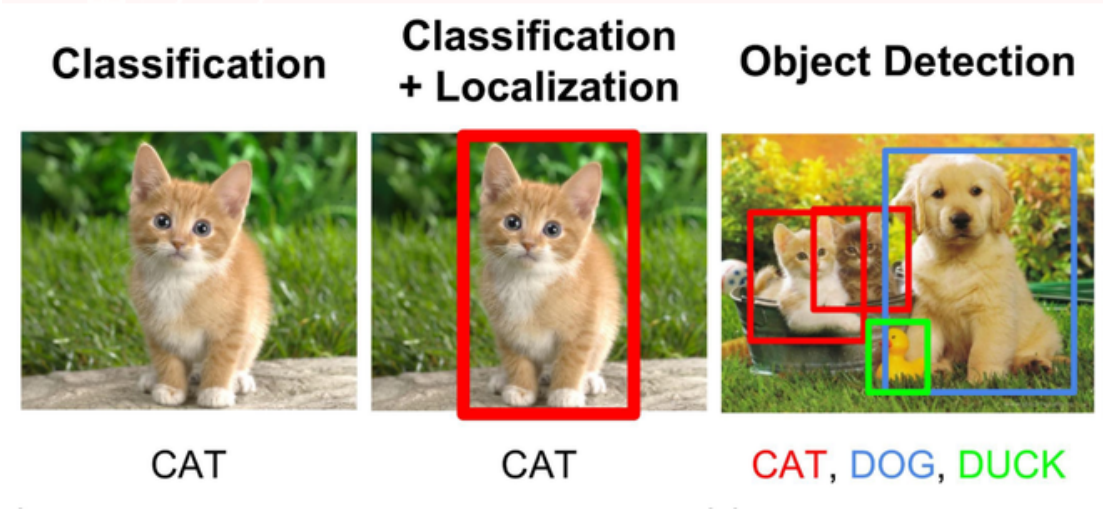

Fig. 1.2: Object Classification vs Object Localization vs Object Detection

1.4 ADVANCEMENTS WITH DEEP LEARNING TECHNIQUES :

The advent of deep learning has brought about a paradigm shift in the way computers perceive and understand visual information. Deep learning models, particularly convolutional neural networks (CNNs), have demonstrated unparalleled success in tasks related to image analysis, including object detection. The intricate hierarchical feature learning enabled by deep architectures allows these models to automatically extract relevant patterns and representations from raw visual data.

Two notable deep learning models that have propelled the field of object detection to new heights are You Only Look Once (YOLO) and Faster R-CNN (Region-based Convolutional Neural Network).

1.4.1 You Only Look Once (YOLO) :

YOLO is a groundbreaking object detection framework that excels in real-time applications with its earliest iterations (YOLOv1) introduced in 2015. What sets YOLO apart is its ability to process an entire image in a single forward pass through the neural

network, providing predictions for object classes and bounding box coordinates simultaneously. This efficiency makes YOLO particularly well-suited for applications where speed is crucial, such as video analysis and real-time object tracking. After that, the YOLO algorithm was continuously improved. YOLOv2 optimised the low accuracy problem in v1, improving the precision and speed of multi-object detection; YOLOv3 chose to add multi-scale training and flexibly process input images, improving the accuracy of small object detection in v2; YOLOv4 solved the problem of GPU training; and YOLOv5 reduced the model size, making it feasible for deployment on mobile edge devices. In this paper, we use YOLOv5 contrasting it with enhanced YOLOv1 for object detection on an intelligent networked car host, so we will introduce it in detail using YOLOv5 as an example [27,28,29,30].

1.4.2. Faster R-CNN :

Faster R-CNN introduced a region-based approach to object detection, addressing some of the limitations of earlier models. By incorporating a region proposal network (RPN), Faster R-CNN can efficiently propose candidate object regions before performing detailed classification and bounding box regression. This two-stage process enhances accuracy and has become a cornerstone in the evolution of object detection algorithms.

These models represent just a fraction of the diverse landscape of deep learning approaches to object detection. The continuous refinement and development of such models have significantly improved the precision, speed, and versatility of object detection systems. As a result, industries and research communities are leveraging these

advancements to create innovative solutions and applications that harness the power of computer vision for a wide range of practical scenarios.

This paper explores the complexities of occluded object detection with the goal of improving our comprehension of how well YOLO and Faster R-CNN function in these kinds of situations. Although both models—YOLO using a one-shot detection procedure and Faster R-CNN using a two-shot approach—have demonstrated superior performance

In conventional object identification tasks, there is still an unanswered question of how well they handle occlusion. This disparity calls for a careful analysis and comparison.

In order to evaluate these models' occluded object identification abilities in detail, we carefully selected a particular dataset that highlights partial blockages of different kinds of objects. Faster R-CNN and YOLO underwent extensive testing on this dataset, and their models were fine-tuned to enable them to handle the challenges presented by occlusion.

Measures of assessment like F1 score, precision, and recall allow for a quantitative comparison of the models' performance at different degrees of occlusion.

Apart from clarifying the specific difficulties related to obscured object identification, this research endeavors to offer pragmatic recommendations for the selection of a suitable model customized to the unique requirements of applications in which occlusion poses a noteworthy concern. The expected results of this study are meant to make a valuable contribution to the current discussion on deep learning model optimization for practical applications, pushing computer vision research in the direction of more robust and dependable solutions.

1.5 MOTIVATION :

The reason for selecting this project is a deep fascination with the revolutionary possibilities of 21st-century autonomous driving technology. A paradigm change in

transportation, autonomous driving offers greater environmental sustainability, economy, and safety. There is a strong incentive to contribute to developments in Computer Vision and Machine Learning since this state-of-the-art technology mostly depends on these fields, especially in the area of object detection.

An essential part of autonomous navigation is object identification, which calls for exact algorithms to identify and locate things. Intriguing issues that pique intellectual curiosity

include the difficulties in managing several possible object placements, or "proposals," and the trade-offs between simplicity, precision, and speed.

New avenues for tackling these problems have been made possible by the development of advanced deep learning models like You Only Look Once (YOLO) and Region-Based Convolutional Neural Networks (R-CNN). This effort is motivated by the desire to investigate and comprehend these sophisticated models, their advantages and disadvantages in the context of actual autonomous driving situations.
Another layer of difficulty is added by the emphasis on obstructed object detection. Finding creative solutions is necessary when navigating through challenging driving situations where things could be partially disguised. Analyzing and contrasting different occluded object identification techniques with the same accessible sensors offers a rare chance to provide insights that could improve the dependability and safety of autonomous cars.

In essence, the motivation behind choosing this project lies in the aspiration to be at the forefront of technological advancements, actively contributing to the evolution of autonomous driving technology. The project aligns with a broader vision of creating more robust and efficient autonomous systems that have the potential to revolutionize the future of transportation.

1.6 PROBLEM STATEMENT :

The creation of a sophisticated object detection system presents a substantial challenge in advancing the capabilities of autonomous vehicles. This task surpasses the conventional identification and categorization of objects in the vehicle's immediate surroundings. The primary objective is to design a comprehensive system seamlessly integrating diverse sensor inputs, including cameras, radar, LiDAR, and state-of-the-art perception algorithms. This holistic approach is essential for achieving a comprehensive and real-time understanding of the vehicle's environment.

Several complexities highlight the multidimensional nature of this challenge. Ensuring the system can consistently detect and track objects, even when they are partially or entirely obscured by surrounding elements like buildings, other objects, or adverse weather conditions, is of utmost importance. Additionally, given the potential unpredictability of pedestrians, cyclists, and other road users, the system must expand its capabilities to predictive functionality, going beyond mere object identification to anticipate future movements and intentions.

The intricacy is further heightened by anomaly detection, necessitating the system to identify uncommon incidents or unexpected objects in the environment, such as construction zones, falling objects, or atypical traffic scenarios. The system's resilience is critical, as it must perform effectively in challenging situations, including poorly lit areas, heavy rain, snow, fog, and varying levels of environmental complexity.

The proposed solution must also showcase scalability and efficiency, operating effectively on hardware with limited computational power while managing the intricacies of complex urban environments. The system should not only identify objects but also comprehend the surrounding environment at a semantic level, covering the recognition of lane boundaries, drivable zones, and traffic signs.

Finally, adherence to laws and best practices is imperative to ensure the secure handling of sensitive data from perception sensors, given the significance of data privacy and security. Effectively addressing these intricate challenges associated with autonomous vehicle object recognition is crucial for advancing safety, reliability, and promoting the widespread adoption of self-driving technology, ultimately reshaping the landscape of transportation.

CHAPTER 2

LITERATURE REVIEW

2.1 RELATED WORKS FOR YOU ONLY LOOK ONCE (YOLO)

In recent literature, researchers have made significant advancements in the field of object detection by proposing innovative models based on both RCNN and YOLO architectures. One noteworthy contribution comes from Wang et al., who introduced RSI-YOLO as an enhanced version of YOLOv5 tailored for efficient object detection in remote sensing images [7]. In their work, they incorporated sophisticated channel attention and spatial attention mechanisms to improve feature fusion. The integration of these attention mechanisms led to superior detection performance compared to traditional object detection algorithms.

Moreover, Sridhar conducted a study that leveraged the YOLO real-time object identification approach to train a machine learning model. YOLO, renowned for its simplicity and efficiency, stands out in predicting bounding boxes and class probabilities for the entire image, resulting in faster detection. Sridhar's work highlights the effectiveness of YOLO in real-time applications and underscores its advantages in terms of speed and simplicity.

In a different domain, Zhou et al. proposed YOLO-CIR, an object detection algorithm specifically designed for infrared images using transfer learning techniques [8]. YOLO-CIR surpassed the performance of both YOLOv5 and Faster R-CNN, particularly in terms of mean average precision (map50). This achievement emphasizes the adaptability of YOLO-based models to different domains, showcasing its versatility in handling specialized data types like infrared imagery.

These studies collectively contribute valuable insights into the versatility and effectiveness of RCNN and YOLO architectures for object detection across diverse domains, ranging from remote sensing to real-time applications and infrared imaging. The incorporation of attention mechanisms, transfer learning, and domain-specific optimizations demonstrates the continuous evolution and refinement of object detection methodologies.

Table 2.1: Literature Review of You Only Look Once (YOLO)

Paper	Conclusions	Insights	Results
A Review on Real Time Object Detection Using Deep Learning (2023) (Priyanka Padmane)	- Implementation of convolutional networks for object detection - YOLO model used for quick implementation of object detection system	The provided paper discusses the implementation of an object detection system using YOLO (You Only Look Once) deep learning model. There is no mention of the RCNN (Region-based Convolutional Neural Network) model in the paper.	- YOLO object detection model achieved an accuracy of 81.2%. - The model has a mean Average Precision (mAP) of 0.5.
Developing a YOLO based Object Detection Application using OpenCV (2023)	- YOLO is a highly efficient object detection algorithm for real-time image processing. - YOLO predicts bounding boxes and class probabilities.	The paper discusses the use of YOLO (You Only Look Once) for object detection, but does not mention the use of RCNN (Region-based Convolutional Neural Network).	- The paper employs the YOLO real-time object identification approach for training the machine learning model.

Paper	Conclusions	Insights	Results
RSI-YOLO : Object Detection Method for Remote Sensing Images Based on Improved YOLO (2023) (Hao Wang)	- RSI-YOLO outperforms the original YOLO in detection performance. - RSI-YOLO algorithm is superior to other algorithms.	The provided paper is about a remote sensing image detection approach called RSI-YOLO, which is based on the YOLOv5 algorithm. There is no mention of using the R CNN algorithm in the paper.	- RSI-YOLO algorithm demonstrated superior detection performance compared to other algorithms.
YOLO-CIR : The network based on YOLO and ConvNeXt for infrared object detection (2023) (Jinjie Zhou, Baohui Zhang)	- YOLO-CIR outperforms YOLOv5 and Faster R-CNN in map50. - YOLO-CIR has advantages in parameters and FLOPs.	The provided paper proposes a new object detection algorithm called YOLO-CIR, which is based on YOLO and ConvNeXt. It does not mention the use of RCNN in the model.	- YOLO-CIR outperforms YOLOv5 by 3% and Faster R-CNN by 5.6% in map50. - YOLO-CIR has significant advantages in parameters.
SSDA-YOLO : Semi-supervise d domain adaptive YOLO for cross-domain object detection (2023)	- Proposed SSDA-YOLO method improves cross-domain detection performance.	The paper proposes a semi-supervised domain adaptive YOLO (SSDA-YOLO) method for object detection, integrating YOLOv5 with domain adaptation techniques.	- Considerable improvements in cross-domain object detection tasks

2.2 RELATED WORKS FOR FASTER R-CNN

The Faster R-CNN (Region-based Convolutional Neural Network) model has emerged as a powerful and widely utilized framework for object detection in diverse domains. Researchers have actively contributed to the continuous improvement of the Faster R-CNN model, proposing modifications and enhancements to address specific challenges and optimize its performance in various applications.

One notable advancement comes from Chongben Tao et al., who introduced the Fast Guided Anchored Stereo RCNN (FGAS RCNN) designed specifically for 3D object detection in autonomous driving scenarios [12]. The key innovation of FGAS RCNN lies in its ability to reduce computational costs while maintaining a high regression rate. This is crucial in autonomous driving, where real-time processing is essential. By integrating guided anchoring techniques, the model achieves efficient 3D object detection, demonstrating its suitability for resource-constrained environments.

In a separate contribution, Wenshun Sheng et al. proposed an enhanced Faster R-CNN variant known as CF-RCNN. CF-RCNN integrates two critical modules, the Convolutional Block Attention Module (CBAM) and the Feature Pyramid Network (FPN). These additions aim to improve the detection and recognition accuracy, particularly for challenging scenarios involving small-sized, occluded, or truncated objects in complex scenes. The integration of attention mechanisms and feature pyramid networks enhances the model's ability to handle objects with varying scales and levels of occlusion.

Furthermore, Hao Wang and Nanfeng Xiao focused on optimizing Faster R-CNN for underwater object detection [14]. They replaced the original VGG16 structure with the Res2Net101 network and introduced optimization techniques such as Online Hard Example Mining (OHEM), Generalized Intersection over Union (GIOU), and Soft Non-Maximum Suppression (Soft-NMS). These modifications were tailored to the unique challenges of underwater environments, resulting in improved performance.

Collectively, these papers showcase the adaptability and effectiveness of the Faster R-CNN model in diverse applications. The proposed enhancements address specific challenges in domains such as autonomous driving, complex scenes, and underwater environments, demonstrating the model's versatility and potential for advancing object detection capabilities in various fields. The continuous refinement of Faster R-CNN through innovative modifications underscores its significance as a foundational framework in the field of computer vision.

Table 2.2 : Literature Review of Faster R CNN

Paper	Conclusions	Insights	Results
Improved Object Detection Algorithm Based on Faster RCNN (2022) (Hua Wang, Shifa Jiang)	- Improved algorithm increases mAP by 5.4% - It addresses regression accuracy and positioning problems	The paper proposes an object detection approach using the Faster R-CNN model. It utilizes a custom dataset for training and pretrained datasets from the model zoo to detect objects in online proctoring exams.	- Improved algorithm increases mAP by 5.4% - Tested on Pascal VOC 2012 dataset
Object Detection Using Adaptive Block Partition and R CNN Algorithm (2023)	- Hardware acceleration of object detection algorithms achieved effective resource utilization.	The provided paper does not mention the Faster R-CNN algorithm for object detection. It focuses on hardware acceleration of different filter algorithms for real-time object detection using Xilinx Zynq-7000 SoC.	- Effective resource utilization: 45.6% logic cells, 51% LUTs, 29.47% Flip Flops, 15% Block RAMs

Paper	Conclusions	Insights	Results
Object Detection in Online Proctoring Through Two Camera Using Faster-RCNN (2023) (Vivien Arief Wardhany)	- Precision value of 0.884615385 for detecting online exam fraud using a side camera. - Recall value of 0.821428571 for detecting online exam fraud.	The paper proposes an object detection approach using the Faster R-CNN model. It utilizes a custom dataset for training and pretrained datasets from the model zoo to detect objects in online proctoring exams.	- The average bbox-AP of the training results is 59.169. - The accuracy of fraud detection is 0.884615385 and the recall is 0.821428571.
Faster RCNN Target Detection Algorithm Integrating CBAM and FPN (2023) (Wenshun Sheng)	- CF-RCNN improves detection and recognition accuracy for small-sized, occluded, or truncated objects.	The provided paper proposes an improved Faster RCNN algorithm called CF-RCNN that integrates CBAM and FPN to improve the detection and recognition accuracy of small-sized, occluded, or truncated objects in complex scenes.	- Mean average precision improved to 76.2% - 13.9 percentage points higher than other algorithms
Traffic Sign Detection Using SSD Mobilenet & Faster RCNN (2023) (Shaikh Asif)	- Deep learning-based techniques can increase the safety and effectiveness of transportation networks.	The paper mentions the use of Faster RCNN as one of the state-of-the-art object detection models utilized for traffic sign detection.	- The average bbox-AP of the training results is 59.169. - The accuracy of fraud detection is 0.884615385.

CHAPTER 3

METHODOLOGY

3.1 DATASET DESCRIPTION

The BDD100K dataset stands as a pivotal resource within the realm of autonomous vehicle research, distinguished by its comprehensive and diverse characteristics that render it paramount for advancing machine learning models in the context of urban driving scenarios. As an extensive and high-quality corpus of annotated data, BDD100K is meticulously curated to encapsulate a multitude of challenges inherent to autonomous driving, ranging from complex traffic scenarios and diverse environmental conditions to nuanced interactions with pedestrians and other road users. Here are some key aspects to consider:

1) **Scale and Diversity** : The dataset comprises over 100,000 high-resolution images, each meticulously labeled with semantic segmentation, instance segmentation, and object detection annotations. This rich annotation schema is pivotal for training and evaluating sophisticated computer vision algorithms, enabling a nuanced understanding of the urban landscape. Notably, BDD100K exhibits a level of granularity and complexity that surpasses many other datasets in the autonomous driving domain, ensuring a more robust evaluation of algorithmic performance under real-world conditions.

2) **Urban Driving Scenarios** : The dataset excels in capturing the intricacies of urban scenes, encompassing diverse weather conditions, lighting variations, and complex traffic scenarios. This diversity is crucial for the development of algorithms that exhibit resilience and adaptability, ensuring efficacy across a spectrum of operational contexts encountered by autonomous vehicles in actual urban environments.

3) **Annotations** : Furthermore, BDD100K is characterized by a meticulous validation of its annotations, ensuring a high degree of accuracy and reliability in the ground truth information. This meticulousness is paramount for fostering trust in the dataset's utility for training and evaluating autonomous vehicle algorithms, as inaccuracies in annotations can significantly compromise the efficacy of machine learning models.

4) **Object Detection and Segmentation** : Researchers can leverage the dataset for tasks such as object detection, where the goal is to identify and locate specific objects within an image, and semantic segmentation, which involves classifying and delineating each pixel in the image according to the object or scene category it belongs to.

5) **Training and Evaluation** : The BDD100K dataset serves as a valuable tool for training and evaluating the performance of computer vision models. Algorithms trained on this dataset can learn to recognize and understand the complex visual information present in urban environments, enabling advancements in technologies like autonomous vehicles.

In summary, the BDD100K dataset provides a rich and extensive resource for researchers, offering the necessary ingredients for developing sophisticated computer vision models capable of understanding and interpreting the complexities of urban driving scenes. The prioritization of BDD100K over alternative datasets is underscored by its capacity to address the limitations of earlier benchmarks. Its sheer scale and granularity enable the exploration of novel architectures and methodologies, pushing the boundaries of autonomous vehicle research. As the demand for robust, adaptive, and safe autonomous systems intensifies, BDD100K emerges as an indispensable asset, fostering advancements that resonate across the spectrum of machine perception, decision-making, and control within the autonomous driving domain.

Table 3.1 : Comparison of BDD100K with other datasets

Aspect	BDD100K	KITTI	Waymo Open	nuScenes
Scale	Over 100,000 images	Thousands of images and LiDAR point clouds	Substantial, real-world data from Waymo	Large-scale dataset with HD maps
Diversity	Diverse urban scenarios	Primarily suburban environments	Diverse urban and suburban scenarios	Urban, suburban, and highways
Annotation Granularity	Semantic & instance segmentation, object detection	3D object detection and depth information	Object detection, tracking, motion forecasting	3D bounding box annotations
Sensor Modality	Images (RGB), diverse weather conditions	Images (RGB), LiDAR	Images (RGB), LiDAR, Radar	Images (RGB), LiDAR, Radar, HD Maps
Temporal Coverage	Static images	Static images	Dynamic scenes with temporal continuity	Dynamic scenes with temporal continuity
Use Cases	Urban autonomous driving	Stereo, LiDAR-based perception	Advanced perception, prediction tasks	Perception, mapping, localization

3.2 YOU ONLY LOOK ONCE (YOLOv1)

YOLO, or You Only Look Once, is a pioneering object detection algorithm designed for real-time applications. Developed in 2015 by Redmon, Divvala, Girshick, and Farhadi, YOLO stands out for its speed and accuracy. The algorithm's fundamental principle is to process the entire image in a single pass through a neural network, unlike traditional methods that involve multiple passes or region proposals.

In essence, YOLO's objective is to efficiently identify and locate objects within an image. It achieves this by dividing the image into a grid, typically 7x7. Each grid cell is responsible for predicting bounding boxes and class probabilities, allowing YOLO to handle the entire image processing using a single convolutional neural network. The algorithm treats object detection as a regression problem, mapping image pixels to bounding box coordinates and class probabilities.

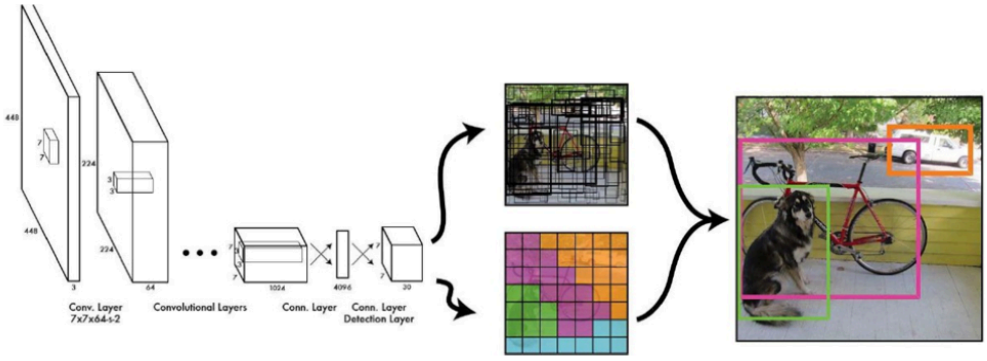

Figure 3.1 Work Flow of YOLO's algorithm

Once the input image has been divided into a T x T grid (usually T = 7), the process entails estimating N bounding boxes per grid cell (often N = 2 in the study). There are four spatial locations and a confidence level in each bounding box. Furthermore, P class probabilities are predicted for every grid cell; the class with the highest probability

determines the final categorization. A T x T x (N * 5 + P) tensor with these predictions is displayed.

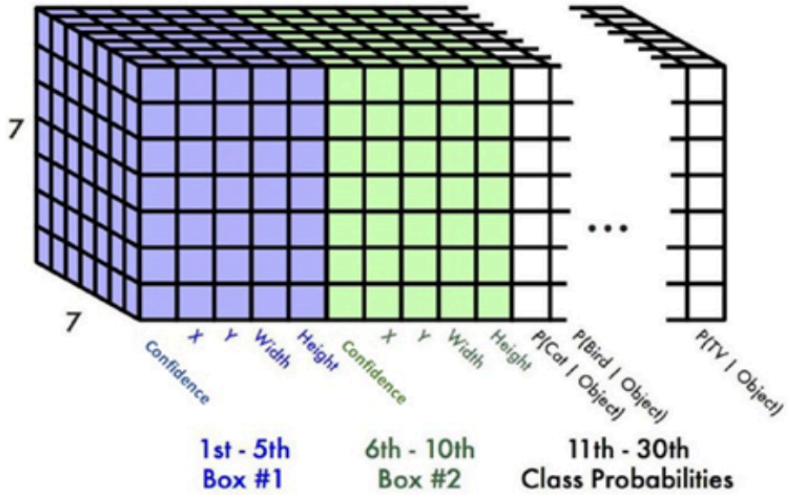

Figure 3.2 Tensor Output of YOLO

YOLO's architecture involves 24 convolutional layers, 4 pooling layers, and 2 fully connected layers. It utilizes 1x1 convolutions inspired by GoogLeNet's Inception Modules to reduce the volume of feature maps. The Leaky ReLu function serves as the activation function for all layers except the final one, and dropout is incorporated between the two fully connected layers to address potential overfitting concerns.

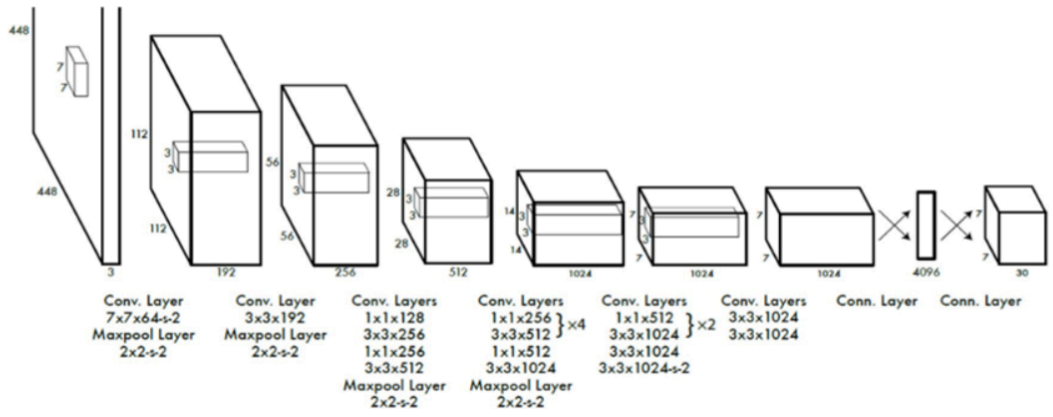

Figure 3.3 Visualised view of YOLO's Architecture

The algorithm's efficiency and effectiveness in real-time object detection have led to its evolution, with YOLOv8 being the latest version as of January 2023. YOLOv8, developed by Ultralytics, continues the tradition of speed and accuracy, and its open-source nature under a GPL license encourages further advancements in the field.

3.2.1 ENHANCEMENTS FOR OBJECT DETECTION IN YOLO

In the pursuit of enhancing the object detection capabilities within the existing YOLO (You Only Look Once) model, several strategic modifications are proposed for implementation. The primary objective is to accommodate scenarios with numerous objects within images more effectively. To achieve this, a pivotal adjustment involves the augmentation of the grid cell size, denoted as S, to a value of 14. Consequently, this alteration leads to an increase in the overall output parameters from the existing 1127 to a refined count of 4508.

A specific focus is directed towards the 23rd convolutional layer, where a strategic refinement is introduced. The typical stride value of 2 is adjusted to a stride of 1 in this layer. This adjustment is made with the explicit intention of preserving the spatial dimensions at 14 x 14, ensuring a meticulous retention of the grid cell size.

Table 3.2 : Comparison of Original YOLO with modified YOLO

Aspect	Original YOLO	Modified YOLO
Grid Cell Size (S)	7x7 grid	Increase to 14x14 grid
Output Parameters	1127 parameters	Increase to 4508 parameters
Convolutional Layer Stride	2	23rd layer: Stride 1

The overarching strategy involves transforming the existing YOLO architecture into a single end-to-end Convolutional Neural Network (CNN). This architectural shift signifies a departure from conventional approaches, treating object detection as a unified regression problem. The comprehensive image is systematically divided into an S x S grid of cells, and predictions are made for each grid cell. Specifically, for every grid cell, the model endeavors to predict B bounding boxes (where B equals 2 in this context), each characterized by four coordinate parameters and an associated confidence score. Additionally, the model predicts C class probabilities to ascertain the classification of objects within the respective grid cell. The pivotal concept here is the mapping of objects to the grid cell that encompasses the center of the object, thereby streamlining the detection process. This strategic realignment aims to enhance the model's efficacy in scenarios featuring a multitude of objects within complex images.

3.2.2 LOSS FUNCTION IN YOLO

Yolo utilizes a specially constructed loss function to control various output areas and how they affect the total loss. Certain parameters are included in this loss function, as seen in Figure 3.4. In addition to imposing penalties for differences in confidence scores, bounding box size, and center coordinates, it introduces a squared classification loss.

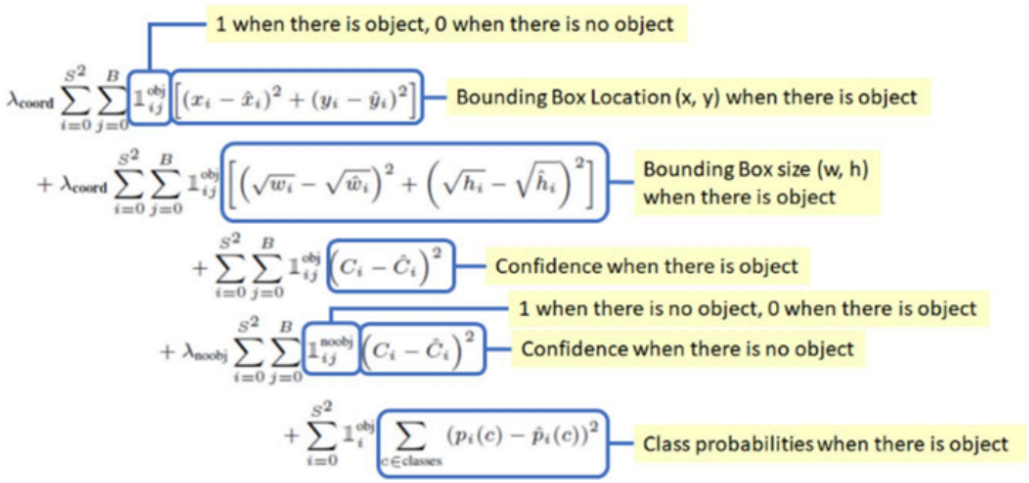

Figure 3.4 YOLO's loss function

3.3 FASTER R CNN

Faster R-CNN, introduced in 2015 by Girshick et al., is characterized by its single-stage, end-to-end training model. Leveraging the RPN for region proposal generation, this model expedites the detection process compared to traditional algorithms. The inclusion of the ROI Pooling layer facilitates the extraction of fixed-length feature vectors from each region proposal. The methodology of Faster R-CNN involves two stages: firstly, the identification of regions of interest, and secondly, the passage of these regions through a convolutional neural network. The resultant feature maps undergo classification using a support vector machine (SVM).

Faster R-CNN holds a prominent position among popular object detection architectures, alongside counterparts like YOLO (You Only Look Once) and SSD (Single Shot Detector), as it proficiently employs convolutional neural networks to enhance the accuracy and efficiency of object detection tasks.

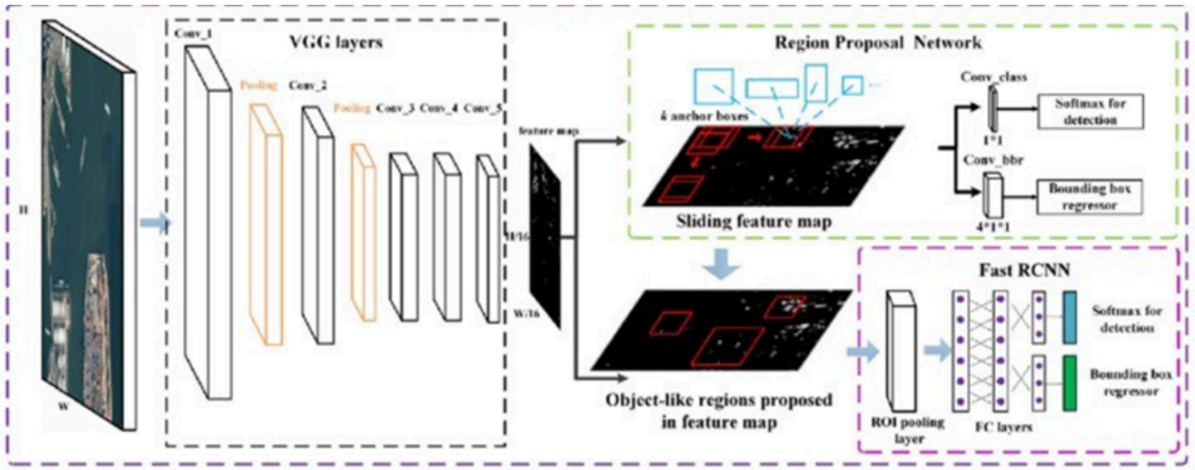

Figure 3.5 Visualised view of Faster RCNNs Architecture

Faster R-CNN, short for Faster Region-Convolutional Neural Network, stands as a sophisticated object detection model conceived by Ren et al. The architectural framework

comprises three integral components: the fast R-CNN module, the region proposal network (RPN), and the network backbone. The high-resolution feature maps are generated by the network backbone, typically a pretrained classification network such as ResNet50 on ImageNet, necessitating a 640 x 640 pixel input. These feature maps, coupled with region proposals stemming from the RPN, are fed into the fast R-CNN module. To maintain consistency in size (7x7) for each region proposal, the module employs ROI pooling.

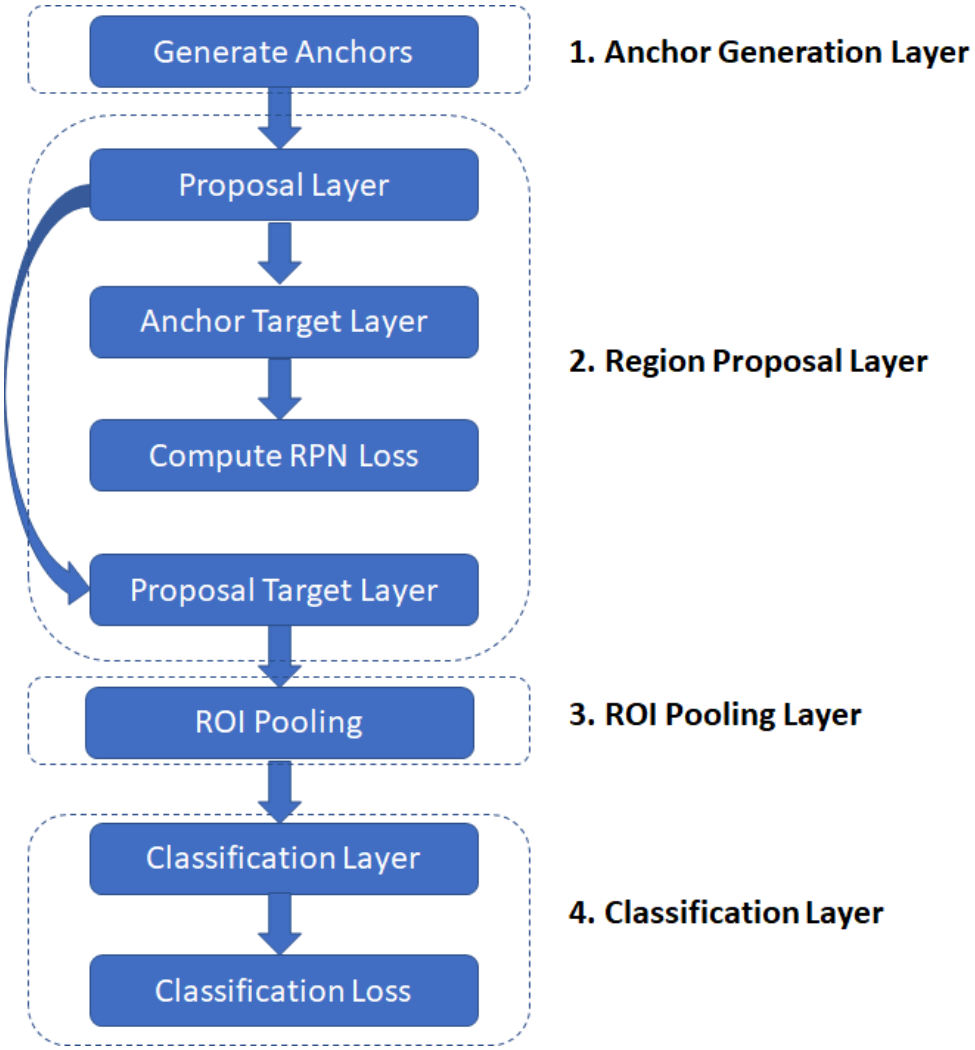

Figure 3.6 Key Components of Faster RCNNs Architecture

Following are the key components of Faster R-CNN :

3.3.1 Network Backbone :

The Network Backbone serves as the foundational element in the Faster R-CNN architecture, initiating the object detection process. Typically implemented with a pre-trained classification network such as ResNet50 on ImageNet, the backbone is responsible for processing input images and extracting high-resolution feature maps. These feature maps retain detailed information crucial for subsequent stages of object detection, providing a comprehensive understanding of the image content.

3.3.2 Region Proposal Network (RPN) :

The Region Proposal Network (RPN) is a pivotal component that operates on the high-resolution feature maps generated by the network backbone. It employs a convolutional layer with two branches: one predicting bounding box coordinates and the other predicting classification scores for proposed boxes. Using anchor boxes as reference points, the RPN efficiently generates region proposals, significantly narrowing down the potential locations of objects in the image. This step is crucial for optimizing the efficiency of subsequent processing stages.

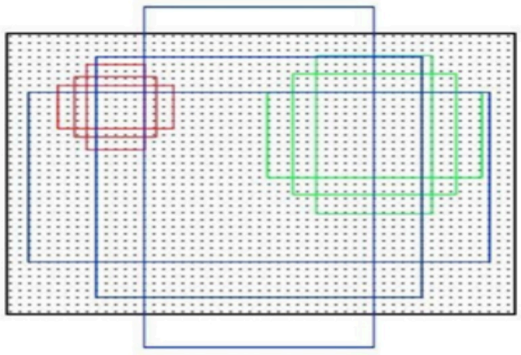

Figure 3.7 Anchor boxes ranging in sizes of 128, 256, and 512, coupled with aspect ratios of 0.5, 1, and 2.

3.3.3 Region of Interest (ROI) Pooling :

Following the generation of region proposals by the RPN, the Region of Interest (ROI) Pooling step standardizes their dimensions for further processing. By partitioning each proposed region into segments and determining the maximum value within each segment, ROI pooling creates a fixed-length feature vector for each region proposal. This process ensures uniformity in size, typically resulting in 7x7 dimensions. Standardized region proposals are essential for subsequent stages to maintain consistency in the model's understanding of object features.

3.3.4 Fast R-CNN Module :

The Fast R-CNN Module plays a critical role in refining the predictions made by the model. It takes the standardized feature vectors generated through ROI pooling and processes them through fully connected layers. These layers further adjust the bounding box predictions and produce softmax scores for each class, indicating the probability of an object belonging to a particular category. This module integrates object localization and classification into a unified framework, contributing to the model's ability to make precise predictions.

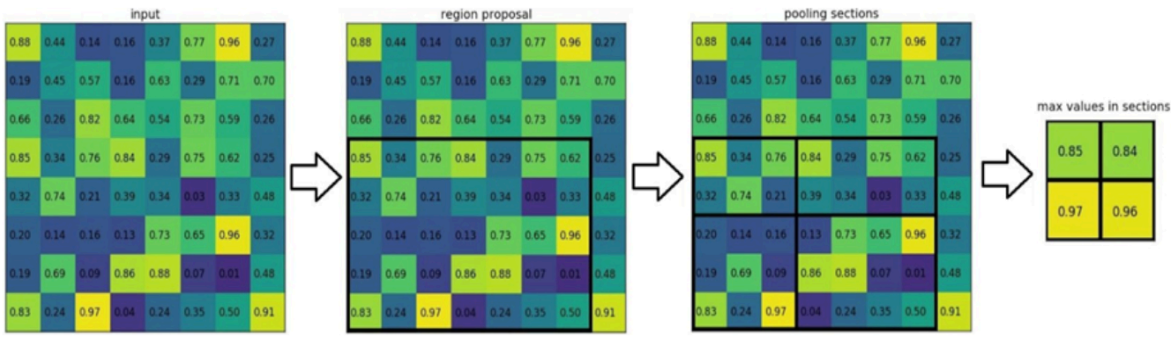

Figure 3.8 Diagram illustrating ROI pooling

3.3.5 Loss Functions :

To guide the training process effectively, Faster R-CNN employs specific loss functions tailored for classification and bounding box adjustments. For classification tasks, log loss, also known as cross-entropy loss, is applied to measure the disparity between predicted and ground truth class probabilities. Simultaneously, smooth L1 loss is utilized to quantify the differences in bounding box coordinates. The combination of these loss functions guides the backpropagation process during training, facilitating the model's learning and parameter adjustments.

The intricate interplay of these components within the Faster R-CNN architecture results in a sophisticated yet efficient object detection system, capable of providing accurate predictions in a variety of visual scenarios.

3.4 You only look once (YOLOv5)

The YOLOv5 network is mainly composed of four parts: input end, backbone main layer, neck feature fusion layer, and head output layer. The overall framework is shown in Figure 3.9.

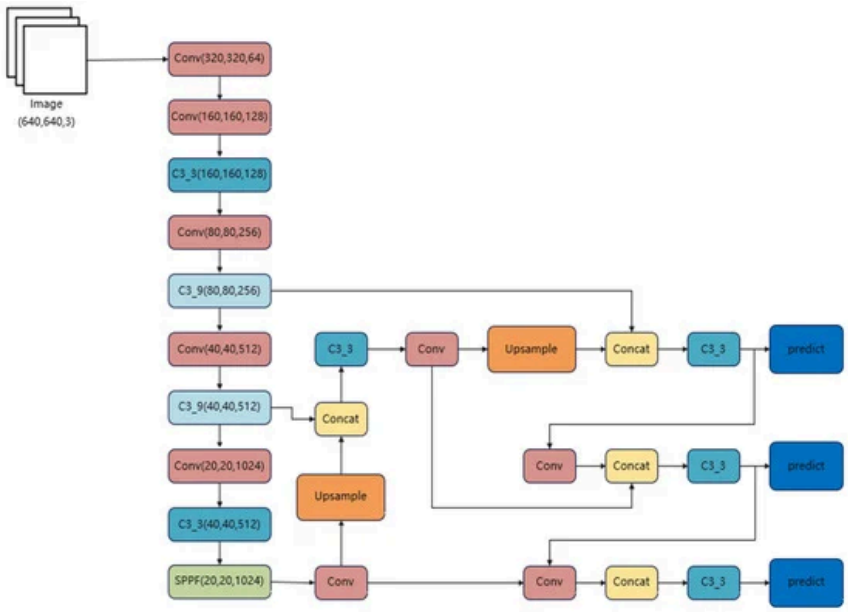

Figure 3.9 Structural diagram of the YOLOv5 network.

The backbone main layer is composed of three parts: a Conv module, a C3 module, and an SPPF module. It is used to extract image features and continually reduce the feature map size. YOLOv5 uses CSPDarknet as the backbone network, which extracts input image features through multiple CBS convolutional layers. After convolution, the C3 module is used for further feature extraction, and the SPPF module performs pooling operations to output feature layers of three scales: 80 × 80, 40 × 40, and 20 × 20.

The Conv module consists of Conv2d, a BatchNorm2d, and a SiLu activation function, mainly used for feature extraction and feature map organization. BatchNorm2d performs batch normalization on batch data, and the SiLu function enhances the non-linear fitting ability of the detection model, as shown in Figure 3.10.

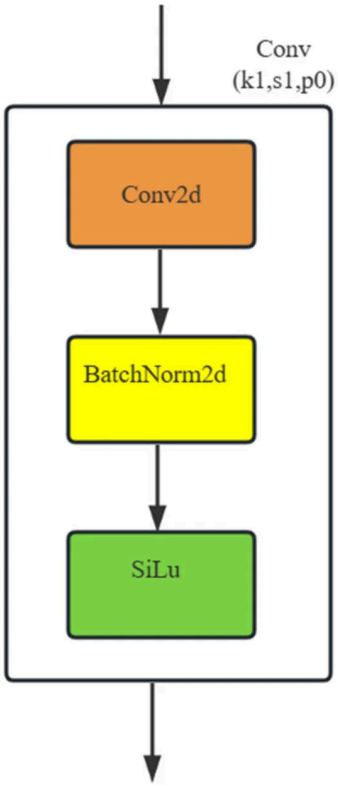

Figure 3.10. CBS convolutional layer.

The C3 module is a feature extraction module that stacks the image features extracted by the CBS convolutional layer to make the feature representation more sufficient. As shown in Figure 3.11, when the feature map enters C3, it will be processed in two ways. The Conv module in C3 reduces the dimension of the feature map to help the convolution kernel better understand the feature information, and then increases the dimension to

extract more complete feature information. Finally, a residual structure is used to extract features, combining the input and output to remove redundant gradient information.

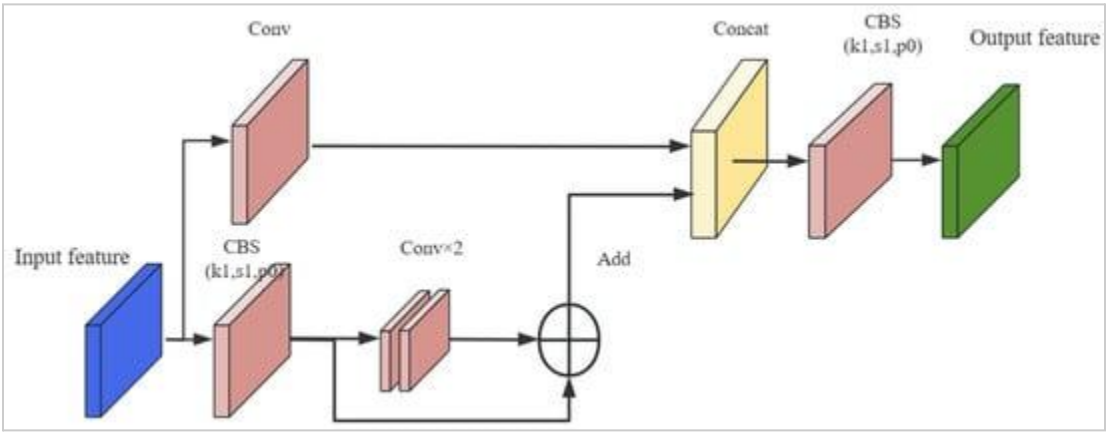

Figure 3.11. Schematic diagram of the C3 module.

SPPF is an improvement over the SPP spatial pyramid pooling, where feature maps of different scales are converted to the same scale through same pooling. As shown in Figure 3.12, it combines CBS convolutional layers and three serial 5 × 5 pooling layers to fuse multi-scale features. While further extracting features, it avoids the problem of incomplete expression of deep-level feature information. It extends the region in the input layer corresponding to the point on the feature map, thereby increasing the receptive field of the detection model.

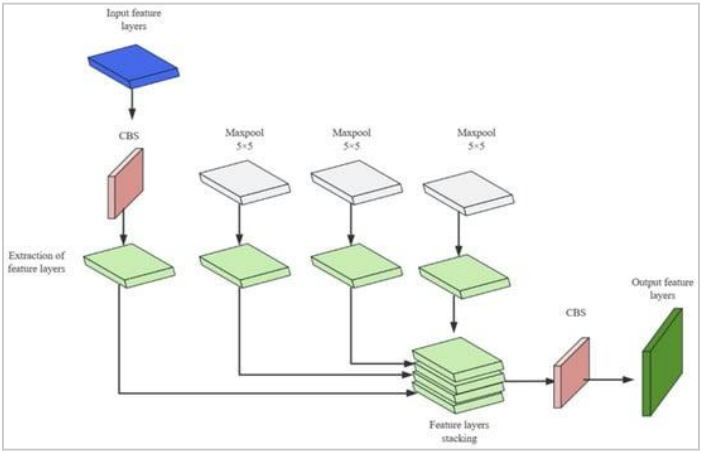

Figure 3.12. Schematic diagram of the SPPF module.

The Neck feature fusion layer obtains shallow image features from the backbone network and concatenates them with deep semantic features to complete the fusion of shallow image features and deep semantic features.

YOLOv5's neck structure is based on the one-way upsampling FPN structure and expanded to the bidirectional PANet structure, as shown in Figure 3.13 and Figure 3.14. By changing the scale of the feature map through interpolation using the upsampling method, the feature map is continuously enlarged to fuse the image features in the backbone network. Different scale feature maps are obtained through downsampling, allowing shallow image features and deep semantic features to complement each other. The neck combines the two paths of different sampling methods to stack the deep features and shallow features of three different scales (20 × 20, 40 × 40, 80 × 80), which are passed layer by layer and finally extracted using the C3 module on the fused features of the three scales, and then passed to the detection layer.

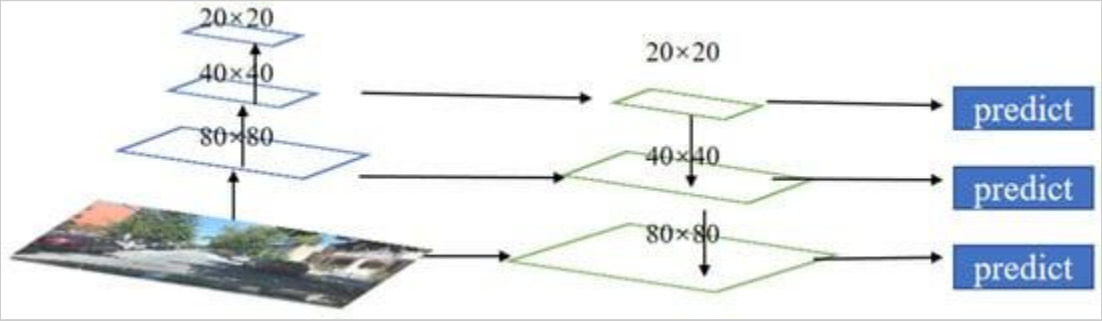

Figure 3.13. Schematic diagram of the FPN fusion structure.

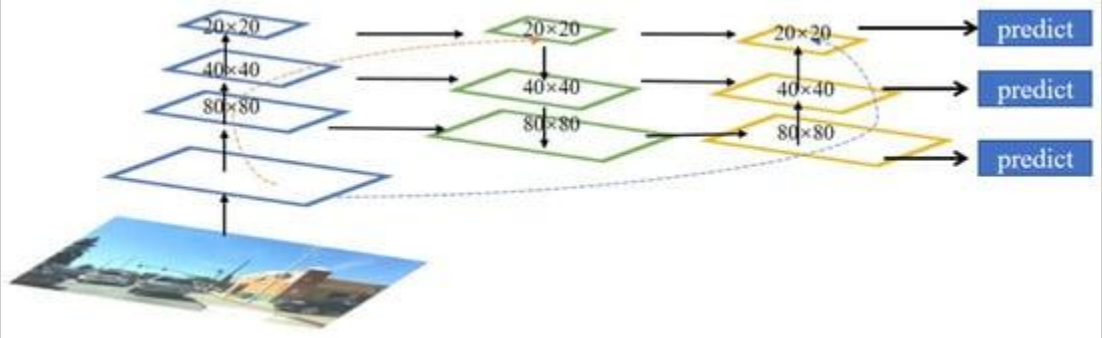

Figure 3.14. Schematic diagram of the PANet fusion structure.

The neck structure effectively fuses shallow and deep features, prevents feature information loss, obtains more complete features, and ensures the feature expression of the detection model for objects of different scales.

The Head output layer is mainly used for detection and consists of the detection module, which includes three 1 × 1 convolutions that correspond to three detection feature layers.

In YOLOv5, the detection layer first divides the three scales of feature maps output by the neck into grids of different scales (80 × 80, 40 × 40, 20 × 20), where each grid corresponds to a pixel, carrying highly condensed feature information. By extracting feature layer information through 1x1 convolutional operations for dimensionality reduction or enhancement, the detection head obtains the position coordinates, categories, and confidence of the anchor in the grid. Then, using anchor boxes of different aspect ratios, the detection layer detects the target object within each grid and adjusts the aspect ratio of the anchor box based on the position information to generate the real box for subsequent detection of position and category information within the box, as shown in Figure 3.15.

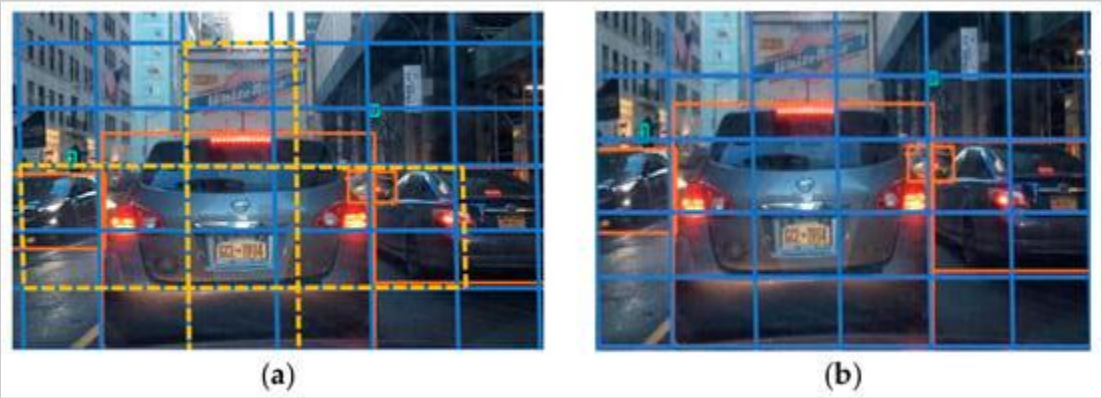

Figure 3.15. Schematic diagram of the detection process. (a) Anchor box. (b) Real box.

3.5 Enhanced You Only Look Once (YOLOv5 + BiFPN)

BiFPN is a novel network structure for multi-scale feature fusion that addresses the issue of traditional one-way FPN not fully utilizing different scale feature information. BiFPN adds a bottom–up feature path to the FPN and achieves multi-scale feature fusion through bidirectional connections and feature fusion on feature nodes in the feature pyramid network, resulting in improved accuracy and efficiency. Traditional FPN only employs a top–down path for feature fusion in the feature pyramid network, leading to the loss of detailed information in lower-resolution features. BiFPN, on the other hand, captures fine-grained details in low-level features by introducing a bottom–up path and fuses them with high-level features. Additionally, BiFPN enables features to propagate and fuse bidirectionally between different levels through its bidirectional connections, further enhancing feature representation. By effectively utilizing features at different scales and implementing bidirectional connections and feature fusion in the network, BiFPN provides more accurate and efficient feature representations for computer vision tasks such as object detection and image segmentation.

In terms of specific implementation, BiFPN removes nodes with only one input edge and adds extra edges between original input and output nodes to fuse more features. Secondly, BiFPN adds a skip connection. A skip connection is added between the input and output nodes in the same scale, which fuses more features at the same layer without adding too much computational cost. In addition, unlike PANet, which only has one top–down and one bottom–up path, BiFPN considers each bidirectional path as a feature network layer and repeats the same layer multiple times to achieve more advanced feature fusion. The structure of BiFPN is shown in the Figure 3.16.

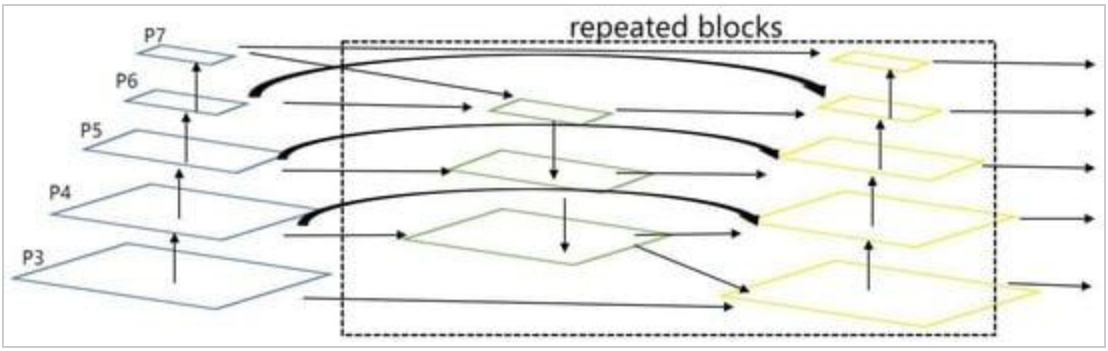

Figure 3.16. Structural diagram of the BiFPN fusion structure.

Weighted Feature Fusion

As different input features have different resolutions, it is crucial to fuse features with different resolutions to improve the accuracy and efficiency of the model. A common method is to adjust features with different resolutions to the same size and then perform addition operation. However, as different features may have different contributions to the output, this method may not achieve the best results. To address this issue, BiFPN proposes a weighted feature fusion method.

BiFPN chooses to add an additional weight for each input feature and lets the network learn the importance of each input feature. To achieve feature fusion, a fast normalization method is adopted, where the weights are divided by the sum of all weights and normalized, as shown in Formula (1):

$$O = \sum_i \frac{w_i}{\varepsilon + \sum_j w_j} \cdot I_i \quad (1)$$

In the formula, O represents the output value, Ii represents the input value of a node, wi represents the weight of the input node, and j represents the sum of all input nodes. The condition wi ≥ 0 is guaranteed by applying the ReLU activation function after each wi, and ε = 0.0001 is a small value used to prevent numerical instability. Similarly, the values

of each normalized weight also fall between 0 and 1. However, since there is no softmax operation involved, the fusion process is more efficient. The final BiFPN integrates both bidirectional cross-scale connections and fast normalized fusion. As an example, we describe here the fusion of two features at level 6 of the BiFPN, as shown in Figure 3.17.

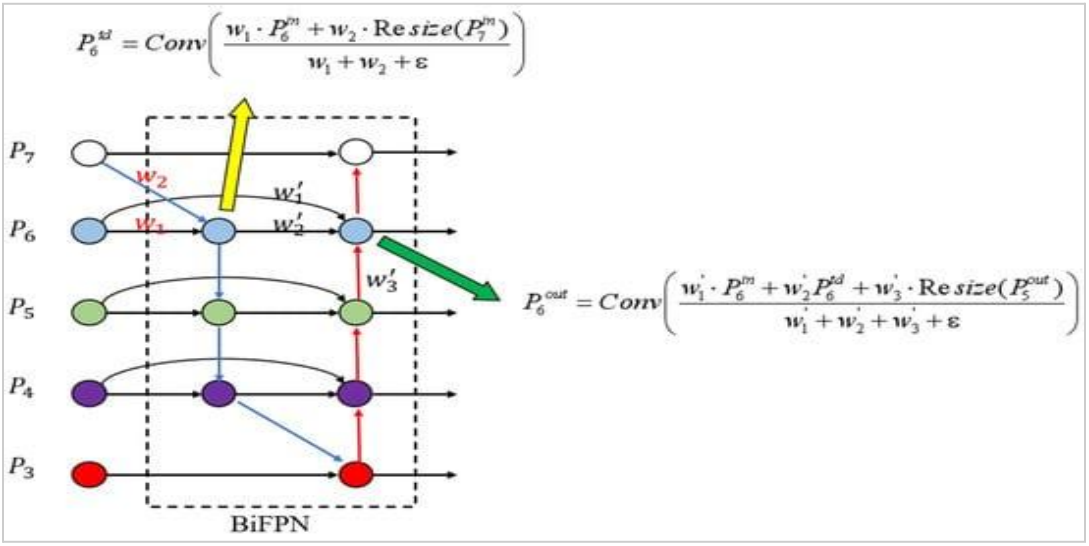

Figure 3.17 . The BiFPN feature fusion process.

$$P_6{}^{td} = Conv\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Re}size(P_7^{in})}{w_1 + w_2 + \varepsilon}\right) \quad (2)$$

$$P_6{}^{out} = Conv\left(\frac{w_1' \cdot P_6^{in} + w_2' P_6^{td} + w_3' \cdot \text{Re}size(P_5^{out})}{w_1' + w_2' + w_3' + \varepsilon}\right) \quad (3)$$

Here, P6td represents the intermediate feature of the sixth layer in the top–down path, P6out represents the output feature of the sixth layer in the bottom–up path, Conv represents the convolution operation, and Resize represents the upsampling or downsampling operation.

3.6 TRAINING :

Parallel to our work on YOLO (You Only Look Once), we have started working on the implementation and improvement of a Faster R-CNN model in our quest to improve object recognition skills. YOLO and Faster R-CNN's respective advantages and disadvantages will be carefully examined in this project, which is designed to address the difficulties presented by real-world driving situations.

3.6.1 Model preparation and data preprocessing :

Much like with YOLO, we started our adventure with Faster R-CNN with a particular focus on data preprocessing. The BDD100K dataset's complexities, which include obstacles like occlusions, varying lighting conditions, and intricate backgrounds in driving scenarios, were thoughtfully considered. A flexible picture pipeline that can extract frames from video sources was designed to demonstrate the usefulness of the approach in real-world situations.

Choosing the best architecture for our Faster R-CNN journey required careful thought, taking into account feature pyramid networks and backbone networks like ResNet. Layers, filters, and anchor boxes were extensively tested in order to find a compromise between computational efficiency and model capacity. To guarantee stability and convergence, hyperparameters pertaining to learning rates, batch sizes, and momentum were meticulously adjusted.

3.6.2 Architectural Decisions and Fine-Tuning :

Choosing the best architecture for our Faster R-CNN journey required careful thought, taking into account feature pyramid networks and backbone networks like ResNet. Layers, filters, and anchor boxes were extensively tested in order to find a compromise between computational efficiency and model capacity. To guarantee stability and convergence,

hyperparameters pertaining to learning rates, batch sizes, and momentum were meticulously adjusted.

3.6.3 Enhancement Techniques and Model Flexibility :

Supplementation techniques, similar to our YOLO approach, were used to increase the model's flexibility. Several augmentation methods were used to strengthen the model's capacity to manage a variety of difficult situations. The architecture of the model was carefully crafted to meet the particular requirements of the BDD100K dataset, guaranteeing optimal performance in object detection.

3.6.4 Iterations in Training and System Flexibility :

Throughout training cycles, flexibility was a key component. By offering the ability to turn off CUDA cores, flexibility was added and compatibility with different processing power levels was guaranteed. Strict quality control procedures were followed when preprocessing the data, eliminating less-than-ideal data points, and carefully checking the labels ahead of time to prevent mistakes.

3.6.5 Basic Methods and Model Effectiveness :

Basic methods such as random image batching were used to encourage robust learning, avoid overfitting, and make effective use of memory. In order to strike a balance between computational efficiency and accuracy, systematic testing driven by methods like cross-validation was required to determine the optimal batch size and epochs.

3.6.6 Post-Processing Techniques :

Post-processing techniques for Faster R-CNN, similar to our YOLO model, were refined to enhance precision and recall. Non-maximum suppression (NMS), confidence

thresholding, and anchor box optimization played pivotal roles in fine-tuning the performance of our Faster R-CNN model.

Our focus on quality goes beyond the improvement of individual models and includes a comprehensive effort to further the field of computer vision. Our goal is still to deliver useful solutions that are exceptional in the complexities of real-world applications, even as we continue to refine both the YOLO and Faster R-CNN models.

CHAPTER 4

RESULT

4.1 EXPERIMENTAL RESULTS

Our investigative findings disclose that YOLO underwent a training regimen spanning 100 epochs, characterized by a diminishing step size of 10^−5 and a batch size of 25. In contrast, Faster R-CNN underwent training for 75 epochs, utilizing a declining learning rate set at 10^−4 with a batch size of 25. The graphical representation of the normalized total loss and mean Average Precision (mAP) trajectories during the training process is depicted in the accompanying figure. For a comprehensive understanding of the implementation details and processed data, we provide access to real-time object detection videos showcasing the performance of both YOLO and Faster R-CNN. In the comparative analysis of YOLO and Faster R-CNN, we conducted assessments of their Frames Per Second (FPS) and mAP on an Apple M1 Pro chip, featuring an 8-core CPU with 6 performance cores, 2 efficiency cores, a 14-core GPU, 16-core Neural Engine, and a memory bandwidth of 200GB/s.

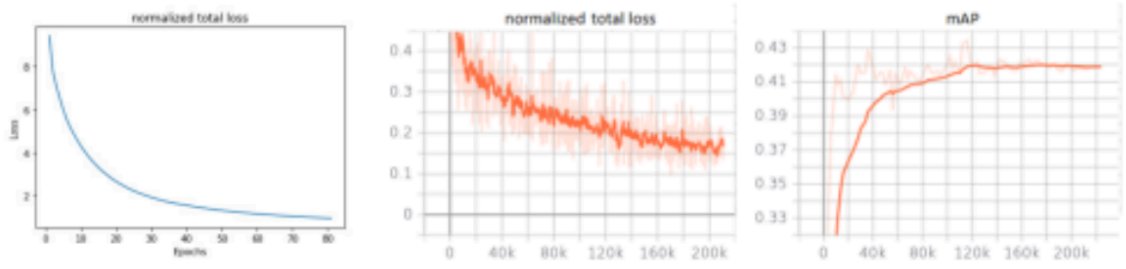

Figure 4.1 Collective loss after normalization for YOLO (positioned on the left), collective loss after normalization for Faster R-CNN (positioned in the middle), and MAP for Faster R-CNN (positioned on the right).

Table 4.1 : Comparison of YOLO, Faster R-CNN, Hybrid Incremental Net, YOLOv5s and YOLOv5s + BiFPN in terms of some standard metrics (mAP) i.e. mean Average Precision (mAP) on the BDD100K dataset.

	mAP
YOLO	18.6
Faster R-CNN	41.8
Hybrid Incremental Net	45.7
YOLOv5s	47.8
YOLOv5s + BiFPN	48.6

Figure 4.2 Outcomes achieved on BDD100K: Faster R-CNN (displayed on the left) and YOLO (shown on the right).

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

Our research focused on the implementation and training of two cutting-edge object recognition models, namely YOLO (You Only Look Once) and Faster R-CNN (Region-based Convolutional Neural Network). The context of our study revolves around the critical demands of split-second decisions and precision in the field of autonomous driving. Leveraging the extensive BDD100K dataset, covering diverse real-world driving scenarios, our analysis brought to light distinct trade-offs between these two leading object detection models.

Faster R-CNN exhibited remarkable accuracy in the precise detection and localization of objects, owing to its advanced two-stage architecture. However, this heightened precision came at the expense of a reduction in Frames Per Second (FPS), illustrating the computational complexity of its intricate two-step detection process. On the other hand, YOLO adopted a speed-centric approach, achieving faster FPS due to its simplified one-stage detection process. Nevertheless, this efficiency was accompanied by a compromise in accuracy, as the streamlined design might not capture subtle details as comprehensively as Faster R-CNN.

As we chart our course for future work, we envisage delving deeper into the evolving landscape of YOLO by exploring its latest iterations. The decision to start with YOLOv1 in this project serves as a foundation for harnessing the capabilities introduced in subsequent versions. Our aim is to continually refine and optimize our models, leveraging advancements in deep learning architectures, optimization techniques, and domain adaptation methods.

Looking ahead, our overarching objective is to strike a harmonious balance between high accuracy and formidable FPS—a vital ratio for the demanding applications of autonomous driving. This entails an ongoing pursuit of innovative models, sophisticated optimizations, and groundbreaking methodologies that bridge the accuracy gap between the speed-centric YOLO and the precision-focused Faster R-CNN. Through these endeavors, we aspire to set new benchmarks in performance, ushering in a new era characterized by unparalleled efficiency and safety in the field of autonomous driving.

CHAPTER 6

FUTURE WORK

6.1 FUTURE WORK

As we navigate our project towards novel horizons, the decision to embrace Faster R-CNN unfolds promising avenues for future exploration. This shift represents a departure from the real-time object detection paradigm inherent in YOLO, emphasizing a more sophisticated approach that prioritizes precision, versatility, and efficacy across diverse applications. Here's a forward-looking perspective on potential areas of focus for our upcoming endeavors:

Adaptability Across Domains :

The versatile nature of Faster R-CNN in managing various object detection routines, including localization and instance segmentation, creates an opportunity for adaptation to distinct domains and scenarios. Future initiatives will involve the implementation of domain adaptation techniques, such as adversarial training or self-training, to enrich the model's generalization across diverse datasets. Explorations into few-shot learning methods seek to empower the model to recognize new object classes with minimal labeled data, thereby bolstering its adaptability in dynamic environments.

Enhancing Robustness:

Tackling challenges posed by occlusions, scale variations, and complex backgrounds is pivotal for fortifying the robustness of Faster R-CNN. Techniques like data augmentation employing realistic synthetic data aim to expose the model to a spectrum of scenarios. Additionally, the incorporation of contextual information and multi-scale features through advanced network architectures or attention mechanisms is anticipated to enhance the model's comprehension of intricate scenes, rendering it more resilient in challenging conditions.

Enhancing Performance:

Faster R-CNN's dual-stage detection mechanism, encompassing region proposal networks (RPN) and object detection networks, promises heightened accuracy. Future endeavors will delve into exploring advanced backbones such as ResNet, DenseNet, or newer architectures like EfficientNet to refine feature extraction. Experimental forays with larger and more diverse datasets, coupled with the application of transfer learning methods, aim to further amplify performance. The integration of attention mechanisms, including self-attention, holds the potential to augment the model's focus on pertinent image segments, thereby enhancing overall accuracy.

Interpretability and Explainability:

Ensuring a profound understanding of the model's decision-making processes is imperative, particularly in critical applications like autonomous vehicles and healthcare. Future research endeavors will concentrate on techniques that provide insights into the model's reasoning process. Methods such as attention visualization or saliency maps are envisioned to offer valuable perspectives into regions of interest, augmenting the model's interpretability and instilling confidence in critical applications.

In summary, the transition from YOLO to Faster R-CNN lays the groundwork for progress in accuracy, efficiency, adaptability, robustness, and interpretability in object detection tasks. By harnessing the latest advancements in deep learning architectures, optimization techniques, and domain adaptation methods, our forthcoming efforts aim to craft highly precise and adaptable object detection systems tailored to specific applications and environments.

CHAPTER 7

REFERENCES

[1] G. Lewis, "Object detection for autonomous vehicles," 2014.

[2] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[3] J. Brownlee, "A gentle introduction to object recognition with deep learning," Machine Learning Mastery, vol. 5, 2019.

[4] F. Yu et al., "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2636–2645.

[5] S. Zhou, J. Zhao, Z. Deng, H. Sun, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 145, 2018.

[6] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[7] Padmane, P.., A Review on Real Time Object Detection Using Deep Learning. 11(4), 2023.

[8] Liu, Z., Gao, Y., Du, Q., Chen, M., & Lv, W.. YOLO-Extract: Improved YOLOv5 for Aircraft Object Detection in Remote Sensing Images, 2023.

[9] Zhou, J., Zhang, B., Yuan, X., Lian, C., Ji, L., Zhang, Q., & Yue, J.. YOLO-CIR: The network based on YOLO and ConvNeXt for infrared object detection. 131, 2023

[10] Bergmann, W.. , SSDA-YOLO: Semi-supervised domain adaptive YOLO for cross-domain object detection. 229, 2023

[11] A. Geiger et al., "Vision meets robotics: The KITTI dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.

[12] Wang, H., & Jiang, S.., Improved Object Detection Algorithm Based on Faster RCNN. 2395(1), 2022

[13] S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," arXiv preprint arXiv:1506.01497, 2015.

[14] Wardhany, V. A.. , Object Detection in Online Proctoring Through Two Camera Using Faster-RCNN. 13(2), 2023

[15] Sheng, W., Yu, X., & Chen, X.. , Faster RCNN Target Detection Algorithm Integrating CBAM and FPN. 13(12), 2023