

MULTIVIEW OBJECT DETECTION AND TRACKING USING DEEP LEARNING TECHNIQUES

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer/ IT Engineering

by

Nirali Anand Pandya
179999913021

under supervision of

Dr. Narendrasinh C. Chauhan



GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

May – 2025

MULTIVIEW OBJECT DETECTION AND TRACKING USING DEEP LEARNING TECHNIQUES

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer/ IT Engineering

by

Nirali Anand Pandya
179999913021

under supervision of

Dr. Narendrasinh C. Chauhan



GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

May – 2025

© Nirali Anand Pandya

DECLARATION

I declare that the thesis entitled Multiview Object Detection and Tracking using Deep Learning Techniques submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from February 2018 to May 2025 under the supervision of Dr. Narendrasinh C. Chauhan and this has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of the Research Scholar:  Date: 02/05/2025

Name of Research Scholar: Nirali Anand Pandya

Place: New Vallabh Vidyanagar

CERTIFICATE

I certify that the work incorporated in the thesis Multiview Object Detection and Tracking using Deep Learning Techniques submitted by Mrs. Nirali Anand Pandya was carried out by the candidate under my supervision/guidance. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of the Research Supervisor:.....Date: 02/05/2025

Name of Supervisor: Dr. Narendrasinh C. Chauhan

Place: New Vallabh Vidyanagar

Course-work Completion Certificate

This is to certify that Mrs.Nirali Anand Pandya enrolment no. 179999913021 is enrolled for PhD program in the faculty Computer / IT Engineering of Gujarat Technological University, Ahmedabad.

(Please tick the relevant option(s))

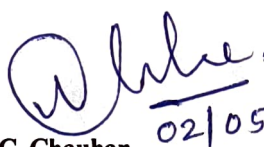
☐ She has been exempted from the course-work (successfully completed during M.Phil Course)

☐ She has been exempted from Research Methodology Course only (successfully completed during M.Phil Course)

☒ She has successfully completed the PhD course work for the partial requirement for the award of PhD Degree. His/ Her performance in the course work is as follows-

Grade Obtained in Research Methodology [PH001]	Grade Obtained in Self-Study Course [PH002]
BB	BB

Supervisor's Sign


02/05/2025

Dr. Narendrasinh C. Chauhan

Originality Report Certificate

It is certified that PhD Thesis titled Multiview Object Detection and Tracking using Deep Learning Techniques by Nirali Anand Pandya has been examined by us. We undertake the following:

- Thesis has significant new work / knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
- The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
- There is no fabrication of data or results which have been compiled / analyzed.
- There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
- The thesis has been checked using Turnitin (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index $\leq 10\%$).

Signature of the Research Scholar: 

Date: 02/05/2025

Name of the Research Scholar: Nirali Anand Pandya

Signature of the Research Supervisor: 


Date: 02/05/2025

Name of the Research Supervisor: Dr. Narendrasinh C. Chauhan

Place: New Vallabh Vidyanagar

Nirali Anand Pandya

MULTIVIEW OBJECT DETECTION AND TRACKING USING DEEP LEARNING TECHNIQUES

 Quick Submit Quick Submit Gujarat Technological University

Document Details

Submission ID

trn:oid::1:3096348300

Submission Date

Nov 29, 2024, 5:06 PM GMT+5:30

Download Date

Nov 29, 2024, 5:19 PM GMT+5:30

File Name

Thesis_final_GTU.pdf

File Size

3.0 MB

83 Pages

19,699 Words

118,241 Characters







7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 14 words)

Match Groups

-  **75 Not Cited or Quoted 7%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 3%  Publications
- 4%  Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

-  **Hidden Text**
294 suspect characters on 2 pages
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 75 Not Cited or Quoted 7%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3% Internet sources
- 3% Publications
- 4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.ijeast.com	1%
2	Student papers	Liverpool John Moores University	1%
3	Internet	www.epfl.ch	0%
4	Internet	arxiv.org	0%
5	Publication	Arjun Paramarthalingam, Jegan Sivaraman, Prasannavenkatesan Theerthagiri, B...	0%
6	Student papers	Kingston University	0%
7	Student papers	University of Technology, Sydney	0%
8	Student papers	Universiti Pertahanan Nasional Malaysia	0%
9	Internet	medium.com	0%
10	Student papers	Bournemouth University	0%

Ph.D. Thesis Non-Exclusive License to

GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a Research Scholar at Gujarat Technological University, and in the interests of the facilitation of research at the University and elsewhere, I, **Mrs. Nirali Anand Pandya** having (Enrollment No 179999913021) hereby grant a non- exclusive, royalty free and perpetual license to the University on the following terms:

- a. The University is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the “Work”) anywhere in the world, for non-commercial purposes, in all forms of media;
- b. The University is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);
- c. The University is authorized to submit the Work at any National / International Library, under the authority of their “Thesis Non-Exclusive License”;
- d. The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;
- e. I undertake to submit my thesis, through my University, to any Library and Archives. Any abstract submitted with the thesis will be considered to form part of the thesis.
- f. I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.
- g. If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.
- h. I understand that the responsibility for the matter as mentioned in the paragraph (g) rests with the authors / me solely. In no case shall GTU have any liability for any acts / omissions / errors / copyright infringement from the publication of the said thesis or otherwise.
- i. I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.
- j. GTU logo shall not be used /printed in the book (in any manner whatsoever) being published or any promotional or marketing materials or any such similar documents.
- k. The following statement shall be included appropriately and displayed prominently in the book or any material being published anywhere: “The content of the published work

is part of the thesis submitted in partial fulfilment for the award of the degree of Ph.D. in Computer / IT Engineering of the Gujarat Technological University”.

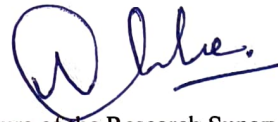
- l. I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this nonexclusive license. I shall keep GTU indemnified from any and all claims from the Publisher(s) or any third parties at all times resulting or arising from the publishing or use or intended use of the book / such similar document or its contents.
- m. I am aware of and agree to accept the conditions and regulations of Ph.D. including all policy matters related to authorship and plagiarism.

Date: 02/05/2025

Place: New Vallabh Vidyanagar

Signature of the Research Scholar

Recommendation of the Research Supervisor:Recommended.....



Signature of the Research Supervisor

Thesis Approval Form

The viva-voce of the PhD Thesis submitted by Smt.Nirali Anand Pandya (Enrollment No 179999913021) entitled Multiview Object Detection and Tracking using Deep Learning Techniques was conducted on 02/05/2025 (Friday) at Gujarat Technological University.

(Please tick any one of the following option)



The performance of the candidate was satisfactory. We recommend that he/she be awarded the PhD degree.




Any further modifications in research work recommended by the panel after 3 months from the date of first viva-voce upon request of the Supervisor or request of Independent Research Scholar after which viva-voce can be re-conducted by the same panel again.


(briefly specify the modifications suggested by the panel)




The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the PhD degree.

(The panel must give justifications for rejecting the research work)


Dr. Narendrasinh C. Chauhan
Name and Signature of Supervisor with Seal


Dr. A. N. Gaikwad
1) (External Examiner 1) Name and Signature


Dr. Sanjay Garg
2) (External Examiner 2) Name and Signature

ABSTRACT

In today's rapidly advancing technological landscape, accurately detecting and tracking objects from multiple viewpoints is crucial in surveillance, autonomous navigation, augmented reality, and robotics. Traditional multiview object detection and tracking methods often struggle with challenges like occlusions, varying viewpoints, and complex scene dynamics. These challenges necessitate the development of more robust and efficient methodologies. Deep learning, a subset of artificial intelligence, has emerged as a powerful tool to address these issues, offering exceptional capabilities in learning complex patterns and representations directly from data. This study begins by reviewing the existing literature on multiview object detection and tracking, highlighting the limitations of traditional methods. It then explores deep learning techniques and their applications to multiview data fusion for object detection and tracking tasks. The research is structured into three main parts.

Firstly, it investigates data augmentation techniques using state-of-the-art deep learning models, such as Faster Region-based Convolutional Neural Network (Faster RCNN), Single Shot Detector (SSD), CenterNet, and EfficientDet. Focusing on the Open Image Dataset, the method applies data augmentation strategies to enhance training data and improve detection performance for three vehicle classes – cars, buses, and bicycles. The effectiveness of various augmentation techniques and detection models are analyzed through rigorous experimentation and evaluation.

Secondly, the thesis investigates hybridizing two object detection and tracking methods in a multiview environment. In this method, a combination of a popular deep learning object detection model You Only Look Once (YOLO) and the Simple Online and real-time tracking with a Deep Association Metric (DeepSORT) tracking algorithm, which examines object detection and tracking across multiple cameras. The proposed technique was tested on the EPFL Multi-view Multi-class Detection dataset. Extensive experiments evaluate the proposed approach's performance in scenarios involving multiple object classes and varying viewpoints. The third proposed method explores the use of a combination of YOLO and ByteTrack (a multiobject tracking algorithm) for tracking pedestrians in a multi-camera environment. This approach was tested on the EPFL multi-camera pedestrian video dataset. The research aims to develop robust techniques that effectively monitor objects across several camera views in challenging conditions, including poor lighting, different viewpoints, and occlusions.

Overall, this research advances the investigation and development of novel deep-learning techniques for multiview object detection and tracking. It contributes insights and practical solutions for surveillance, transportation, and security applications, focusing on data augmentation for improved detection accuracy and multi-camera systems for robust object monitoring and tracking.

Acknowledgement

I am deeply grateful to the almighty for the strength, patience, and knowledge granted to me to undertake and complete this research study. This achievement would not have been possible without such blessings.

I am profoundly thankful to my guide, Dr. Narendrasinh C. Chauhan, professor & head of the information technology department at A. D. Patel Institute of Technology, Gujarat, for his invaluable guidance, motivation, and support throughout this journey. His continuous encouragement and insightful feedback were crucial in shaping my research and transforming initial ideas into effective solutions.

My sincere thanks also go to the Doctoral review Committee members of my research, Dr. Narendra M. Patel from the computer engineering department at BVM Engineering College, and Dr. Tanmay D. Pawar, professor and head of the electronics department at BVM Engineering College. Their assistance, collaboration, and constructive feedback significantly enriched the quality of this work.

I extend my gratitude to the management of Charutar Vidya Mandal, Vallabh Vidyanagar, and the principal and faculty members of Madhuben and Bhanubhai Patel Institute of Technology at New Vallabh Vidyanagar for their cooperation and support throughout my research.

A heartfelt thank goes to my family, my parents and brother. I would like to express my heartfelt gratitude to my late mother, Late. Smt. Pragnaben Thakkar, whose love, support, and guidance continue to inspire me. I am deeply grateful to my husband, Anand Pandya, for being my steadfast companion through every challenge. His unwavering support and encouragement have been invaluable throughout my research journey. I am also deeply thankful for my loving daughter, Trushti Pandya, whose joy and love provided comfort during challenging times. Finally, I am grateful to all who have directly or indirectly supported me during this work.

Nirali Anand Pandya (Enr.No.179999913021)

Research Scholar,

Gujarat Technological University

Table of Contents

Abstract	xii
Acknowledgement	xiv
Tables of Contents	xv
List of Abbreviation	xviii
List of Figures	xix
List of Tables	xxi
CHAPTER-1 Introduction	1
1.1 Introduction to Object Detection and Tracking	1
1.2 Need for Multiview Object Detection and Tracking	2
1.3 Open Research Challenges in Multiview Systems	4
1.4 Motivation of Work	5
1.5 Definition of Problem	6
1.6 Objectives of Research	6
1.7 Scope	7
1.8 Important Thesis Contributions	7
1.9 Organization of Thesis	8
CHAPTER-2 Literature Survey	11
2.1 Overview of Object Detectors	11
2.1.1 RCNN	13
2.1.2 Fast RCNN	14
2.1.3 Faster RCNN	15
2.1.4 CenterNET	16
2.1.5 EfficientDet	16
2.1.6 YOLO	17
2.1.7 YOLOv7	18
2.1.8 YOLOv8	19

2.2	Multi-view Object Detection Methods	21
2.3	Overview of Object Tracking	23
2.3.1	SORT	25
2.3.2	DeepSORT	26
2.3.3	ByteTrack	26
2.4	Multi-view Object Tracking	27
2.5	Evaluation Parameters	28
2.5.1	Evaluation Parameters for Object Detection	29
2.5.2	Evaluation Parameters for Object Tracking	30

CHAPTER-3 Enhancing Multiview Object Detection through Image Data Augmentation and Deep Neural Networks

3.1	Introduction	32
3.2	Image Data Augmentation	33
3.2.1	Methods of Image Data Augmentation	33
3.2.2	Benefits of Image Data Augmentation	35
3.2.3	Applications in Multiview Object Detection	36
3.3	Proposed Approach	40
3.4	Experiment and Results	40
3.4.1	Implementation Detail	41
3.4.2	Dataset	41
3.4.3	Results	45
3.5	Application of Multi-View Object Detection in Autonomous Driving using Deep Learning Approach	45
3.6	Conclusion and Discussion	49

CHAPTER-4 Multi-camera Object Detection and Tracking: A YOLOv7 and DeepSORT-Based Approach

4.1	Overview	52
4.2	Proposed Approach	52
4.2.1	YOLOv7	52

4.2.2	Object Tracking with YOLOv7 with DeepSORT	53
4.3	Experiments and Results	57
4.3.1	Implementation Detail	58
4.3.2	Results	58
4.4	Conclusion and Discussion	64
 CHAPTER-5 Multi-Camera Object Tracking using YOLOv8 and ByteTrack on Multi-camera Pedestrians Video Dataset		66
5.1	Overview	66
5.2	Proposed Approach	67
5.2.1	YOLOv8 and ByteTrack Overview	67
5.2.2	Multi-Threaded Tracking	70
5.2.3	Dataset Description and Preprocessing	70
5.2.4	Proposed Algorithm for Multi-camera Object Tracking	71
5.3	Results	73
5.4	Conclusion and Discussion	79
 CHAPTER-6 Conclusion and Future Scope		81
6.1	Conclusion	81
6.2	Future Scope	83
	References	84
	List of Publications	96

List of Abbreviation

AP	Average Precision
CNN	Convolutional Neural Networks
DDFD	Deep Dense Face Detector
DeepSORT	Simple Online and Realtime Tracking with a Deep Association Metric
DNN	Deep Neural Network
ELAN	Enhanced Layer Aggregation Network
FPN	Feature Pyramid Network
GFLOP	Giga Floating-Point Operations per Second
GPU	Graphical Processing Unit
GT	Ground Truth
HOG	Histogram of Oriented Gradient
IoU	Intersection over Union
mAP	Mean Average Precision
MOT	Multi-Object Tracking
MOTA	Multi Object Tracking Accuracy
MOTP	Multi Object Tracking Precision
NMS	Non-maximum suppression
RCNN	Region-based Convolutional Neural Network
RESNET	Residual Network
RNN	Recurrent neural networks
ROI	Region of Interest
SIFT	Scale-Invariant Feature Transform
SORT	Simple Online and Realtime Tracking
SSD	Single Shot Detector
YOLO	You Only Look Once

List of Figures

Figure 1.1	Major contributions of Thesis	8
Figure 2.1	Object detection methods	11
Figure 2.2	Object detectors (a) Two-stage Detector (b) One-stage Detector	12
Figure 2.3	Methods for object tracking	24
Figure 3.1	Architecture of multiview object detection using image data augmentation and deep neural networks	37
Figure 3.2	Detail steps of Multiview Object Detection using Image Data Augmentation	38
Figure 3.3	Sample annotation file in XML format	39
Figure 3.4	Sample images after applying Image Data Augmentation	40
Figure 3.5	Comparison of Deep learning models with and without augmentation at Train-test ratio 90%-10%	43
Figure 3.6	Comparison of Deep learning models with and without augmentation at Train-test ratio 80%-20%	43
Figure 3.7	Comparison of Deep learning models with and without augmentation at Train-test ratio 70%-30%	44
Figure 3.8	Sample Output Images of Multiview Object Detection on Open Image Dataset	44
Figure 3.9	Sample Output Images of Multiview Object Detection on Open Image Dataset	45
Figure 3.10	Steps applied for object detection using YOLOv8 Object Detector	47
Figure 3.11	The convergence of both training and validation losses for the YOLOv8 algorithm object detector and classification is observed at 100 epochs	49
Figure 3.12	Sample Output (Udacity Car Dataset)	49
Figure 4.1	Proposed Model for object detection and tracking based on YOLOv7 and DeepSORT	54
Figure 4.2	Sample annotated image in YOLOv7 format using labelling tool	59
Figure 4.3	Sample annotations of the image shown in the previous figure	59

Figure 4.4	The convergence of both training and validation losses for the YOLOv7 algorithm object detector and classification is observed at 300 epochs, as demonstrated on the Multi-view multi-camera dataset.	61
Figure 4.5	(a) illustrates the precision (P) plotted against confidence (C) (b) demonstrates the recall plotted against confidence. (c) Correspond to the mean average precision, which is calculated by comparing the ground truth bounding boxes with the detected bounding boxes. (d) highlights the F1 score, reaching 93% at a confidence level of 0.449. This score emphasizes the balance between precision and recall, as observed in the Multi-view multi-camera dataset	62
Figure 4.6	(a) and (b) Sample output of object tracking using DeepSORT	63
Figure 5.1	Propose YOLOv8 and ByteTrack for Multi-Camera Object Tracking	68
Figure 5.2	Algorithmic Steps YOLOv8 and ByteTrack for Multi-Camera Object Tracking	72
Figure 5.3	Sample annotated image taken from Roboflow	73
Figure 5.4	Annotations for person in YOLOv8 format shown in figure 5.3	74
Figure 5.5	(a) illustrates the precision (P) plotted against confidence (C) (b) Demonstrates the recall plotted against confidence. (c) correspond to the mean average precision, which is calculated by comparing the ground truth bounding boxes with the detected bounding boxes. Additionally (d) highlights the IDF1 score	76
Figure 5.6	Sample video file and annotations	77
Figure 5.7	Sample video frames for the passageway sequence	78
Figure 5.8	Sample video frames for the laboratory sequence	78
Figure 5.9	Sample video of multi-camera person dataset with non-overlapping camera view – Laboratory Sequence	79
Figure 5.10	Sample output of multi-camera person dataset with non-overlapping camera view – Main Entry	79

List of Tables

Table 2.1	Comparison of the speed and accuracy of different object detectors on the MS COCO dataset (test-dev 2017).	13
Table 3.1	Performance of object detection models without Image Data Augmentation on Open Image Vehicle Dataset V6	42
Table 3.2	Performance of object detection models with Image Data Augmentation on Open Image Vehicle Dataset V6	42
Table 3.3	Performance of object detection modes for vehicle detection	48
Table 4.1	Performance evaluation of Fine-tuned YOLOv7 on EPFL Multi-View Multi-Camera Dataset	60
Table 4.2	Performance evaluation of Fine-tuned YOLOv7 + DeepSORT on EPFL Multi-View Multi-Camera Dataset	60
Table 5.1	Performance evaluation of Fine-tuned YOLOv8 on Open Image Dataset + ByteTrack on Multi-camera Pedestrian Dataset	74
Table 5.2	Performance evaluation ByteTrack on Multi-camera Pedestrian Dataset and Real-time Multi-camera person dataset with non-overlapping camera view	75
Table 5.3	Data Format for Evaluation of Object Tracking	77

CHAPTER – 1

Introduction

1.1 Introduction to Object Detection and Tracking

Object detection and tracking are transformative technologies that enable machines to recognize, locate, and follow objects within visual data, opening powerful new capabilities across numerous fields. These technologies, from autonomous vehicles and drones to smart surveillance systems and healthcare robots, provide critical situation awareness by identifying and tracking objects in real time. By analyzing video frames or live feeds, systems can not only detect objects like people, vehicles, or specific items but also monitor their movements over time, supporting dynamic, interactive, and responsive applications. This ability to interpret and respond to visual data with precision is fueling innovation in safety, automation, and human-robot interaction, driving the next generation of intelligent systems.

Object detection involves identifying and localizing objects within an image or video frame, while object tracking entails following the movement of objects over time across consecutive frames. Traditional approaches to object detection relied on handcrafted features and machine learning classifiers. However, recent advancements in deep learning have revolutionized object detection by enabling end-to-end learning of feature representations directly from data. Deep learning-based object detection methods typically utilize convolutional neural networks (CNNs) to extract hierarchical features from input images and employ region proposal algorithms, such as R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6], to generate candidate object bounding boxes. These bounding boxes are then refined and classified into different object categories using region-based or anchor-based classification techniques. One of the significant breakthroughs in object detection is the introduction of single-stage detectors, such as – you only look once (YOLO) [10] and single shot detector (SSD) [7], which can simultaneously predict object bounding boxes and class probabilities in a single pass through the network. These models offer real-time performance and are well-suited for applications requiring low latency.

Object tracking involves following the spatial and temporal trajectory of objects across consecutive frames in a video sequence. The primary goal of object tracking is to maintain consistent identities for objects over time, even in the presence of occlusions, motion blur, and changes in appearance. Traditional object tracking methods often relied on handcrafted features and motion models to estimate the state of objects and associate them between frames. However, deep learning has also made significant strides in improving object tracking accuracy and robustness.

Deep learning-based object tracking methods typically formulate object tracking as a regression or classification problem and use recurrent neural networks (RNNs) or convolutional neural networks (CNNs) to predict the position or motion of objects in subsequent frames. Reinforcement learning techniques, such as deep reinforcement learning (DRL), have also been applied to learn optimal tracking policies directly from data. Hybrid approaches that combine the strengths of traditional and deep learning-based methods have shown promising results in object tracking. These approaches leverage the complementary capabilities of handcrafted features and deep feature representations to achieve robust and accurate tracking performance across various scenarios.

1.2 Need for Multiview Object Detection and Tracking

Single-view systems face significant limitations in object detection and tracking, primarily due to their restricted field of vision and lack of depth perception. Since they capture scenes from only one angle, these systems struggle with occlusions—if an object is blocked by another, it may go undetected or be lost during tracking. Additionally, single-view systems cannot accurately determine depth, making it difficult to gauge an object's distance or spatial positioning, which is critical in applications like autonomous navigation and 3D modeling. Their limited perspective also affects their ability to handle crowded or complex environments, where objects might overlap or appear similar. Moreover, single-view systems are more susceptible to environmental variations, such as changes in lighting or shadows, as they lack alternative viewpoints to verify and adjust for these inconsistencies.

Multiview object detection and tracking involve the simultaneous analysis of data from multiple cameras or viewpoints to identify and track objects within a scene accurately. The

need for multiview object detection and tracking arises from the inherent limitations of single-view systems and the increasing demands for more comprehensive and accurate scene understanding in various applications. Multi-view systems help overcome many of the limitations of single-view systems and offers the following advantages:

With multiple viewpoints, multi-view systems can continue tracking objects even if they are obscured from one camera's perspective, providing more consistent tracking. Multi-view systems can more accurately estimate depth and spatial positioning, which is especially valuable for applications requiring precise location data, like robotics or autonomous vehicles. Multiview systems provide coverage from multiple viewpoints, enhancing the overall visibility of the scene and reducing blind spots. By integrating information from different views, multiview object detection and tracking systems can compensate for occlusions, shadows, and other factors that may hinder visibility in a single view. Multiview systems provide richer contextual information about the scene by capturing object interactions, scene dynamics, and spatial relationships from multiple perspectives. This holistic view enables more comprehensive scene understanding and facilitates higher-level reasoning tasks, such as activity recognition, scene understanding, and behaviour analysis. Multiview object detection and tracking increase system robustness by providing redundancy and resilience to failures or occlusions in individual camera views. By aggregating information from multiple sources, these systems can mitigate the effects of noise, sensor errors, and environmental variability, leading to more reliable object detection and tracking performance.

Multiview object detection and tracking enable accurate 3D localization and reconstruction of objects within a scene. By triangulating information from multiple camera views, these systems can estimate the spatial positions and dimensions of objects in three-dimensional space, facilitating applications such as augmented reality, autonomous navigation, and 3D scene modelling. Multiview object detection and tracking are crucial for surveillance and security applications, where comprehensive monitoring and accurate tracking of objects are essential for threat detection, anomaly detection, and situational awareness. By integrating information from multiple cameras, these systems can provide comprehensive coverage of large areas and improve the effectiveness of surveillance operations. Multiview object detection and tracking play a vital role in advancing autonomous systems, including

autonomous vehicles, drones, and robots. By integrating information from multiple sensors or camera views, these systems can perceive and interpret the surrounding environment more accurately, enabling safer navigation, obstacle avoidance, and interaction with the environment.

While single-view systems are simpler and more cost-effective, they struggle with occlusion, depth estimation, and spatial awareness, multi-view systems, though more complex, provide superior robustness, accuracy, and continuity in object detection and tracking. They play a crucial role in advancing the capabilities of computer vision systems and addressing the diverse needs of various real-world applications.

1.3 Open Research Challenges in Multiview Systems

Research challenges in multi-view object detection and tracking encompass various technical and practical aspects that need to be addressed to advance the state-of-the-art in the field.

- *Scalability and efficiency:* Addressing the computational complexity and scalability of multi-view object detection and tracking algorithms, especially in real-time applications. Developing efficient algorithms and optimization techniques to handle large volumes of data from multiple views while maintaining real-time performance is crucial.
- *Generalization and adaptation:* Ensuring the robustness and generalization of multi-view object detection and tracking algorithms across diverse environments, lighting conditions, and object types. This involves developing techniques for domain adaptation, transfer learning, and robust feature representations to enhance algorithm adaptability.
- *Integration with other sensor modalities:* Information from multiple sensor modalities, such as LiDAR, radar, and thermal sensors, with multi-view camera data to enhance object detection and tracking performance. Developing fusion techniques and sensor calibration methods to effectively combine data from heterogeneous sensors is crucial.
- *Data fusion and alignment:* Developing robust methods for fusing information from multiple camera views while accounting for differences in viewpoint, resolution, and imaging characteristics. This includes accurate camera calibration, synchronization, and geometric alignment to ensure spatial and temporal coherence across views.

- *Object association and occlusion handling:* Designing effective algorithms for associating objects detected in different camera views and handling occlusions and interactions between objects. This involves resolving ambiguities in object correspondence and maintaining consistent object identities over time despite occlusions and scene dynamics.

Addressing these challenges requires interdisciplinary collaboration between researchers from computer vision, machine learning, robotics, and other related fields. By tackling these challenges, we can advance the state-of-the-art in object detection and tracking and enable the development of more reliable, accurate, and efficient systems for various applications.

1.4 Motivation of Work

Single-view systems often struggle with accuracy due to limited perspectives, leading to incomplete or erroneous detections. Multiview systems can observe the same scene from different angles, significantly improving the accuracy and reliability of object detection and tracking. Objects in real-world scenarios frequently become partially or fully obscured, causing traditional single-view systems to miss detections or incorrectly track objects. Multiview approaches offer alternative viewpoints, enhancing the system's ability to maintain accurate tracking even when occlusions occur. Accurate depth perception is crucial for understanding the spatial relationships between objects, especially in applications like autonomous driving and robotics. Multiview setups provide the necessary parallax, resulting in more robust and precise depth estimation.

The ability to scale and adapt to different environments is a significant advantage of multiview systems. By integrating additional cameras or viewpoints, these systems can be tailored to cover larger areas or more complex scenes, making them highly versatile for various applications. Recent advancements in deep learning, particularly in convolutional neural networks (CNNs) and transformers, have shown remarkable improvements in object detection and tracking performance. Applying these techniques to multiview systems can further enhance their capabilities, offering new solutions to longstanding challenges. The demand for reliable and accurate object detection and tracking systems is growing across numerous fields, including autonomous vehicles, surveillance, and robotics. Multiview systems powered by deep learning have the potential to meet these demands, providing safer

and more efficient solutions. While significant progress has been made in single-view object detection and tracking, multiview systems remain a relatively underexplored area with immense potential. This research aims to push the boundaries by exploring how deep learning techniques can effectively apply to multiview scenarios, opening new avenues for innovation and development. By addressing these motivations, the research seeks to contribute to advancing computer vision technology, offering practical solutions that enhance the performance and applicability of object detection and tracking systems in real-world scenarios.

1.5 Definition of problem

This research aims to design and develop advanced algorithms for multi-view object detection and tracking, focusing on enhancing both accuracy and efficiency. Specifically, the study seeks to create robust multi-view object detection algorithms capable of accurately identifying objects from various camera perspectives, regardless of variations in angle, position, or orientation. In parallel, the research emphasizes the development of efficient object-tracking algorithms designed to reliably maintain object identity across multiple views over time, even in complex and dynamic environments.

A key objective of this work is the seamless integration of these detection and tracking components into a unified, real-time system. By leveraging state-of-the-art deep learning techniques, the research endeavors to optimize the performance of both the detection and tracking processes. This involves improving the accuracy of object recognition across diverse views and enhancing the reliability of tracking mechanisms to ensure consistent performance under real-world conditions.

Ultimately, the study aspires to contribute to the advancement of multi-view object detection and tracking by delivering a cohesive framework that addresses existing challenges in this field. By focusing on both algorithmic innovation and practical implementation, this research has the potential to significantly improve real-time applications in areas such as surveillance, autonomous vehicles, and smart environments.

1.6 Objectives of Research

The primary objectives of the research are as follows:

- Design and develop deep learning-based algorithms for object detection and tracking in multiview environments for different environmental conditions.
- Explore methodologies for improving robustness and model performance for multiview object detection tasks through data augmentation methods.
- Investigate and utilize deep learning based advanced tracking algorithms to track objects as they move across multiple camera views.
- Validate the proposed methods through multiview or multi-camera datasets.

1.7 Scope

The scope of the research encompasses several key aspects related to leveraging deep learning techniques for addressing the challenges of object detection and tracking across multiple camera views. The article will explore how deep neural networks, a core component of deep learning, are utilized for object detection and tracking in scenarios with multiple viewpoints. This refers to detecting objects in environments where multiple cameras capture the scene from different angles. The article will likely explore how deep learning can handle the complexities of combining information from these various viewpoints to improve detection accuracy, especially for small objects. After objects are detected in each view, the additional challenge is to track them across multiple frames or views. The article discusses deep learning methods for establishing consistent object identities despite variations in appearance due to different viewpoints or occlusions.

1.8 Important Thesis Contributions

The important contribution of the thesis is shown in Figure 1.1:

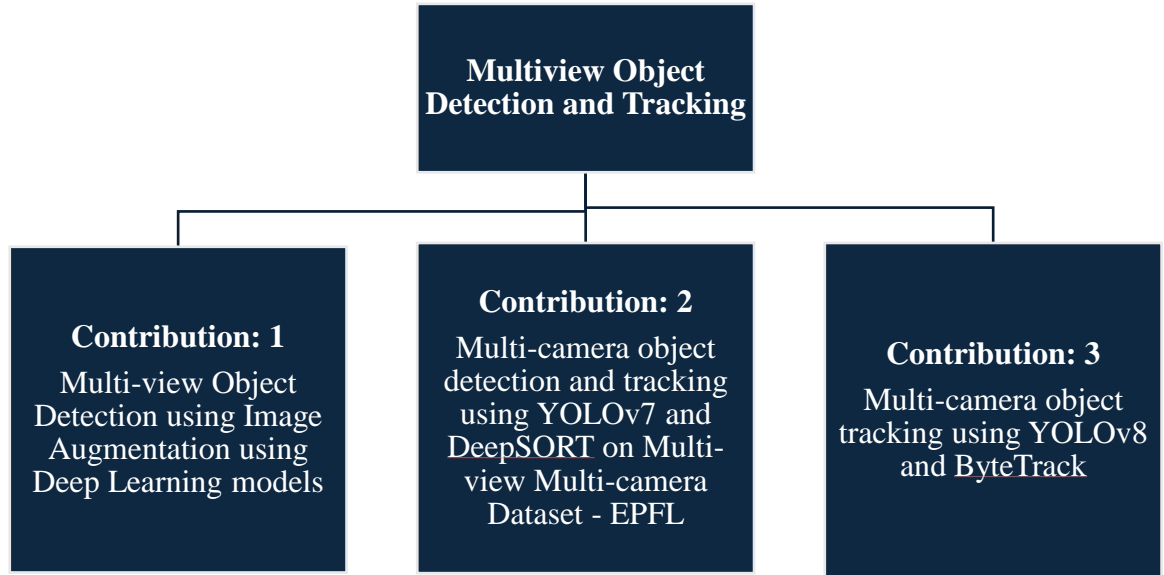


FIGURE 1.1 Major Contribution of Thesis

- **Multi-view object detection using image augmentation using deep learning models:** This component investigates the application of advanced data augmentation techniques to enhance multi-view object detection. By employing state-of-the-art deep learning models such as Faster RCNN, SSD, CenterNet, and EfficientDet, the study aims to improve detection accuracy for various object categories. The Open Image dataset v6 [26] serves as the primary source of training data, enriched through augmentation strategies to better handle occlusions, viewpoint variations, and dynamic scenes.
- **Multi-camera object detection and tracking using YOLOv7 and DeepSORT:** This segment focuses on multi-camera object detection and tracking, leveraging the YOLOv7 detection model alongside the DeepSORT tracking algorithm. The research utilizes the EPFL Multi-view Multi-camera Object Detection dataset to evaluate the effectiveness of the proposed approach in handling multiple object classes across different camera viewpoints. The integration of YOLOv7 and DeepSORT aims to enhance tracking accuracy and robustness in complex environments.
- **Multi-camera object tracking using YOLOv8 and ByteTrack:** This component explores the use of YOLOv8 and ByteTrack for tracking pedestrians in multi-camera setups. The study employs multiple datasets, including the person subset of the Open Image dataset v6, the CVLAB's Multi-camera Pedestrian dataset from EPFL, and a real-time multi-camera

person dataset with non-overlapping views. The goal is to develop robust tracking systems capable of effectively monitoring individuals across several camera views, even in challenging conditions such as poor lighting and occlusions.

1.9 Organization of Thesis

The structure of this thesis is organized into six chapters, each dedicated to a specific aspect of the research, facilitating a comprehensive understanding of the study.

Chapter 1 provides an introduction to the research, laying the groundwork for the study. It explains the motivation behind investigating multi-view object detection and tracking, highlights the objectives and scope of the work, and presents an overview of the proposed research contributions. This chapter sets the stage by outlining the relevance and importance of the chosen topic.

Chapter 2 is dedicated to a thorough review of the existing literature, offering a detailed analysis of both traditional methods and contemporary deep learning techniques employed in object detection and tracking. This chapter provides the necessary background and context for the research, identifying gaps in the existing body of knowledge that the thesis aims to address.

Chapter 3 focuses on enhancing multi-view object detection. It explores the implementation of advanced image data augmentation techniques and the application of deep neural networks to improve detection accuracy and robustness. The methodologies and experimental results presented here contribute to the refinement of multi-view detection approaches.

Chapter 4 shifts attention to multi-view, multi-camera object detection and tracking. It investigates the integration of deep learning methodologies to process and analyze data from multiple cameras, offering innovative solutions to challenges in this domain.

Chapter 5 narrows the focus to multi-camera object tracking. It emphasizes the application of state-of-the-art deep learning techniques, including the ByteTrack algorithm, to achieve efficient and accurate tracking of objects across multiple camera views.

Finally, Chapter 6 concludes the thesis by summarizing the key findings of the research. It reflects on the contributions made to the field and identifies potential directions for future work, aiming to inspire continued advancements in multi-view object detection and tracking.

CHAPTER – 2

Literature Review

This chapter reviews techniques employed in multi-view object detection and tracking. It serves as an introductory guide, outlining key concepts essential for understanding the thesis at hand. The examination of pertinent literature predominantly revolves around machine learning and deep learning principles. Ultimately, the chapter wraps up with insightful reflections derived from the extensive review of relevant literature.

2.1 Object Detectors

Object detection has seen significant advancements in recent years and is poised to play an increasingly vital role in future technological developments. The evolution of object detection began with the Viola-Jones detector [1][2], enabling real-time human face detection, followed by the adoption of Histogram of Oriented Gradient (HOG) detectors [3] for pedestrian detection. HOG detectors evolved into Deformable Part-based Models (DPMs), pioneering multiple object detection. The introduction of the regions with convolutional neural network (R-CNN) [4] model in 2014 marked a breakthrough in deep learning-based object detection, enhancing Mean Average Precision (mAP). Subsequent advancements in deep neural networks and GPU technology facilitated faster and more efficient real-time object detection. Figure 2.1 depicts a taxonomy of modern deep learning-based object detectors.

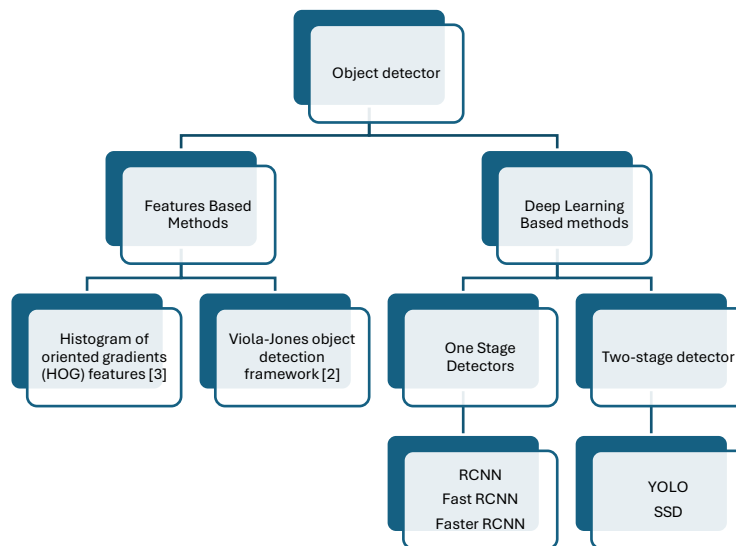


FIGURE 2.1 Object detection methods

Presently, an array of deep learning algorithms is employed for object detection, broadly categorized into one-stage and two-stage detectors [93]. Two-stage deep learning-based detectors propose regions and then classify objects. First, they generate Regions of Interest (ROIs) that are likely to contain objects, followed by ROI selection and object classification. Examples include RCNN [4], Fast R-CNN [5], and Faster R-CNN [6]. In contrast, one-stage detectors directly produce object bounding boxes without the intermediate step of region proposal. This characteristic renders these algorithms faster, less computationally intensive, and suitable for real-time applications. Widely used one-stage detectors include YOLO [10], SSD [7], EfficientNet [8], and CenterNet [9]. Figure 2.2 illustrates the distinction between these two detector types.

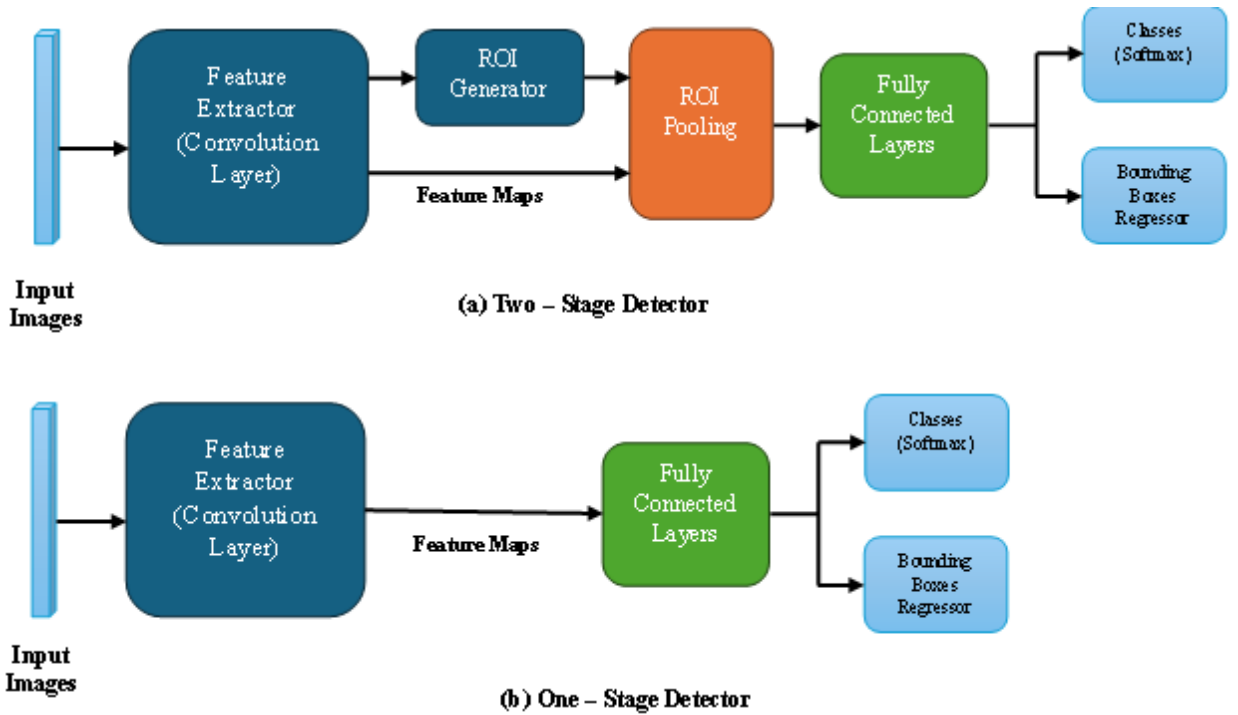


FIGURE 2.2 Object detectors (a) Two-stage Detector (b) One-stage Detector

A comparison of speed and accuracy metrics across multiple object detection models [56] on the MS COCO [32] dataset is depicted in the table, facilitating a detailed understanding of their relative performance.

TABLE 2.1 Evaluation of the performance and precision of various object detection models on the MS COCO test-dev 2017 dataset.

Method	Dataset	Backbone	Size	AP	AP ₅₀	AP ₇₅
RetinaNet[13]	MS COCO	ResNet-50 [33]	640	37.0%	-	-
RetinaNet [13]	MS COCO	ResNet-101 [33]	640	37.9%	-	-
YOLOv3 [16]	MS COCO	Darknet-53	608	42.4%	63.0%	47.4%
EfficientDet-D0 [8]	MS COCO	Efficient-B0 [35]	512	3.8%	52.2%	42.1%
EfficientDet-D1 [8]	MS COCO	Efficient-B1 [35]	640	39.6%	58.6%	45.1%
EfficientDet-D2 [8]	MS COCO	Efficient-B2 [35]	768	43.0%	62.3%	47.4%
EfficientDet-D3 [8]	MS COCO	Efficient-B3 [35]	896	45.8%	65.0%	49.2%
YOLOv4 [14]	MS COCO	CSPDarknet-53	608	43.5%	65.7%	47.3%
SSD [7]	MS COCO	Mobilenet [34]	640	-	29.1%	-
CenterNet [9]	MS COCO	Resnet 101 [33]	512	-	34.2	-
Faster RCNN [6]	MS COCO	Resnet 101 [33]	640	-	31.8	-

2.1.1 RCNN

RCNN , or region-based convolutional neural network, is a seminal deep learning-based object detection model. Introduced in 2014 by Girshick et al.[4], RCNN was a breakthrough in object detection, significantly advancing the state-of-the-art performance. RCNN generates region proposals, or candidate bounding boxes, using selective search or a similar method. These proposals are regions within the image that are likely to contain objects. Each region proposal is warped to a fixed size and passed through a pre-trained CNN, such as VGG [11], to extract a feature representation. The extracted features from each region proposal are then fed into a set of support vector machines (SVMs) [94] to classify the presence of objects and their

respective classes. This step determines whether the region proposal contains an object and, if so, which class it belongs to. Additionally, bounding box regression is applied to refine the location of the bounding boxes, adjusting their position and size to better fit the object within the proposal. Finally, non-maximum suppression is performed to eliminate redundant and overlapping bounding boxes, retaining only the most confident detections.

While RCNN achieved impressive detection accuracy, its main drawback was its slow inference speed due to the need to process each region proposal individually through CNN. This led to the development of faster variants such as Fast R-CNN and Faster R-CNN, which improved upon the efficiency of the original RCNN model while maintaining its accuracy.

2.1.2 Fast RCNN

Fast R-CNN is an improvement over the original RCNN (Region-based Convolutional Neural Network) model, designed to address its computational inefficiency. Introduced by Ross Girshick in 2015 [5], Fast R-CNN offers faster inference speeds while maintaining high object detection accuracy. Instead of generating region proposals separately, as in RCNN, Fast R-CNN shares the entire image's convolutional features to generate region proposals. This is typically achieved using an algorithm like selective search or edge boxes. The shared convolutional features of the input image are passed through a convolutional neural network (CNN) to generate a feature map. Fast R-CNN performs RoI pooling on the feature map for each region proposal to extract fixed-size feature vectors. This allows for consistent-sized feature representations regardless of the size or aspect ratio of the region proposals. The extracted RoI features are fed into two sibling fully connected layers: one for object classification and another for bounding box regression. The classification branch predicts the probability of object presence and its class, while the regression branch refines the bounding box coordinates. Fast R-CNN is trained end-to-end using a multi-task loss function that combines classification and bounding box regression losses.

Fast R-CNN efficiently processes the entire image during inference to generate region proposals, extract features, and perform classification and bounding box regression in a single forward pass through the network. By sharing convolutional features and performing RoI pooling, Fast R-CNN significantly reduces computational overhead compared to RCNN. This

improvement in efficiency makes Fast R-CNN more practical for real-time object detection applications while achieving comparable or better accuracy.

2.1.3 FASTER RCNN

Faster R-CNN is a state-of-the-art deep learning-based object detection model introduced by Shaoqing Ren et al. in 2015 [6]. It builds upon the Faster R-CNN architecture, addressing the inefficiencies of its predecessor while maintaining high detection accuracy. Faster R-CNN introduces a Region Proposal Network (RPN) that shares convolutional features with the object detection network. The RPN generates region proposals (bounding boxes) directly from the feature map, eliminating the need for separate algorithms like selective search or edge boxes.

The RPN operates by sliding a small network called an anchor box over the feature map. At each position, the anchor box predicts multiple bounding box proposals and their corresponding objectness scores (the likelihood of containing an object). These anchor boxes have predefined scales and aspect ratios, providing a diverse set of proposals.

Like Fast R-CNN, Faster R-CNN performs RoI pooling to extract fixed-size feature maps for each region proposal from the shared convolutional features. The extracted RoI features are fed into separate branches for object classification and bounding box regression. The classification branch predicts the presence and class of objects within each region proposal, while the regression branch refines the bounding box coordinates. Faster R-CNN is trained end-to-end using a multi-task loss function, combining classification loss, regression loss, and a loss term for the RPN. Faster R-CNN efficiently generates region proposals during inference and performs object classification and bounding box regression in a single forward pass through the network.

Faster R-CNN significantly improves upon the speed and efficiency of its predecessors by integrating the region proposal generation directly into the network architecture. This makes it one of the most widely used and effective object detection models, capable of achieving state-of-the-art performance on various datasets and applications.

2.1.4 CENTERNET

CenterNet is a recent state-of-the-art object detection architecture introduced by Zhou et al. in 2019 [9]. Unlike traditional object detection models, which focus on predicting bounding boxes directly, CenterNet takes a different approach by directly regressing the center points of objects and their corresponding bounding box dimensions and object categories. CenterNet first detects the center points of objects within the input image. It employs a convolutional neural network to predict a heatmap where each peak corresponds to the center of an object. Once the center points are identified, CenterNet regresses the bounding box dimensions (width and height) and orientation (if applicable) around each detected center point. Simultaneously, CenterNet predicts the object category for each detected object center. CenterNet is trained using a combination of losses, including the heatmap loss (to ensure accurate center point detection), the regression loss (to refine bounding box dimensions), and the classification loss (to correctly classify object categories). During inference, CenterNet generates object detections by identifying peaks in the heatmap, regressing bounding box dimensions around these peaks, and assigning object categories.

CenterNet has several advantages over traditional object detection models: CenterNet simplifies the object detection pipeline by directly predicting object centers instead of bounding boxes, reducing the complexity of the model. By focusing on detecting object centers and regressing bounding boxes around them, CenterNet achieves high accuracy in object detection tasks. CenterNet is computationally efficient compared to some other state-of-the-art object detection architectures, making it suitable for real-time applications.

Due to its effectiveness and efficiency, CenterNet has gained popularity in the computer vision community and has been applied to various object detection tasks, including pedestrian detection, vehicle detection, and instance segmentation.

2.1.5 EFFICIENTDET

EfficientDet is a family of state-of-the-art object detection models introduced by Mingxing Tan et al. in 2019 [8]. It is built upon the EfficientNet [12] architecture, which focuses on optimizing both model accuracy and computational efficiency by using a compound scaling

method. EfficientDet employs the EfficientNet architecture as its backbone network. EfficientNet achieves a balance between model size and accuracy by scaling the depth, width, and resolution of the network in a principled manner. EfficientDet introduces a novel feature fusion module called BiFPN (Bidirectional Feature Pyramid Network). BiFPN efficiently integrates features from different scales and enhances information flow in both bottom-up and top-down directions. EfficientDet uses a lightweight object detection head to predict bounding boxes, object categories, and objectness scores. It consists of convolutional layers and a set of prediction heads for each output task. EfficientDet employs compound scaling to optimize both accuracy and efficiency. It scales the depth, width, and resolution of the network simultaneously, achieving better performance than simply scaling one aspect of the network. EfficientDet is trained using standard supervised learning techniques with labeled training data. It utilizes techniques like focal loss and smooth L1 loss to handle class imbalance and regression tasks efficiently. EfficientDet comes in different variants (EfficientDet-D0 to EfficientDet-D7), with varying model sizes and computational costs. Users can choose a model variant based on their specific requirements for accuracy and efficiency.

EfficientDet has several advantages over previous object detection architectures: EfficientDet achieves state-of-the-art performance on various object detection benchmarks while maintaining high efficiency. It offers better computational efficiency compared to other models of similar accuracy, making it suitable for deployment on resource-constrained devices or real-time applications. EfficientDet provides a range of model variants with different trade-offs between accuracy and efficiency, allowing users to choose the most suitable model for their specific needs. EfficientDet has become widely adopted in the computer vision community and has been applied to a wide range of tasks, including object detection, instance segmentation, and human pose estimation.

2.1.6 YOLO

YOLO, which stands for “You Only Look Once”, is a pioneering object detection system introduced by Joseph Redmon et al. [11] YOLO revolutionized object detection by offering real-time performance, achieving impressive speed and accuracy in detecting objects within images and video frames.

- *Single forward pass:* YOLO adopts a single-stage approach, processing the entire image in a single feedforward pass through the neural network. This contrasts with traditional two-stage methods, which involve region proposal and classification separately.
- *Grid-based prediction:* YOLO divides the input image into a grid of cells and predicts bounding boxes and class probabilities for each grid cell. Each grid cell is responsible for predicting objects whose centers fall within that cell.
- *Bounding box prediction:* For each grid cell, YOLO predicts a fixed number of bounding boxes (anchors) along with their corresponding confidence scores and class probabilities. The bounding boxes are represented by their coordinates (x, y, width, height) relative to the grid cell.

Object Confidence and Class Prediction: YOLO predicts a confidence score for each bounding box, indicating the likelihood that the box contains an object. Additionally, class probabilities are predicted for each bounding box, indicating the probability of the object belonging to different predefined classes. After prediction, YOLO applies non-maximum suppression (NMS) to eliminate redundant bounding boxes with overlapping regions, retaining only the most confident detections for each object class.

YOLO achieves high-speed object detection, making it suitable for real-time applications such as video surveillance, autonomous vehicles, and robotics. YOLO's single-stage architecture simplifies the object detection pipeline by combining object localization and classification into a single step. YOLO demonstrates strong generalization across different object categories and environments, making it versatile for various object detection tasks. Since its inception, YOLO has undergone several iterations, with subsequent versions (YOLOv3[16], YOLOv4[14], YOLOv5[18], YOLOv7[15] etc.) introducing improvements in speed, accuracy, and model efficiency. YOLO remains one of computer vision's most popular and influential object detection frameworks.

2.1.7 YOLOv7

YOLOv7, a cutting-edge object detection model, has emerged as a significant breakthrough in the field [15]. Developed by Alexey Bochkovskiy, it builds upon the strengths of its predecessors, YOLOv3 and YOLOv5, while introducing novel techniques to enhance accuracy

and speed. This comprehensive overview delves into the key features, performance, and applications of YOLOv7.

- *Efficient architecture:* YOLOv7 employs a highly efficient architecture comprising a backbone and a neck. The backbone extracts features from the input image, while the neck integrates these features to generate detection predictions.
- *EfficientNet-based backbone:* The backbone of YOLOv7 is based on EfficientNet, a family of neural networks renowned for their high performance-to-parameter ratio [12]. This choice enables YOLOv7 to achieve high accuracy while maintaining a relatively small model size.
- *Enhanced feature integration:* YOLOv7 introduces a novel feature integration mechanism that effectively combines features from different levels of the network [18]. This mechanism enhances the model's ability to capture objects of various sizes and scales.
- *ELAN block:* YOLOv7 incorporates the Enhanced Layer Aggregation Network (ELAN) block, designed to improve feature flow and reduce computational cost [15].
- *Bag of freebies and tricks:* YOLOv7 leverages a "bag of freebies" and "bag of tricks" to further enhance performance without increasing the model's complexity. These techniques include data augmentation, label smoothing, learning rate warmup, mosaic augmentation, copy-paste augmentation, and selective focus [14].
- *Head optimization:* The detection head of YOLOv7 has been optimized for efficiency and accuracy. It uses a combination of anchor boxes and prediction heads to generate detection predictions.

YOLOv7 sets new state-of-the-art results on several object detection benchmarks, including COCO and Pascal VOC. It achieves high accuracy while maintaining a fast inference speed, making it suitable for real-time applications. Moreover, YOLOv7 offers a good balance between accuracy and speed, making it a versatile choice for various object detection tasks.

2.1.8 YOLOv8

YOLOv8 is the latest and most advanced version of the YOLO (You Only Look Once) series, offering substantial improvements in accuracy, speed, and flexibility for tasks such as object

detection, segmentation, and classification [24]. YOLOv8 builds upon its predecessor, YOLOv7, by integrating new architectural innovations and advanced techniques to improve performance and efficiency. YOLOv8 brings a redesigned architecture to improve multi-scale object detection while maintaining real-time performance. The model leverages CSPDarknet [16] (Cross Stage Partial Darknet) as the backbone, which enhances gradient flow and reduces computational redundancy during training. CSPDarknet divides the feature extraction process into stages that reduce the complexity and size of the model while retaining rich feature representations. This design promotes higher inference speed without sacrificing accuracy, making YOLOv8 more efficient than its predecessors. YOLOv8 also incorporates an Enhanced Feature Pyramid Network (FPN), which improves the handling of multi-scale object detection. The FPN ensures that low-level, mid-level, and high-level features are effectively combined, enhancing the model's ability to detect objects of varying sizes and in cluttered environments. One of the significant shifts in YOLOv8 is the move towards an anchor-free object detection mechanism. In contrast to earlier versions (such as YOLOv4 and YOLOv5), which required predefined anchor boxes for bounding box prediction, YOLOv8 eliminates the need for manually tuned anchor boxes. Instead, it directly predicts the center, width, and height of objects. This anchor-free approach simplifies the training process, making the model more adaptable to diverse datasets and reducing the computational overhead associated with anchor box tuning. This advancement aligns YOLOv8 with recent trends in object detection, where anchor-free methods are gaining traction due to their simplicity and flexibility [54][55].

A significant innovation in YOLOv8 is the integration of transformer blocks within the model's architecture. Transformers, originally developed for natural language processing, have proven highly effective in capturing long-range dependencies in image data. In YOLOv8, transformers enhance the model's ability to understand spatial relationships between objects and their surroundings. This addition improves the detection of objects in complex scenes, particularly when objects are occluded or surrounded by background clutter. The transformer components enable YOLOv8 to focus attention on the most relevant regions of an image, enhancing detection accuracy. YOLOv8 adopts a decoupled head architecture, where the tasks of object classification and localization are separated into two distinct branches. This approach allows the model to optimize each task independently, improving both the accuracy of bounding box predictions and the precision of class assignments. By decoupling these tasks,

YOLOv8 addresses the challenge of balancing the competing objectives of classification and localization, leading to superior performance on detection benchmarks. YOLOv8 benefits from advanced training strategies that contribute to its robust performance:

- *Label Assignment Optimization:* This improvement refines how ground-truth labels are assigned to predicted bounding boxes during training. The optimized label assignment enhances the model's ability to match predicted boxes with ground-truth boxes accurately, resulting in more precise detections.
- *Data Augmentation:* YOLOv8 uses advanced augmentation techniques, such as mosaic augmentation and MixUp, to improve generalization. These augmentations simulate a wide range of real-world scenarios, including variations in lighting, scale, and occlusion, helping the model become more robust against challenging conditions.

YOLOv8 incorporates a series of Bag-of-Freebies (BoF) techniques that further improve model accuracy without adding computational complexity during inference. For instance, YOLOv8 uses IoU-aware loss functions to enhance the precision of bounding box regression. Additionally, advanced non-maximum suppression (NMS) methods are employed to remove redundant bounding boxes, ensuring that only the most accurate detections are retained.

YOLOv8 represents a significant advancement in the YOLO series, with improvements in architectural design, training techniques, and multi-task learning. Its anchor-free detection, decoupled head, and transformer integration make it one of the most accurate and efficient object detection models currently available. YOLOv8's versatility extends beyond object detection to tasks like segmentation and classification, maintaining the speed and efficiency that the YOLO family is known for, making it a valuable tool for real-time computer vision applications.

2.2 Multi-view Object detection Methods

Multiview object detection has garnered significant attention in the field of computer vision due to its applicability in various real-world scenarios such as autonomous driving, surveillance systems, and robotics. Leveraging multiple viewpoints provides richer contextual information, enhancing the robustness and accuracy of object detection systems. Data

augmentation techniques play a crucial role in augmenting the available training data to improve the generalization and performance of multiview object detection models. This literature survey aims to provide a comprehensive overview of the existing research on multiview object detection using data augmentation techniques.

Research in multiview object detection has witnessed the development of various techniques leveraging convolutional neural networks (CNNs) and data augmentation methods. For instance, Hou et al. [19] proposed a multiview object detection framework based on a single-stage detector architecture, integrating data augmentation techniques such as rotation and scaling to handle object variability across views. Similarly, Zhang et al. [20] introduced a two-stage detection approach that incorporates viewpoint-aware feature fusion and utilizes geometric transformations for data augmentation, achieving improved performance on multiview datasets. For instance, Zhou et al. [21] introduced a viewpoint-aware framework that incorporates multiple viewpoint-specific detectors and fuses their outputs to achieve robust object detection across views.

In a study cited as [25], researchers explore a multi-class boosting method termed joint boosting. This method effectively reduces computational requirements and sample complexity by identifying common features that can be leveraged across multiple object classes or viewpoints, thus simplifying the training process. Instead of training detectors separately for each class, this approach employs joint training techniques to improve efficiency. In another investigation referenced as [26], the focus is on achieving optimal efficiency in distributed camera networks while maintaining low power consumption and bandwidth requirements. The innovation revolves around a distinctive compression framework tailored for encoding SIFT-based object histograms. This approach capitalizes on three essential properties of multi-view histograms for a 3D object: histogram sparsity, non-negativity, and the shared sparsity observed across various viewpoints. The study investigates the efficacy of a multi-view object detection strategy underpinned by deep learning techniques. In this approach, the detection outcomes from distinct views are merged, thereby augmenting the system's capability to accurately identify objects across diverse perspectives. The experiments were conducted utilizing the VOC 2007 dataset, and three prominent models, namely YOLO, YOLOv2, and SSD, were employed for evaluation. The results demonstrate notable variations in mean

Average Precision (mAP) scores across the models, with SSD exhibiting the highest performance at 64.4%, followed by YOLOv2 at 56.3%, and YOLO at 38.6%. This underscores the effectiveness of the proposed multi-view approach, particularly when coupled with advanced deep learning models, in enhancing object detection accuracy.

The paper referred to as [83] presents a novel approach to enhancing object detection by utilizing multi-view techniques combined with deep learning models such as YOLO, YOLOv2, and SSD. The research focuses on addressing the challenges of detecting small objects, which are often difficult for conventional methods. The study proposes a multi-view framework that integrates views from different perspectives to improve detection accuracy. Experimental results demonstrate that the multi-view versions of these models outperform their classical counterparts in both retrieval capability (measured by Average F-measure) and detection accuracy (measured by mean Average Precision). Specifically, the proposed approach achieves significant improvements in detecting small objects while maintaining faster processing times compared to methods based on region proposals, such as Faster R-CNN. This research highlights the potential of multi-view techniques for improving real-time object detection, especially in applications where small objects are prevalent. The paper referred to as [84] introduces the Deep Dense Face Detector (DDFD), a deep learning-based approach for detecting faces across a wide range of orientations without relying on pose or landmark annotations. Unlike other methods, it simplifies the architecture by eliminating additional components such as bounding-box regression or SVM classifiers.

2.3 Overview of Object Tracking

Object tracking is a critical aspect of computer vision and is used to monitor the movement and location of objects within a series of frames in a video or in real time. The primary goal is to ensure that the system maintains accurate identification and spatial information about objects of interest over time. Object tracking is essential in various applications, including surveillance, autonomous driving [67][68][69], human-computer interaction, augmented reality, and video editing, medical diagnosis systems [70], and robotics [71].

Object tracking methods can be categorized in various ways. For instance, Fiaz et al. [57] conducted a comprehensive study that classifies tracking methods into two main groups: correlation filter-based and non-correlation filter-based methods. Li et al. [58] reviewed and compared deep learning methods for object tracking. Additionally, Verma [59] categorized tracking methods into five types: feature-based, segmentation-based, estimation-based, appearance-based, and learning-based methods. Classification of the object tracking methods is shown in the Figure 2.3:

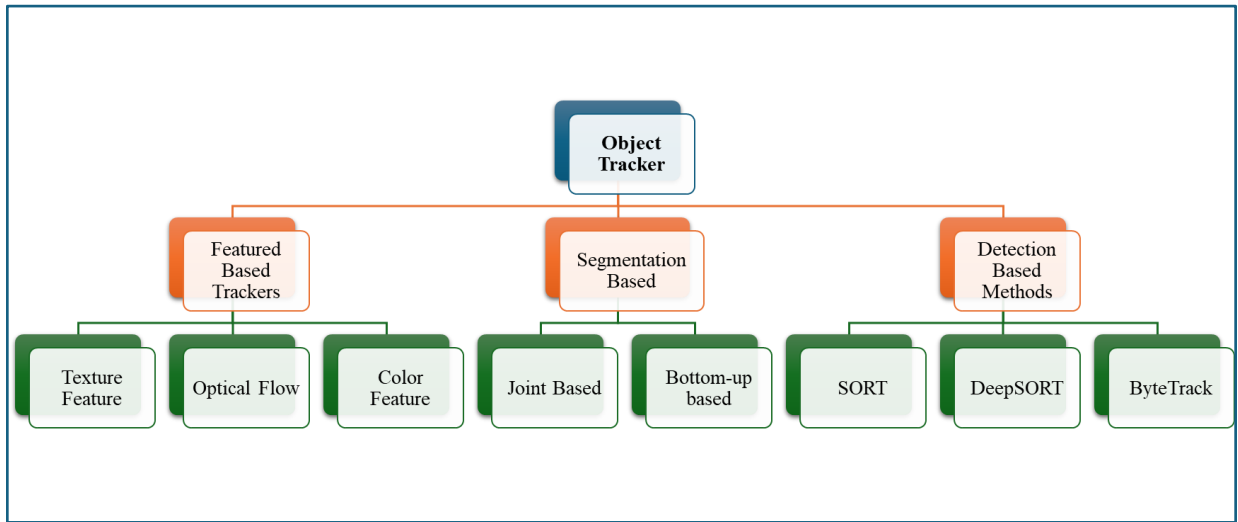


FIGURE 2.3 Methods for Object Tracking

The feature-based method is one of the simpler approaches to object tracking. It begins with extracting features such as color, texture, and optical flow. These features must be distinct to ensure objects can be easily identified in the feature space. The next step involves using these features to find the most similar object in the subsequent frame based on a similarity criterion. One challenge with these methods lies in the extraction step, as the features must be unique, precise, and reliable to effectively distinguish the target object from others. Here are some features commonly used for object tracking [60][61].

Segmenting foreground objects from a video frame is essential and the most crucial step in visual tracking. This process involves separating foreground objects, typically the moving elements in a scene, from the background. To track these objects effectively, they must be isolated from the background. In the Bottom-Up tracking approach, the process is divided into two main tasks: foreground segmentation and object tracking. First, low-level segmentation is used to identify regions in each frame. Then, features are extracted from these foreground

regions, enabling tracking based on these identified features [59][62]. The joint-based method is typically divided into three stages. First, an appearance model is developed using a probabilistic framework. Next, this model is used to perform segmentation. Finally, the tracking process is carried out based on the segmented data [63][64][65].

Detection-based Tracking leverages pre-trained object detectors to find objects and then track them across frames [66]. This is a popular approach using machine learning. Basic Steps of Detection-based trackers are shown below:

- *Object Detection:* The first step is to detect objects in each video frame using object detection algorithms such as YOLO, SSD etc.
- *State Estimation:* The state of each tracked object needs to be estimated, including its position, velocity, size, and other attributes. This is typically done using filtering techniques such as Kalman filters or particle filters, which predict the future state of each object based on its past motion and incorporate new observations to refine the estimates.
- *Data Association:* Data association refers to the process of correctly linking object detections across frames, even in cases where objects may occlude each other, or undergo significant appearance changes.

2.3.1 SORT

Simple Online and Realtime Tracking (SORT) is a widely recognized algorithm designed for real-time multi-object tracking in videos [28]. It is particularly valued for its simplicity, efficiency, and ease of implementation, making it a popular choice in research and practical applications. SORT operates in an online manner, meaning it processes frames sequentially without requiring access to future frames, making it well-suited for real-time applications.

SORT relies on external object detectors, such as Faster R-CNN, YOLO, or SSD, to identify objects in individual video frames. These detectors provide bounding boxes around detected objects, along with confidence scores. SORT employs a Kalman filter to predict the future position of each tracked object based on its previous state. The Kalman filter assumes a linear motion model, which predicts object movement based on position and velocity. This helps in handling slight variations or occlusions between frames. To associate detections with

existing tracks, SORT uses the Hungarian algorithm with a cost function based on the Intersection over Union (IoU) metric. IoU measures the overlap between detected bounding boxes and predicted bounding boxes, ensuring that the most likely matches are paired. SORT maintains and updates tracks using the following strategies:

- *Creation of New Tracks:* If a detection does not match any existing track, a new track is initialized.
- *Track Updates:* Matched tracks are updated with the associated detections, refining their positions using the Kalman filter.
- *Track Termination:* Tracks that remain unmatched for a predefined number of frames are terminated to avoid false positives.

SORT offers a lightweight and efficient solution for online tracking tasks, balancing speed and accuracy in dynamic environments. Its simplicity and compatibility make it an essential tool for researchers and practitioners in the field of computer vision.

2.3.2 DeepSORT

Deep Simple Online and Realtime Tracking (DeepSORT) is an advanced multi-object tracking algorithm that enhances the original SORT by integrating deep appearance features [29]. These features, extracted using a convolutional neural network, enable the algorithm to distinguish objects based on visual similarity, improving robustness in crowded scenes and during occlusions. By combining motion information from a Kalman filter with appearance embeddings, DeepSORT achieves reduced ID switches and more accurate tracking. It is widely used in applications like surveillance, autonomous vehicles, and sports analytics due to its efficiency and reliability.

2.3.3 ByteTrack

ByteTrack is a state-of-the-art multi-object tracking algorithm designed to improve tracking accuracy by leveraging both high and low-confidence detections from an object detector [24]. Unlike conventional methods that discard low-confidence detections, ByteTrack integrates them into the tracking process to handle occlusions and crowded scenes effectively. It uses a

two-stage association strategy: matching high-confidence detections first, followed by low-confidence ones, to maintain object trajectories. ByteTrack is computationally efficient, highly robust, and excels in challenging scenarios, making it ideal for real-time applications like surveillance and autonomous driving.

2.4 Multi-view Object Tracking

Multi-camera object tracking has emerged as a pivotal area in computer vision, enhancing the ability to track objects with greater accuracy and robustness by leveraging multiple viewpoints. This literature review explores key advancements, methodologies, and applications in multi-camera object tracking, highlighting the evolution and current state of the field.

Accurate camera calibration and synchronization are foundational for effective multi-camera tracking. Zhang introduced a flexible new technique for camera calibration, which has been widely adopted due to its robustness and ease of implementation [51]. Methods for synchronizing cameras typically involve time-stamping video frames or using hardware triggers to ensure frames are captured simultaneously across all cameras [49]. Feature-based tracking methods extract distinctive features from objects and match these features across different camera views [51][52]. Alahi et al. proposed using ORB (Oriented FAST and Rotated BRIEF) features for pedestrian tracking across multiple cameras, demonstrating robustness to variations in lighting and perspective. Feature-based methods often rely on descriptors like SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features) to achieve reliable cross-camera matching. Graph-based approaches model the multi-camera tracking problem as a network flow or graph optimization task [43]. Berclaz et al. developed a network flow-based algorithm that represents the tracking task as finding the optimal paths in a graph, with nodes representing detected objects and edges representing possible transitions between frames and views. This method effectively handles occlusions and maintains consistent object identities. Probabilistic methods leverage statistical models to manage uncertainties in tracking [45]. Fleuret et al. introduced a probabilistic occupancy map framework, which estimates the likelihood of an object's presence in various locations within the camera network. Bayesian

networks and particle filters are also commonly used to fuse data from multiple cameras and track objects over time [42][47].

The advent of deep learning has significantly advanced multi-camera object tracking. Ristani and Tomasi proposed a deep learning approach that combines Convolutional Neural Networks for feature extraction with Recurrent Neural Networks (RNNs) for modeling temporal dependencies. This method achieved state-of-the-art performance on several benchmark datasets by effectively capturing complex object appearances and movements [48]. Tracking multiple objects across multiple cameras adds complexity due to occlusions and interactions between objects. Zhang et al. introduced a framework that integrates appearance features and motion patterns to track multiple objects across disjoint camera views. Their approach effectively addresses challenges related to occlusions and re-identification, demonstrating high accuracy in crowded environments [52]. Data association is crucial for maintaining object identities across frames and camera views. Bae and Yoon presented a confidence-based data association method that uses a combination of appearance, motion, and contextual information to associate detected objects across cameras. Their algorithm demonstrated robustness in complex environments with frequent occlusions and interactions. Multi-camera tracking has diverse applications, including surveillance, traffic monitoring, and sports analytics. For instance, Fleuret et al. applied multi-camera tracking for pedestrian monitoring in public spaces, significantly improving detection and tracking accuracy [46]. Wang et al. utilized multi-camera systems for analyzing traffic flow and detecting accidents, demonstrating the practical utility of multi-camera tracking in real-world scenarios [50]. Despite significant advancements, multi-camera object tracking faces ongoing challenges such as handling large-scale data, ensuring real-time performance, and addressing privacy concerns. Future research is expected to focus on developing more efficient algorithms, enhancing scalability, and integrating advanced technologies such as edge computing and the Internet of Things (IoT) for broader real-world applications.

2.5 Evaluation Parameters

Evaluating object detection and tracking systems involves several key parameters to assess their performance effectively. These parameters help determine how accurately and efficiently the systems can detect and track objects across different scenarios and conditions.

2.5.1 Evaluation Parameters for Object Detection

Accuracy metrics for object detection include precision (P), recall (R), F1 score, average precision (AP), and mean average precision (mAP). Intersection over Union (IoU) is utilized for object localization.

Precision (P) is the ratio of true positive detections (TP) to the total number of positive predictions (TP+FP), where FP stands for false positives. Precision measures the accuracy of the detected objects.

$$P = \frac{TP}{TP + FP} \quad (2.1)$$

Recall is the ratio of true positive detections (TP) to the total number of actual positives (TP+FN), where FN stands for false negatives. Recall measures the ability of the detector to find all relevant objects.

$$R = \frac{TP}{TP + FN} \quad (2.2)$$

The F1 score is the harmonic mean of precision and recall, providing a single metric to evaluate the balance between precision and recall.

$$F1 \text{ score} = 2 \times \frac{P \times R}{P + R} \quad (2.3)$$

Average Precision (AP) is the area under the precision-recall curve, calculated as the weighted mean of precisions at each threshold (P_n), with the increase in recall ($R_n - R_{n-1}$) as the weight.

$$AP = \sum_n (R_n - R_{n-1}) P_n \quad (2.4)$$

Mean Average Precision (mAP) is the mean of the average precision values for all N object classes.

$$mAP = \frac{1}{N} \sum_{i=1}^n AP_i \quad (2.5)$$

Intersection over Union (IoU) measures the overlap between the predicted bounding box (A_{pred}) and the ground truth bounding box (A_{gt}).

$$IoU = \frac{A_{pred} \cap A_{gt}}{A_{pred} \cup A_{gt}} \quad (2.6)$$

2.5.2 Evaluation Parameters for Object Tracking

Multi-object tracking metrics[80][81] are metrics used to evaluate the accuracy of tracking algorithms. There are two primary metrics that experts consider while evaluating tracking algorithms: MOTA and MOTP.

The MOTA [31] is perhaps the most widely used metric to evaluate a tracker's performance. MOTA is calculated as:

$$MOTA = 1 - \frac{\sum_t FN_t + FP_t + IDSW_t}{\sum_t GT_t} \quad (2.7)$$

where FN represents the number of false negatives (missed detections), FP represents the number of false positives (false alarms), IDSW represents the number of identity switches, and GT represents the total number of ground truth objects.

The percentage MOTA $(-\infty, 100]$. MOTA can also be negative in cases where the number of errors made by the tracker exceeds the number of all objects in the scene.

MOTP measures the localization accuracy of the tracked objects. It calculates the average distance between the predicted positions of the tracked objects and their corresponding ground truth positions. MOTP is calculated as:

$$MOTP = \frac{\sum_{t=1}^T \sum_{i=1}^{N_t} d(i)}{\sum_{t=1}^T N_t} \quad (2.6)$$

Where T is total number of frames, N_t is the number of tracked objects in frame t , $d(i)$ is the Euclidean distance between the predicted position and the ground truth position of the i th tracked object in frame t .

Evaluating object detection and tracking systems requires a combination of accuracy, localization, robustness, trajectory, and efficiency metrics. These parameters provide a comprehensive view of the system's performance and help identify areas for improvement[82].

CHAPTER – 3

Enhancing Multiview Object Detection through Image Data Augmentation and Deep Neural Networks

3.1 Introduction

In the rapidly evolving field of computer vision, object detection remains a cornerstone task with applications ranging from autonomous driving to surveillance systems. The advent of deep neural networks (DNNs) has significantly advanced the accuracy and efficiency of object detection systems, yet challenges persist, particularly in the context of multiview object detection. Multiview object detection involves recognizing and localizing objects from multiple viewpoints, a task that inherently demands robust models capable of handling diverse perspectives and occlusions.

Traditional methods of object detection often struggle with the variability and complexity presented by multiple viewpoints. These limitations underscore the need for innovative approaches that can effectively generalize across different angles and conditions. One promising avenue to address these challenges is through image data augmentation — a technique that artificially expands the dataset by creating modified versions of images. Data augmentation has been shown to improve model robustness and generalization, particularly in scenarios where acquiring a large volume of diverse training data is impractical. This chapter explores the intersection of image data augmentation and deep neural networks to enhance multiview object detection. By leveraging advanced augmentation techniques, this research aims to enrich the training datasets, thereby enabling DNNs to better learn and generalize from various perspectives. Furthermore, this work investigates the integration of these augmented datasets with state-of-the-art neural network architectures to evaluate their performance improvements in multiview object detection tasks. The primary objectives in this chapter are threefold: first, to develop effective image data augmentation strategies tailored for multiview object detection; second, to integrate these augmented datasets with deep neural networks; and third, to assess the impact of these combined methods on the accuracy and robustness of multiview object detection systems. Through a comprehensive experimental framework, this

thesis seeks to contribute to the broader understanding of how data augmentation and neural network design can synergistically improve multiview object detection.

By addressing the challenges associated with multiview object detection, this research not only aims to enhance the performance of current detection systems but also provides insights into the broader implications of data augmentation and deep learning in computer vision. The findings of this study have the potential to inform future research and development, paving the way for more advanced and reliable object detection technologies in diverse application domains.

3.2 Image Data Augmentation

Image data augmentation is a critical technique in computer vision used to artificially expand the size and diversity of a training dataset without the need to collect additional data. This process involves creating modified versions of existing images, which helps improve the robustness and generalization of machine learning models, particularly deep neural networks. Data augmentation is especially beneficial in scenarios where acquiring a large and varied dataset is impractical or costly. Here, we explore the key methods, benefits, and applications of image data augmentation, with a particular focus on its role in enhancing multiview object detection.

3.2.1 Methods of Image Data Augmentation

Image data augmentation can be categorized into three types: geometric transformations, and photometric transformations.

- **Geometric Transformation:** Geometric transformations change the spatial structure of an image, altering the arrangement of pixels without modifying their color or intensity. These transformations are especially useful in making models invariant to changes in viewpoint, rotation, scale, or position. Let $x \in R^{H \times W \times C}$ where H , W and C are the height, width, and number of color channels, respectively.
 - *Rotation:* Rotate the image by a certain angle θ . Rotating images helps the model recognize objects in different orientations and is common for datasets where the object's angle is not fixed.

$$x' = Rotate(x, \theta) = x \times \theta \quad (3.1)$$

Where θ is the rotation angle, and R is the rotation transformation matrix.

- *Scaling*: Resize the image by scaling it along the x and y dimensions.

$$Scale(I, S_x, S_y) = resize(I, S_x * width, S_y * height) \quad (3.2)$$

where, S_x and S_y scaling factors along the x and y dimensions, respectively. This transformation makes the model robust to objects of different sizes and distances.

- *Translation*: Shift the image horizontally and/or vertically by a certain number of pixels. This helps the model generalise objects that may appear in different parts of an image.

$$Translate(I, \Delta x, \Delta y) = I(x - \Delta x, y - \Delta y) \quad (3.3)$$

Where Δx and Δy are the horizontal and vertical translation distances, respectively.

- *Flipping*: Flip the image horizontally and/or vertically. Horizontal flipping is commonly used for images where objects are symmetrical along the vertical axis, such as faces or animals.

$$Flip(I) = I(:, end:-1:1) \text{ (for horizontal flip)} \quad (3.4)$$

$$Flip(I) = I(end:-1:1, :) \text{ (for vertical flip)} \quad (3.5)$$

- **Photometric Transformations**: Photometric transformations modify the image's pixel values to change its color, brightness, contrast, or other photometric properties without altering the spatial layout. These transformations help models generalize to different lighting conditions, exposure levels, and camera characteristics.

- *Brightness Adjustment*: Changing the brightness levels to simulate different lighting conditions.

$$x' = x + \Delta b \quad (3.6)$$

where Δb is the brightness adjustment factor. This makes the model robust to changes in illumination.

- *Contrast Adjustment*: Modifying the contrast to emphasize or de-emphasize features.

$$x' = \alpha x + \beta \quad (3.7)$$

where α is the contrast adjustment factor, and β is a bias term. Increasing contrast makes shadows and highlights more prominent, while decreasing it flattens the tones, simulating overexposed or underexposed scenes.

- *Color Jittering*: Randomly changing the color properties, such as hue, saturation, and brightness.

$$x' = J(x, \delta_h, \delta_s, \delta_v) \quad (3.8)$$

where δ_h , δ_s , and δ_v are adjustments to hue, saturation, and brightness, respectively, and J is the color jittering function. This transformation simulates natural variations in lighting and color tones, increasing the robustness of the model.

- *Gaussian Noise*: Adding random noise to simulate sensor imperfections and environmental conditions.

$$x' = x + \mathcal{N}(0, \sigma^2) \quad (3.9)$$

Where $\mathcal{N}(0, \sigma^2)$ is Gaussian noise with mean 0 and variance σ^2 . Noise addition helps the model handle image imperfections, such as those from low-light conditions or poor-quality sensors.

3.2.2 Benefits of Image Data Augmentation

Data augmentation provides several key benefits for training machine learning models, especially in computer vision and other fields where labeled data can be limited or expensive to obtain. The primary benefits of image data augmentation are as below:

- *Improved Generalization*: Data augmentation exposes the model to a wider variety of scenarios, reducing overfitting and improving the model's ability to generalize to unseen data.

- *Enhanced Robustness:* By training on augmented data, models become more resilient to variations in viewpoint, lighting, occlusions, and other real-world conditions.
- *Reduced Data Collection Costs:* Augmentation allows for the creation of a larger and more diverse training set without the need for extensive and expensive data collection efforts.
- *Mitigation of Class Imbalance:* Augmentation techniques can be used to balance the distribution of classes in the training dataset, addressing issues related to underrepresented classes.
- *Facilitates Training of Deep Models:* Deep neural networks often require large amounts of data to perform well. Augmentation helps meet this requirement by generating additional training samples.

3.2.3 Applications in Multiview Object Detection

In the context of multiview object detection, image data augmentation is particularly valuable. Multiview detection involves recognizing and localizing objects from different angles and perspectives, making it crucial for models to handle a wide range of variations. Here are specific applications:

- *Different Viewpoints:* Geometric transformations like rotation and translation help simulate different camera angles and positions, enabling the model to learn from multiple perspectives.
- *Handling Occlusions and Variability:* Techniques such as cutout and Mixup help the model learn to detect objects even when parts are occluded or when they appear in varied contexts.
- *Enhancing Real-World Performance:* Augmented datasets better represent the diversity encountered in real-world scenarios, improving the model's performance in practical applications such as autonomous driving and surveillance.
- *Training with Limited Data:* When only a small dataset is available, augmentation significantly boosts the model's training process by providing varied and enriched samples.

Image data augmentation is a powerful tool in the enhancement of multiview object detection. By expanding and diversifying the training dataset, it helps create more robust and

generalizable models. In combination with deep neural networks, data augmentation techniques contribute to the advancement of object detection technologies, enabling more accurate and reliable performance across a wide range of applications. This research aims to explore and leverage these techniques to push the boundaries of what is possible in multiview object detection, addressing existing challenges and paving the way for future innovations.

3.3 Proposed Approach

Figure 3.1 illustrates a detailed workflow for improving object detection using image data augmentation and deep neural networks. The process starts with a dataset of images, specifically from the Open Image Dataset v6 [36]. Various data augmentation techniques are applied to these images to enhance the diversity and robustness of the training dataset.

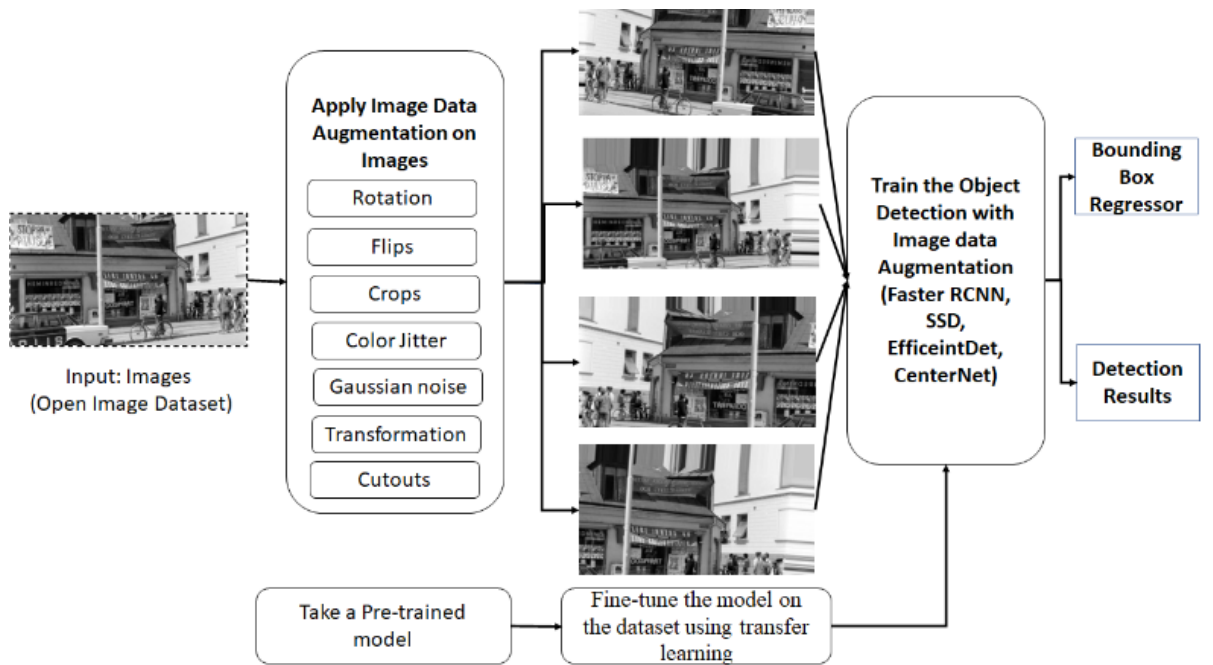


FIGURE 3.1 Architecture of multiview object detection using image data augmentation and deep neural networks

Next, object detection models are trained with and without augmented images. Pre-trained models, such as Faster R-CNN, SSD, EfficientDet, and CenterNet, are used as the initial starting point. These models have been previously trained on the COCO dataset and have already learned useful feature representations. The pre-trained models are then fine-tuned on

the augmented dataset, allowing the models to adapt their learned features to the specific characteristics of the new dataset and improving their performance.

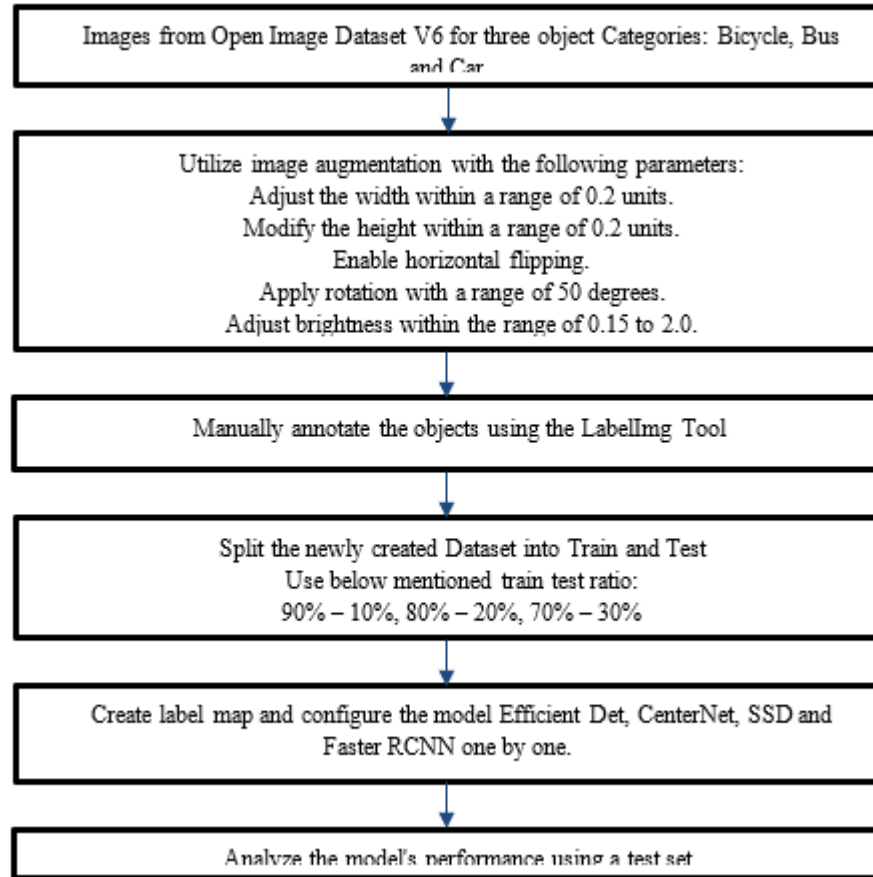


FIGURE 3.2 Detail steps of Multiview Object Detection using Image Data Augmentation

Figure 3.2 illustrates the comprehensive steps for multi-view object detection utilizing image data augmentation. Initially, images of three object categories—Bicycle, Bus, and Car—are sourced from the Open Image Dataset. These images undergo data augmentation to produce new images with different viewpoints of the objects. Next, the objects are annotated using the LabelImg tool for the categories mentioned above. Figure 3.3 presents example annotations in XML format for an image. The figure also displays the filename, file path, image dimensions, and the coordinates of the objects and their bounding boxes.

```
▼<annotation>
  <folder>open_images_volume</folder>
  <filename>00d17e785bbf2ca1.jpg</filename>
  <path>/mnt/open_images_volume/00d17e785bbf2ca1.jpg</path>
  ▼<source>
    <database>Unknown</database>
  </source>
  ▼<size>
    <width>1024</width>
    <height>768</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  ▼<object>
    <name>Bus</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    ▼<bndbox>
      <xmin>0</xmin>
      <ymin>179</ymin>
      <xmax>297</xmax>
      <ymin>574</ymin>
    </bndbox>
  </object>
  ▼<object>
    <name>Bus</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    ▼<bndbox>
      <xmin>175</xmin>
      <ymin>72</ymin>
      <xmax>796</xmax>
      <ymin>662</ymin>
    </bndbox>
  </object>
</annotation>
```

FIGURE 3.3 Sample annotation file in XML format

Figure 3.4 displays sample output images produced through image data augmentation, demonstrating how multiple variations can be created from a single image, highlighting the versatility and advantages of this technique.



FIGURE 3.4 Sample images after applying Image Data Augmentation

3.4 Experiment and Results

The performance of the object detection models is evaluated with and without using Image Data Augmentation. Images are sourced from Open Image Dataset V6.

3.4.1 Implementation Detail

All modules were coded in Python version 3.10.12. Deep learning models were developed with the help of Tensorflow (version 2.17.0) and Cuda(version 12.1). The Faster RCNN, SSD, EfficientDet D1 and CenterNet network underwent training with image dimensions set at 640×640 , 640×640 , 640×640 and 512×512 respectively across 10000 steps, utilizing a mini-batch size of 8 images. Additionally, the model's weights were set using the COCO pre-trained model. The backbone for Faster RCNN and CenterNet Resnet 101 [95], the backbone for EfficientDet D1 is EfficientNet and the backbone for SSD is Mobile net [96].

3.4.2 Dataset

Open Images Dataset is a large-scale visual dataset containing millions of images with extensive annotations. It is one of the most comprehensive resources available for computer vision research. It provides diverse and high-quality data for tasks such as object detection, image classification, and instance segmentation.

The deep-learning models are trained using the Open Images Dataset, focusing on three object categories: bus, bicycle, and car. Various training and testing ratios are employed specifically 90%-10%, 80%-20%, and 70%-30%. A total of 2,000 images are used to train the models without image data augmentation, while 2,500 images are used with image data augmentation.

3.4.3 Experimental Results

Table 3.1 presents the mean average precision (mAP) of various object detection models—CenterNet, Efficient Det D1, Faster RCNN, and SSD—evaluated at Intersection over Union (IoU) thresholds of 0.50 and 0.75, without applying Image Data Augmentation. The results are reported for different train-test split ratios of 90%-10%, 80%-20%, and 70%-30%. Table 3.2, on the other hand, displays the mAP scores for the same models, but this time with Image Data Augmentation applied, using the same train-test split ratios. Additionally, Figures 3.4, 3.5, and 3.6 illustrate the performance comparison between the object detection models with and without the use of image data augmentation at an IoU threshold of 0.50.

TABLE 3.1 Performance of object detection models without Image Data Augmentation on Open Image Vehicle Dataset V6

Model	CenterNet	Efficient Det D1	Faster RCNN	SSD
Backbone	Resnet 101	EfficientNet	Resnet 101	Mobile net
Train – Test Ratio: 90% 10%				
mAP@IoU=0.50	68.4	71.7	65.6	64.8
mAP@IoU=0.75	50	51.4	45.1	45.5
Train – Test Ratio: 80% 20%				
mAP@IoU=0.50	63.4	67.9	59.1	59.8
mAP@IoU=0.75	48.4	46.9	42.7	43.3
Train – Test Ratio: 70% 30%				
mAP@IoU=0.50	63.2	67.5	58.5	59.7
mAP@IoU=0.75	46.9	46.6	41.9	42.1

TABLE 3.2 Performance of object detection models with Image Data Augmentation on Open Image Vehicle Dataset V6

Model	CenterNet	Efficient Det D1	Faster RCNN	SSD
Backbone	Resnet 101	EfficientNet	Resnet 101	Mobile net
Train – Test Ratio: 90% 10%				
mAP@IoU=0.50	72.7	73.5	68.3	67
mAP@IoU=0.75	52.7	53.4	46.4	45.2
Train – Test Ratio: 80% 20%				
mAP@IoU=0.50	68.1	67.7	60.1	61.7
mAP@IoU=0.75	49.3	48.2	43	44.8
Train – Test Ratio: 70% 30%				
mAP@IoU=0.50	71.6	70.4	62.8	63.4
mAP@IoU=0.75	50.9	49.8	47.5	47.5

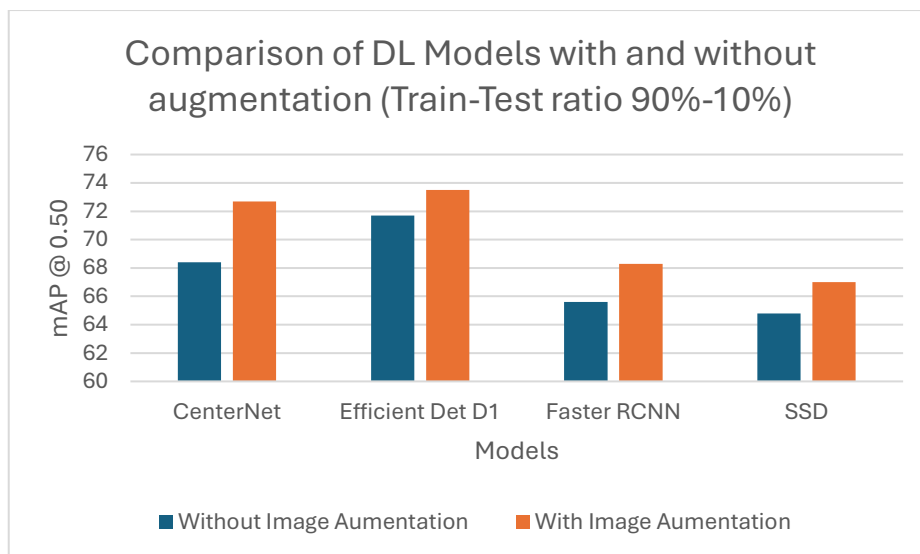


FIGURE 3.5 Comparison of Deep learning models with and without augmentation at Train-test ratio 90%-10%

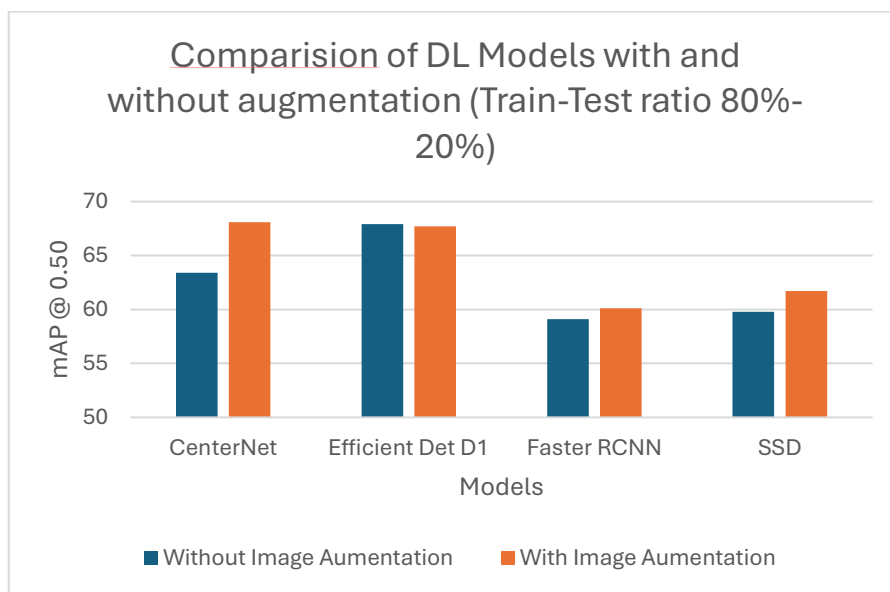


FIGURE 3.6 Comparison of Deep learning models with and without augmentation at Train-test ratio 80%-20%

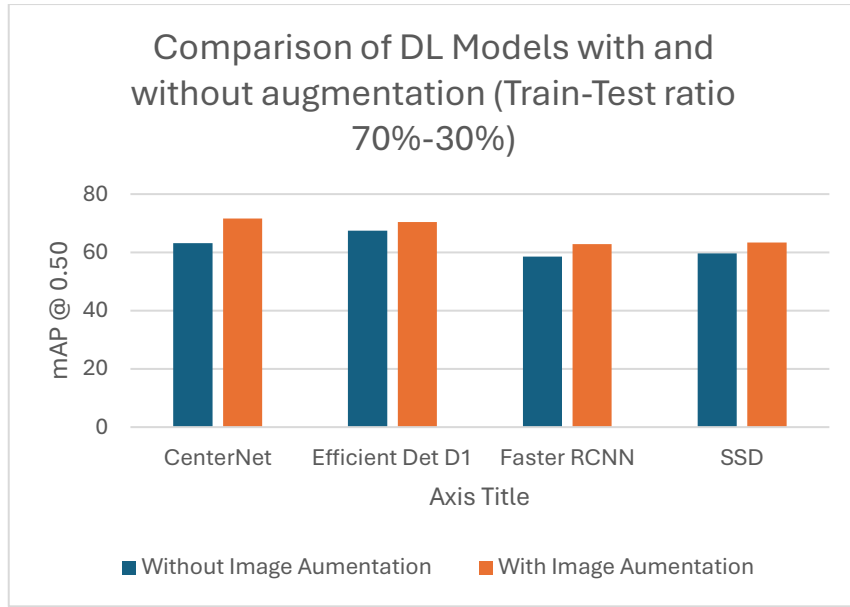


FIGURE 3.7 Comparison of Deep learning models with and without augmentation at Train-test ratio 70%-30%

Figures 3.8 and 3.9 present sample outputs of multiview object detection applied to the Open Image Dataset. These figures display the bounding box coordinates and confidence scores for various object categories.

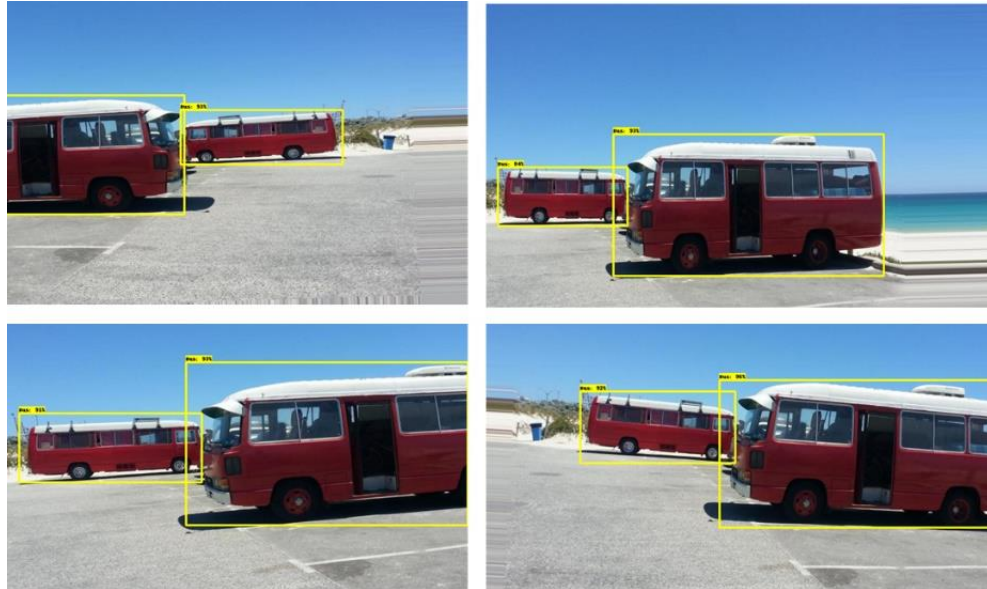


FIGURE 3.8 Sample Output Images of Multiview Object Detection on Open Image Dataset.



FIGURE 3.9 Sample Output Images of Multiview Object Detection on Open Image Dataset.

3.5 Application Of multi-view object detection in Autonomous Driving using Deep Learning Approach

Autonomous vehicles rely heavily on their ability to perceive their surroundings in order to maintain safe and effective driving. Object detection plays a key role in this system by enabling the vehicle to identify and localize essential objects, including pedestrians, other vehicles, traffic signs, and more. In real-time applications, deep learning-based object detectors are crucial for accurately identifying these elements.

The field of autonomous driving is evolving rapidly, with the potential to enhance road safety, increase operational efficiency, and provide greater convenience for users. A central aspect of this innovation is the vehicle's capability to perceive and understand its environment. To overcome limitations inherent in single-view systems, such as narrow field of vision and challenges with occlusion, multi-view object detection offers an enhanced solution by combining data from several perspectives. This approach provides a more comprehensive understanding of the vehicle's surroundings.

Deep learning methods have shown exceptional performance in this area, with models like YOLO (You Only Look Once) setting the standard for real-time object detection. The most recent version, YOLOv8, brings notable advancements in both detection speed and accuracy, positioning it as a prime candidate for autonomous driving technologies. This paper examines the application of YOLOv8 in multi-view object detection, leveraging the Udacity self-driving car dataset for training and validation purposes.

The Udacity self-driving car dataset is a valuable resource, containing images captured from various camera viewpoints to simulate the conditions a vehicle would encounter on the road. By using this dataset, YOLOv8 can be trained to detect a broad array of objects from different angles, including pedestrians, vehicles, and traffic signs. The multi-view approach not only enhances the accuracy of detection but also boosts the vehicle's ability to respond to a wide range of dynamic scenarios.

This study focuses on the integration of YOLOv8 with the Udacity self-driving car dataset, showcasing how this combination can improve current autonomous vehicle perception systems. The paper covers the methodology, implementation details, and the outcomes of the experiments, as well as the potential benefits and challenges associated with deploying multi-view object detection in practical autonomous driving situations.

The Udacity self-driving car dataset, hosted on Roboflow [98], is an extensive collection of images designed to aid the development and evaluation of autonomous driving technologies. This dataset includes a large number of front-facing camera images captured in a variety of driving conditions and environments, providing a diverse and reliable foundation for training object detection models. The dataset contains a total of 3,000 images, which are divided into training (80%), testing (10%), and validation (10%) sets.

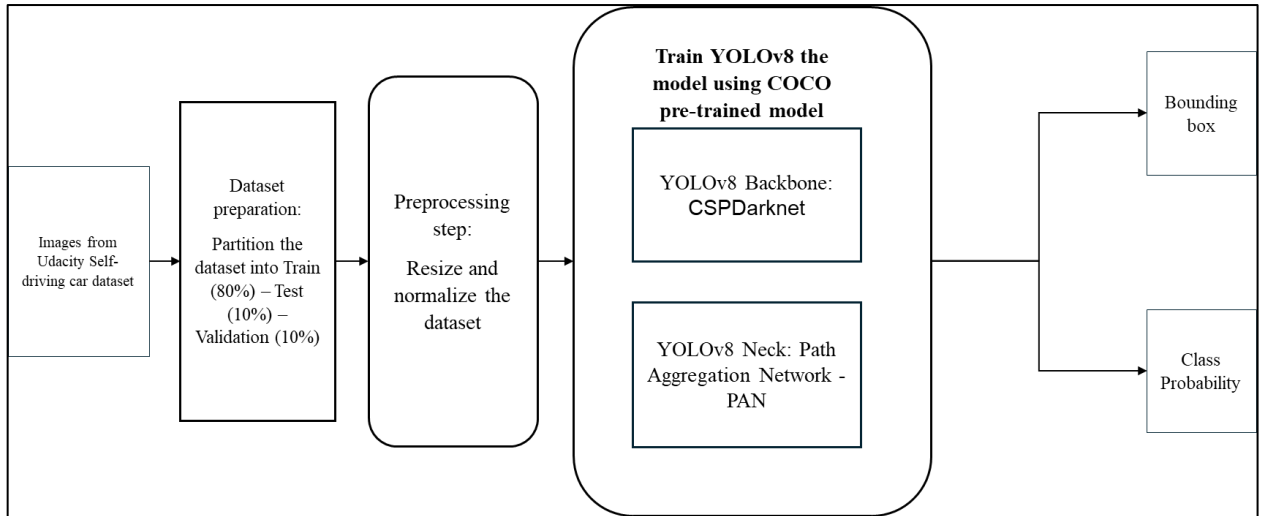


FIGURE 3.10 Steps applied for object detection using YOLOv8 Object Detector

Within the dataset, there are annotations for 11 distinct object classes, which are representative of common elements encountered during driving. These classes include 'biker', 'car', 'pedestrian', and various traffic light states such as 'trafficLight', 'trafficLight-Green', 'trafficLight-GreenLeft', 'trafficLight-Red', 'trafficLight-RedLeft', 'trafficLight-Yellow', 'trafficLight-YellowLeft', and 'truck'. The inclusion of these diverse object types enables the model to improve its detection and classification capabilities across a broad range of real-world driving scenarios.

For the training process, Python 3.10.12 and the PyTorch framework (version torch-2.2.1+cu121) were employed, running on a Tesla T4 GPU with 15,102 MiB of memory. The YOLOv8 model was trained on images with a resolution of 640×640 pixels over the course of 100 epochs, using a pre-trained model that had been initially trained on the COCO dataset. The YOLOv8 architecture consists of 168 layers and includes a total of 3,012,993 parameters and 3,012,977 gradients. Its computational performance reaches 8.2 Giga Floating-Point Operations per Second (GFLOPs), which ensures efficient processing of large datasets for real-time object detection in autonomous driving systems.

TABLE 3.3 Performance of object detection modes for vehicle detection.

Model	Dataset	mAP (Mean Average Precision)
Deterministic RetinaNet (Baseline) [99]	KITTI	37.11%
Output Redundancy [99]	KITTI	34.99%
Our approach - YOLOv8	Udacity Self-driving car dataset	46%

In this study, we assessed the performance of our approach using the YOLOv8 model on the Udacity Self-driving Car dataset, comparing it with other established methods, including the deterministic RetinaNet (Baseline) and the Output Redundancy technique, both evaluated on the KITTI dataset [97]. The performance of these methods was measured using Mean Average Precision (mAP), a widely recognized metric for evaluating object detection models.

The deterministic RetinaNet baseline, when tested on the KITTI dataset, achieved an mAP score of 37.11%. This score served as the benchmark for comparing the performance of the other methods. The Output Redundancy method, also evaluated on the KITTI dataset, achieved a slightly lower mAP of 34.99%. This indicates that while the Output Redundancy method may offer some advantages in specific situations, it did not surpass the baseline performance in this particular test.

In contrast, our proposed method, which leverages the YOLOv8 model applied to the Udacity Self-driving Car dataset, achieved a notably higher mAP score of 46.00%. This represents a significant improvement over both the baseline and the Output Redundancy method, demonstrating the effectiveness of our approach in the context of autonomous vehicle perception. These results highlight the potential of the YOLOv8 model in enhancing object detection accuracy for real-world driving environments.

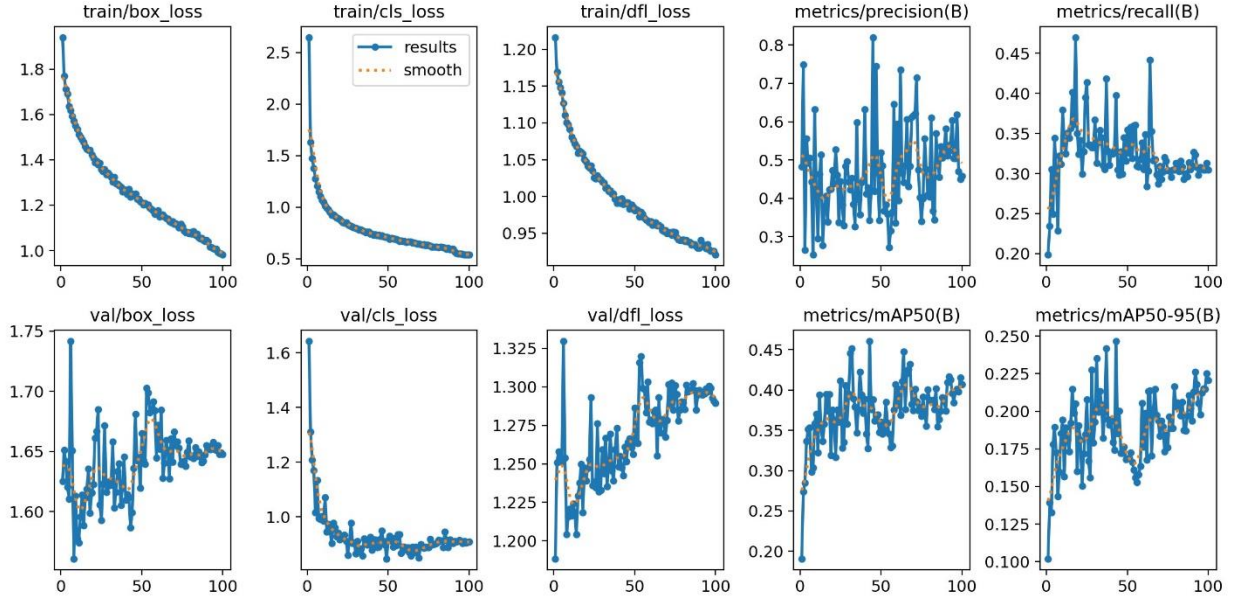


FIGURE 3.11 The convergence of both training and validation losses for the YOLOv8 algorithm object detector and classification is observed at 100 epochs



FIGURE 3.12 Sample Output (Udacity Car Dataset)

3.6 Conclusion and discussion

In this chapter, we have explored the critical role of image data augmentation in enhancing multiview object detection using deep neural networks. By applying a variety of augmentation techniques, including geometric and photometric transformations, we demonstrated how these methods expand dataset diversity, improve model robustness, and address challenges associated with multiview object detection, such as varying viewpoints and occlusions.

The proposed approach effectively integrates augmented datasets with state-of-the-art object detection models like Faster R-CNN, SSD, EfficientDet, and CenterNet. The

experimental results, conducted using the Open Images Dataset, confirmed that models trained with augmented data consistently outperformed those trained without it, achieving higher mean average precision (mAP) across various training-test ratios and IoU thresholds. This improvement highlights the impact of data augmentation on both the accuracy and reliability of detection systems. Additionally, the chapter introduced the application of YOLOv8 in autonomous driving scenarios using the Udacity Self-driving Car Dataset. The study illustrated how multiview object detection enhances perception in real-world conditions, overcoming the limitations of single-view systems. The significant mAP improvements achieved by YOLOv8 further underscore the potential of combining advanced neural networks with robust training datasets. In summary, this chapter demonstrates that leveraging image data augmentation alongside modern deep learning models is a powerful strategy for improving multiview object detection. The findings pave the way for further research into innovative augmentation methods and their integration with advanced architectures, contributing to the development of reliable, high-performing object detection systems in diverse domains.

CHAPTER – 4

Multi-camera Object Detection and Tracking: A YOLOv7 and DeepSORT-Based Approach

4.1 Overview

Multi-camera object tracking is a critical area of research in computer vision, aimed at maintaining consistent trajectories of objects as they move across the fields of view of multiple cameras. This technology has become increasingly important in applications such as video surveillance, autonomous driving, sports analytics, and smart cities [72][73]. By leveraging the complementary perspectives provided by multiple cameras, multi-camera object tracking systems address key challenges of single-camera tracking, such as occlusions, limited fields of view, and re-identification (Re-ID) difficulties [74][75][76].

In recent years, significant progress has been made in object detection, primarily driven by the development of convolutional neural networks. Among these, the YOLO family of models has gained widespread popularity due to its efficiency and accuracy in detecting objects in real-time. The latest iteration of this model, YOLOv7 [15], introduces several improvements in speed and accuracy over its predecessors, making it a promising candidate for multi-camera applications. YOLOv7's ability to balance precision and computational efficiency makes it well-suited for real-time object detection, even in complex environments with multiple overlapping objects and varying scales.

While YOLOv7 provides a strong foundation for object detection, tracking these objects across multiple camera feeds presents additional challenges. Traditional tracking algorithms often rely on heuristics or simplistic models, which may fail in dynamic and cluttered environments. To address these challenges, DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric) has emerged as a robust tracking algorithm that combines deep learning-based feature extraction with the Kalman filter and the Hungarian algorithm for data association [29]. DeepSORT's integration of appearance features with motion information allows for more accurate and reliable tracking, even when objects move between different camera views or are temporarily occluded.

This chapter presents a novel approach to multi-camera object detection and tracking by integrating YOLOv7 for object detection with DeepSORT for tracking. The proposed system aims to leverage the strengths of both YOLOv7 and DeepSORT, addressing the challenges of real-time processing, occlusions, and cross-camera tracking. By combining these two state-of-the-art methods, the system is designed to provide a scalable and efficient solution for monitoring large areas with multiple cameras, ensuring that objects are accurately detected and tracked across different views [77].

This study presents three primary contributions aimed at enhancing object detection and tracking in multi-camera surveillance systems. Firstly, it demonstrates the practical utility and high performance of the YOLOv7 object detection model when deployed in a multi-camera setup. The experiments highlight YOLOv7's capability to deliver accurate detections while operating in real time, which is essential for applications requiring low-latency and high-throughput processing. The results confirm that YOLOv7 is not only effective in single-camera environments but also adapts well to the complexities introduced by multi-camera systems, such as overlapping fields of view and varying lighting conditions. Secondly, the work introduces the integration of YOLOv7 with DeepSORT, a popular multiple object tracking algorithm. This integration combines the strengths of both models—YOLOv7's high-precision object detection and DeepSORT's ability to track identities over time using a combination of appearance descriptors and motion cues. By fusing these features, the tracking component becomes more robust against occlusions, re-identification issues, and abrupt object movements, leading to improved tracking continuity and accuracy across different camera perspectives. Overall, the proposed approach offers a significant improvement in multi-camera object detection and tracking [92].

4.2 Proposed Approach

The proposed approach for multi-camera object tracking focuses on the seamless integration of object detection, feature extraction, and cross-camera association to achieve robust and accurate tracking across multiple camera views. This approach addresses challenges such as varying camera perspectives, occlusions, and differences in lighting conditions by leveraging

advanced detection and tracking algorithms alongside sophisticated data association techniques.

4.2.1 YOLOv7

YOLOv7 represents a significant advancement in real-time object detection. Building on the success of its predecessors, YOLOv7 introduces several architectural innovations and optimizations that enhance both the speed and accuracy of object detection tasks, making it one of the most efficient models available for various computer vision applications. YOLOv7 retains the core philosophy of the YOLO family, which is to perform object detection as a single-stage process, allowing for rapid inference times. However, YOLOv7 introduces key modifications and enhancements that set it apart from earlier versions:

Extended Efficient Layer Aggregation Networks (E-ELAN): YOLOv7 incorporates an E-ELAN, which improves feature fusion and the model's capacity to learn complex patterns. E-ELAN builds on the original ELAN architecture by extending the depth and introducing better layer aggregation techniques. This results in a more expressive feature representation, particularly beneficial for detecting small objects or objects in cluttered scenes.

Dynamic anchor boxes: YOLOv7 employs dynamic anchor boxes that adjust during training, which helps the model better adapt to the scale and aspect ratio of objects in the dataset. This dynamic adjustment enhances the model's ability to detect objects of varying sizes and shapes more accurately, particularly in complex scenes where traditional static anchor boxes might struggle.

In summary, YOLOv7 represents a significant advancement in object detection, offering a compelling balance of speed and accuracy. Its innovative architecture and training techniques make it a powerful tool for various computer vision applications.

4.2.2 Object tracking using YOLOv7 with DeepSORT

In multi-camera surveillance systems, accurate object detection and tracking across various viewpoints are crucial for comprehensive monitoring. This study leverages YOLOv7, a state-of-the-art object detection model, in conjunction with DeepSORT, a robust tracking algorithm, to address these challenges. The Simple Online and Realtime Tracking (SORT) algorithm is a

widely used tracking framework that provides a simple and efficient solution for multi-object tracking (MOT) in real-time applications. SORT uses a combination of the Kalman filter for motion prediction and the Hungarian algorithm for data association to track objects between frames [28]. While highly efficient, SORT relies solely on spatial information (e.g., position and motion) for object association, which makes it susceptible to failures in scenarios involving occlusions and identity switches. To address these limitations, DeepSORT was introduced as an enhancement to SORT. DeepSORT augments SORT with a deep learning-based appearance feature extraction mechanism, enabling more robust object association through a combination of motion and appearance cues. This enhancement significantly improves the ability to maintain consistent object identities, even in challenging conditions such as partial occlusions. Using a multi-view multi-camera dataset, the research aims to demonstrate the effectiveness of this combined approach in achieving high accuracy and reliability in detecting and tracking objects. Integrating YOLOv7 and DeepSORT is expected to provide a powerful tool for enhancing surveillance capabilities in complex environments. Figure 4.1 provides a comprehensive overview of a multi-camera object tracking pipeline, specifically employing a DeepSORT-based framework for robust tracking. It can be divided into two major stages: Object Detection and Multi-Object Tracking.

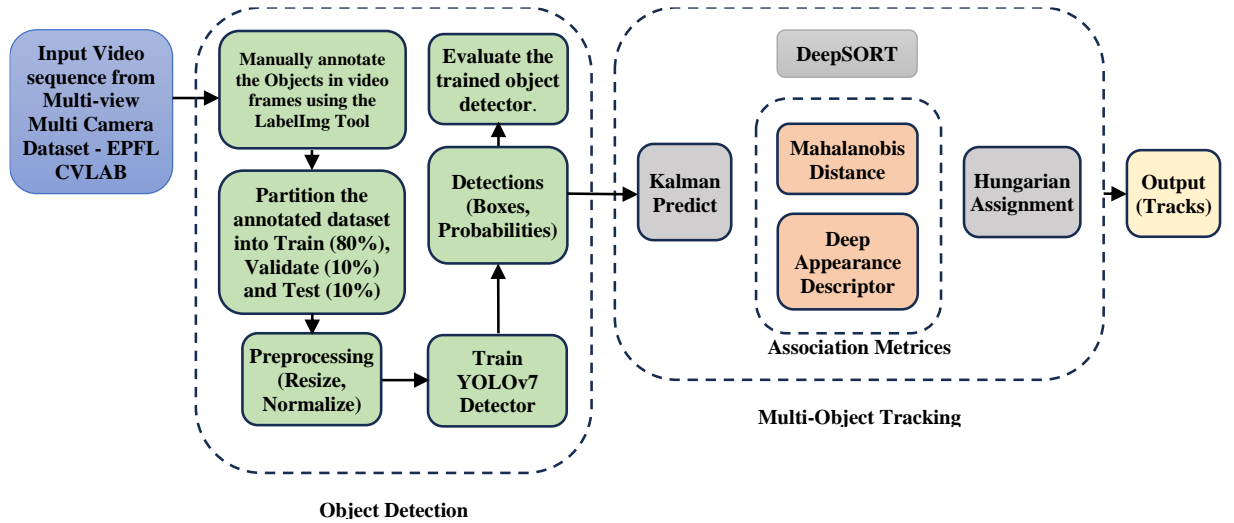


FIGURE 4.1 Proposed Model for object detection and tracking based on YOLOv7 and DeepSORT

We adopt Object detection and tracking using a two-step process: In the first step, we do the detection and localization of the object using the YOLOv7 object detector. In the second step, using a motion predictor we predict the future motion of the object using its past information using DeepSORT.

First, we manually annotate objects in the video frames using the LabelImgTool in the YOLOv7 format. The dataset is subsequently split into training, validation, and testing subsets. During preprocessing, the images are resized according to the YOLOv7 specifications. Following this, we train the YOLOv7 model. The newly trained YOLOv7 object detector is then used as an input for the DeepSort algorithm.

Deep SORT extends the original SORT algorithm. It enhances the basic SORT by integrating a deep learning feature extractor for improved measurement of appearance similarity between objects across frames. This results in better handling of long occlusions and interactions between objects, which are common challenges in multi-object tracking scenarios. Steps involved in the Deep SORT algorithm:

Step: 1 Detection

Before tracking can begin, objects in each frame must be detected. Use the YOLOv7 detector trained on the Multi-view Multi-class Detection dataset CVLAB – EPFL [41]. Each detection is represented as a bounding box $d_i=[x, y, w, h]$, where x and y are the coordinates of the center of the box, and w and h are the width and height of the box, respectively.

Step: 2 Feature Extraction

Appearance Feature Extraction: For each detected bounding box, a deep neural network (such as a CNN) is used to extract a feature vector f_i . This vector captures the appearance information of the object inside the bounding box. The feature vector typically has a fixed length and dimensions.

Step: 3 Motion Prediction

Kalman Filter: Each tracked object is associated with a Kalman filter [78][79] that predicts its next position. The state vector \mathbf{X} of the Kalman filter includes the position and velocity of the object:

$$X_k = [x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]^T \quad (4.1)$$

State Prediction: The Kalman filter predicts the state for the next time step $k + 1$ based on the current state \mathbf{X}_k and the motion model:

$$X_{k+1} = F X_k \quad (4.2)$$

where \mathbf{F} is the state transition matrix.

Covariance Prediction: The predicted error covariance matrix \mathbf{P} is updated as follows:

$$P_{k+1|k} = F P_k F^T + Q \quad (4.3)$$

Where Q is the process noise covariance matrix.

Step: 4 Data Association

Cost Matrix Calculation: A cost matrix \mathbf{C} is calculated based on two metrics:

- *Mahalanobis Distance:* Measures the distance between the predicted Kalman state and the detected bounding boxes.

$$d_{mahal}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \quad (4.4)$$

- *Cosine Distance:* Measures the similarity between appearance feature vectors. The smallest cosine distance between the i -th track and j -th detection in appearance space:

$$d_{cos} = \min \{ 1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in R_i \} \quad (4.5)$$

The combined cost for associating detection j with track i is given by:

$$C_{ij} = \lambda d_{mahal}(d_j, T_i) + (1 - \lambda) d_{cos}(f_j, T_i) \quad (4.6)$$

where T_i is the state of track d_{mahal} is the Mahalanobis distance, d_{cos} is the cosine distance, and λ is a weighting parameter.

Hungarian Algorithm: The Hungarian algorithm is used to solve the assignment problem based on the cost matrix \mathbf{C} , providing the optimal assignment of detections to tracks.

Step: 5 Update

Kalman Filter Update: For each matched detection-track pair, the Kalman filter is updated with the new measurement z :

$$y_k = z_k - Hx_{k|k-1} \quad (4.7)$$

$$S_k = HP_{k|k-1}H^T + R \quad (4.8)$$

$$K_k = P_{k|k-1}H^T S_k^{-1} \quad (4.9)$$

$$x_{k|k} = x_{k|k-1} + K_k Y_k \quad (4.10)$$

$$P_{k|k} = (I - K_k H)P_{k|k-1} \quad (4.11)$$

where y_k is the innovation, H is the measurement matrix, R is the measurement noise covariance, K_k is the Kalman gain, and I is the identity matrix.

Track Management: Tracks are managed based on their states:

- *Confirmed:* Tracks that have been successfully matched for a predefined number of frames.
- *Tentative:* Newly created tracks that are still being confirmed.
- *Deleted:* Tracks that haven't been matched for a certain number of frames are deleted.

Step: 6 Track Initialization

New Track Creation: For detections that are not matched to any existing track, new tracks are initialized. Each new track starts with a state vector x_0 and the corresponding appearance feature vector f_i .

DeepSORT combines motion (using a Kalman filter) and appearance (using deep learning features) for robust tracking. The algorithm leverages data association techniques to handle occlusions and re-identification of objects across frames, making it suitable for real-time multi-object tracking applications.

We use the Multi-view Multi-class Detection dataset CVLAB – EPFL [7]. This Dataset consists of 23 minutes and 57 seconds of synchronized frames taken at 25fps from 6 different calibrated DV cameras. The ground truth contains 242 annotated multi-view non-consecutive frames. The frames contain different real situations where pedestrians, cars and buses appear and can cause high occlusions among them. A total number of 1297 persons, 3553 cars and 56

buses were manually annotated with a bounding box around them. The cameras were calibrated using the Tsai calibration model.

4.3 Experiments and Results

This section discusses the details of datasets, implementation detail and results. We use the CVLAB dataset [19] for Multi-view object detection and tracking. This dataset captures a dynamic scene encompassing 22 meters by 22 meters on the EPFL university campus. It features 23 minutes and 57 seconds of synchronized video footage, recorded from six calibrated DV cameras at 25 frames per second. The cameras are positioned at varying heights, including ground level, first floor, and second floor, offering diverse perspectives. The recording showcases real-world scenarios with persons, cars, and buses inter-acting, potentially causing occlusions. To facilitate analysis, a total of 56 buses, 1297 and 3553 cars have been manually annotated with bounding boxes across 242 non-consecutive multi-view frames. The cameras were calibrated using the Tsai calibration model for accurate spatial mapping. The dataset is then divided into 80% for training, 10% for testing, and 10% for validation.

4.3.1 Implementation Detail

All modules were implemented using Python 3.10.12. The deep learning models were built using the PyTorch framework (version 2.1.0+cu121). The YOLOv7 network was trained with images of 640×640 pixels for 300 epochs, with a mini-batch size of 4 images. The weights for the YOLOv7 model were initialized using a COCO pre-trained model. This implementation was carried out on GOOGLE COLAB PRO. The evaluation of the proposed approach consisted of two parts: first, an assessment of YOLOv7, followed by an evaluation of Deep-SORT. The YOLOv7 model summary includes 415 layers, 37,207,344 parameters, and 37,207,344 gradients. The object categories used were car, person, and bus.

4.3.2 Results

Figure 4.2 illustrates sample image annotations created using the LabelImg tool, a widely used graphical annotation software designed for generating datasets in object detection tasks. The

annotations visually highlight the objects of interest within the image, enclosed by bounding boxes corresponding to their labels.

Figure 4.3 presents the bounding box coordinates generated by the LabelImg tool in the YOLO format. This format represents annotations in a compact and efficient way, detailing the class label, along with the normalized coordinates of the bounding box (center coordinates, width, and height) relative to the image dimensions. These coordinates are essential for training object detection models using the YOLO algorithm.



FIGURE 4.2 Sample annotated image in YOLOv7 format using labelImg tool

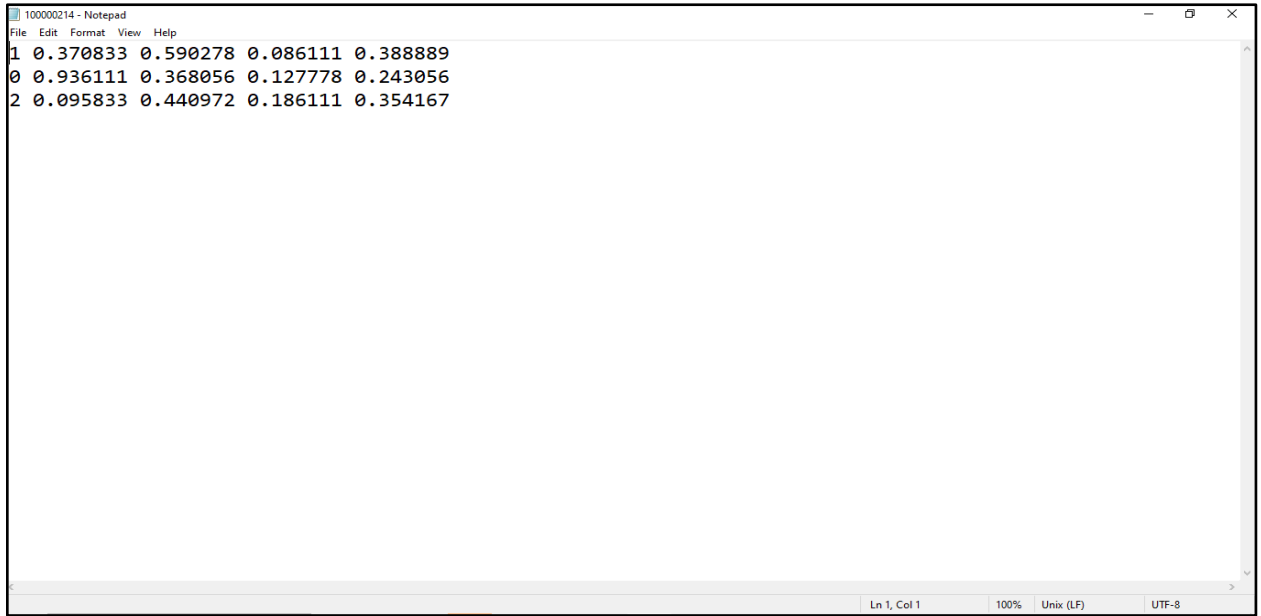


FIGURE 4.3 Sample annotations of the image shown in the previous figure

Table 4.1 provides a detailed performance assessment of the fine-tuned YOLOv7 model on the EPFL Multi-View Multi-Camera Dataset. The first row captures the overall results for all object categories, while subsequent rows break down each class's performance. The table outlines critical metrics, including image size, FLOPs (Floating Point Operations per second), Precision, Recall, and mAP scores at 0.50 and 0.75 thresholds, offering a comprehensive look at the model's capabilities. Additionally, Table 4.2 evaluates the performance of the fine-tuned YOLOv7 in combination with DeepSORT, highlighting comparisons with other methods and datasets, offering insights into relative effectiveness across different approaches.

TABLE 4.1 Performance evaluation of Fine-tuned YOLOv7 on EPFL Multi-View Multi-Camera Dataset

Class	Size	FLOPs	Precision	Recall	mAP@0.5	mAP@0.75
All	640	4.38G	0.923	0.948	0.955	0.761
Car	640	4.38G	0.979	0.997	0.985	0.855
Person	640	4.38G	0.788	0.973	0.985	0.733
Bus	640	4.38G	0.997	0.874	0.894	0.694

TABLE 4.2 Performance evaluation of Fine-tuned YOLOv7 + DeepSORT on EPFL Multi-View Multi-Camera Dataset

	Dataset	MOTA	MOTP
SORT [32]	MOT16 [31]	59.8%	79.6%
Faster RCNN + Deep SORT [33]	MOT16 [31]	61.4%	79.1%
Conditional Random Fields [34]	EPFL Multi-View Multi-Camera Detection Dataset	-	80%
Proposed YOLOv7 + DeepSORT	EPFL Multi-View Multi-Camera Detection Dataset	63.2%	83.0%

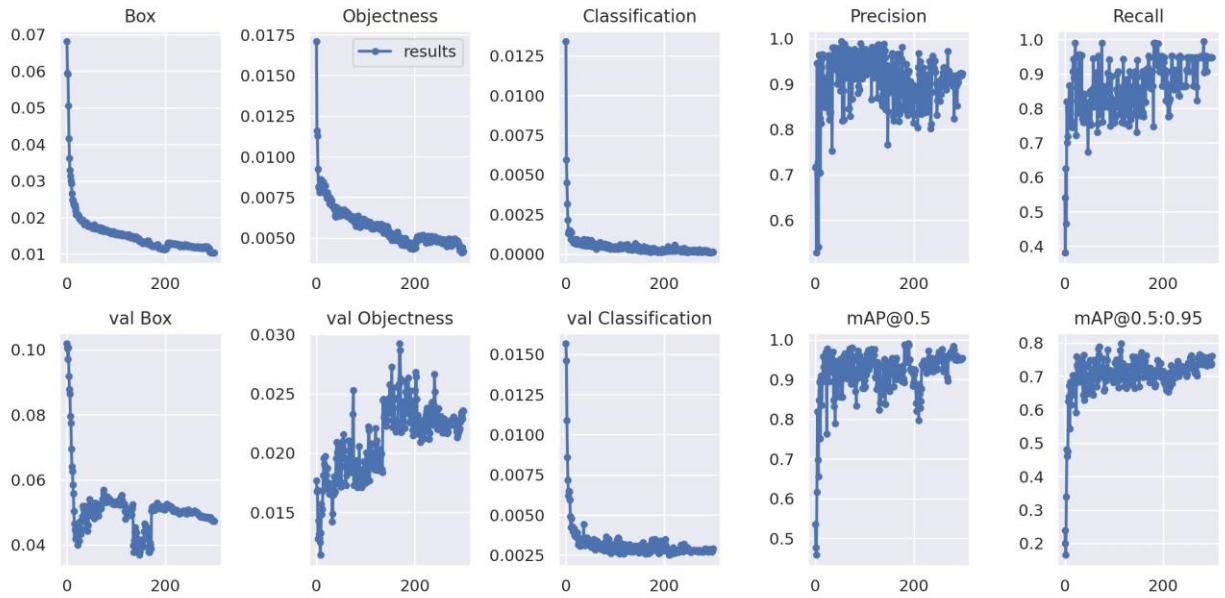


FIGURE 4.4 The convergence of both training and validation losses for the YOLOv7 algorithm object detector and classification is observed at 300 epochs, as demonstrated on the Multi-view multi-camera dataset.

Figure 4.4 shows the convergence of both training and validation losses, as well as performance metrics for the YOLOv7 object detection algorithm, evaluated over 300 epochs

using the EPFL Multi-View Multi-Camera Dataset with three object classes: car, bus, and person. In the top row, the training loss plots are shown and the bottom row, the validation loss and metrics follow a similar trend:

- **Box Loss:** Measures how well the model predicts the bounding box locations for objects. The loss steadily decreases, indicating improving box predictions as training progresses.
- **Objectness Loss:** This refers to the confidence score for detecting any object. It also reduces smoothly, suggesting that the model becomes better at discerning object presence.
- **Classification Loss:** Represents the error in classifying detected objects into categories (car, bus, person). The steep decline shows rapid convergence early on.
- **Precision and Recall:** Both metrics rise toward high values near 1, indicating strong model performance in detecting and correctly classifying objects over the epochs.
- **Validation Box, Objectness, and Classification Loss:** These losses also converge, showing the model generalizes well on the validation set.
- **mAP (Mean Average Precision) at 0.5 and 0.5:0.95:** These are key metrics that combine precision and recall across multiple IoU (Intersection over Union) thresholds. The model maintains strong mAP scores above 0.7 and approaches 1 for mAP@0.5, indicating high accuracy across different IoU thresholds.

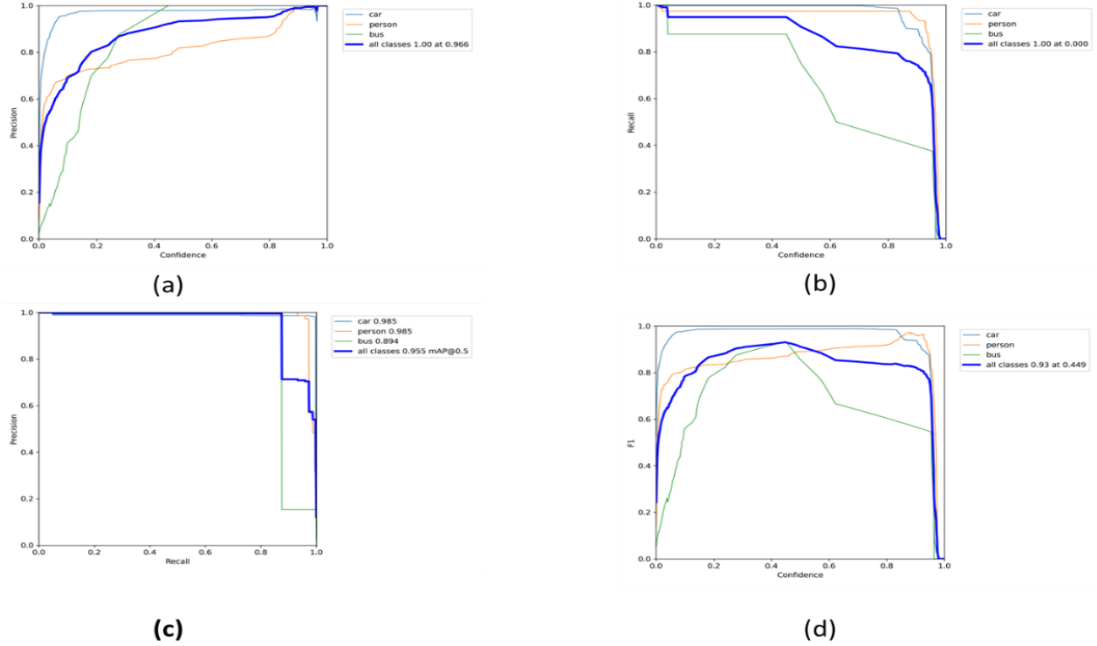


FIGURE 4.5 (a) illustrates the precision (P) plotted against confidence (C) (b) demonstrates the recall plotted against confidence. (c) Correspond to the mean average precision, which is calculated by comparing the ground truth bounding boxes with the detected bounding boxes. (d) highlights the F1 score, reaching 93% at a confidence level of 0.449. This score emphasizes the balance between precision and recall, as observed in the Multi-view multi-camera dataset

Overall, the plots indicate successful training and validation with the losses converging and the model maintaining high precision, recall, and mAP across 300 epochs. Figure 4.5 evaluates the YOLOv7 object detection model on the EPFL Multi-View Multi-Camera Dataset with classes: car, bus, and person. It shows strong performance for objects, with high precision, recall, and F1 scores. Precision vs. confidence and recall vs. confidence plots indicate the model is highly accurate at detecting objects. The combined F1 score of 0.93 at a confidence of 0.449 suggests the model is well-balanced for most object categories.

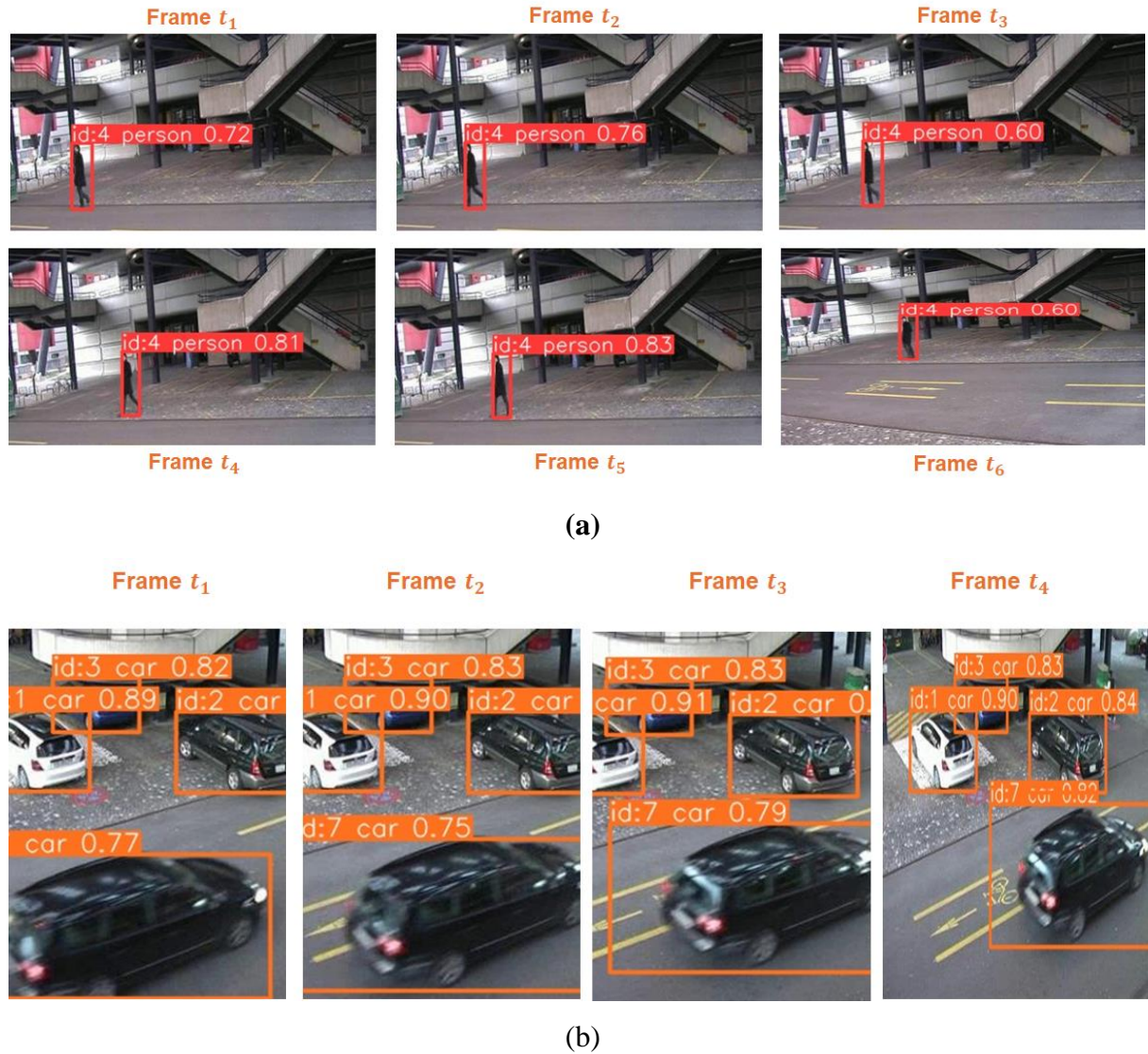


FIGURE 4.6 (a) and (b) Sample output of object tracking using DeepSORT

Figure 4.6 provides a clear depiction of the integration between the DeepSORT algorithm and the fine-tuned YOLOv7 object detector, showcasing their combined effectiveness in object tracking. The sample frames are sourced from the EPFL Multi-View Multi-Camera Dataset, a benchmark designed to test the capabilities of tracking algorithms in complex scenarios. These frames illustrate the model's ability to accurately detect and track multiple objects simultaneously across various camera views.

The results demonstrate the strength of the combined YOLOv7 and DeepSORT approach in addressing challenges associated with multi-object tracking in dynamic, multi-camera

environments. By leveraging YOLOv7's advanced object detection capabilities, fine-tuned for optimal performance, and DeepSORT's efficient tracking methodology, the system ensures continuity and accuracy in identifying and following objects. This integration not only enhances detection precision but also maintains object identities across frames, even in challenging multi-view settings.

This figure underscores the robustness and adaptability of YOLOv7 when paired with DeepSORT, making it well-suited for applications in surveillance, autonomous systems, and other scenarios requiring reliable multi-object detection and tracking. The successful performance depicted in the sample frames highlights the practical potential of this combined approach for real-world implementation in complex visual environments.

4.4 Conclusion and Discussion

In this chapter, we presented a comprehensive approach to multi-camera object detection and tracking by integrating the YOLOv7 object detection framework with the DeepSORT tracking algorithm. The proposed methodology leverages YOLOv7's advancements in real-time detection, such as dynamic anchor boxes and Extended Efficient Layer Aggregation Networks (E-ELAN), to achieve high precision and recall. Simultaneously, the integration with DeepSORT enhances object tracking performance by combining motion prediction with appearance-based re-identification, addressing challenges like occlusions and identity switches. Using the CVLAB-EPFL multi-view multi-camera dataset, our experiments demonstrated the system's robustness in handling dynamic and occlusion-heavy environments. Quantitative evaluations highlighted the superior performance of the YOLOv7 + DeepSORT pipeline, achieving significant improvements in metrics like MOTA, MOTP, and average precision (mAP). Visualizations of training convergence, precision-recall trade-offs, and tracking outputs further validated the efficacy of the proposed approach. The results illustrate the scalability and effectiveness of the system for real-world applications in surveillance, autonomous driving, and other domains requiring continuous multi-camera monitoring. While the approach successfully addresses many challenges inherent in multi-camera object detection and tracking, opportunities for further research remain, such as extending the system to support additional object classes or enhancing its adaptability to diverse environmental

conditions. In summary, this chapter contributes a robust, scalable solution to the field of multi-camera object tracking, emphasizing the synergistic benefits of combining state-of-the-art detection and tracking methodologies.

CHAPTER – 5

Multi-Camera Object Tracking using YOLO and ByteTrack

5.1 Overview

Multi-camera object tracking is an essential area of research within computer vision, enabling the accurate identification and tracking of objects across multiple video feeds. This technology has become increasingly significant due to its applications in various domains, such as surveillance systems, autonomous vehicles, crowd monitoring, and smart city solutions. By utilizing multiple camera perspectives, the technology addresses limitations inherent to single-camera systems, such as occlusions and limited fields of view, and provides a more comprehensive understanding of dynamic and complex environments [90][91].

The integration of data from multiple cameras allows for seamless object tracking, particularly for moving entities like pedestrians and vehicles. This capability enhances situational awareness, safety, and security by enabling the tracking of objects across diverse viewpoints and locations. In scenarios like public spaces, busy intersections, or crowded events, where monitoring from a single perspective is insufficient, multi-camera tracking proves to be a robust solution.

However, despite advancements in object detection and tracking, challenges such as object occlusions, re-identification across camera feeds, and varying environmental conditions, including lighting and background clutter, persist. These challenges underline the need for developing advanced algorithms that can ensure accuracy, reliability, and real-time performance in multi-camera systems.

This chapter explores the techniques and methodologies employed in multi-camera object tracking, with a focus on recent advancements such as YOLOv8 [30] and ByteTrack [24]. The discussion includes their capabilities to address persistent challenges and their effectiveness in real-world applications. By testing these algorithms on multi-camera datasets, this chapter aims to provide insights into improving tracking accuracy and reliability. The findings will

contribute to advancing the field and unlocking the potential for more efficient and practical applications in intelligent surveillance, traffic management, and other critical areas.

5.2 Proposed Approach

The proposed approach aims to develop a robust and efficient multi-camera object tracking system by leveraging state-of-the-art object detection and tracking algorithms. The framework integrates YOLOv8, a cutting-edge object detection model, with ByteTrack, an advanced tracking algorithm, to address the challenges of multi-camera tracking, such as object occlusions, re-identification, and diverse environmental conditions.

5.2.1 YOLOv8 and ByteTrack Overview

The proposed approach for multi-camera object tracking focuses on seamlessly integrating object detection, feature extraction, and cross-camera association to achieve robust and accurate tracking across multiple camera views. This approach addresses challenges such as varying camera perspectives, occlusions, and differences in lighting conditions by leveraging advanced detection and tracking algorithms alongside sophisticated data association techniques.

YOLOv8, an advanced real-time object detection algorithm, builds upon its predecessors with improvements in detection speed and accuracy. Its ability to rapidly process video frames makes it ideal for multi-camera tracking scenarios. In this work, YOLOv8 serves as the primary object detector, providing bounding box predictions for pedestrians across multiple camera feeds.

ByteTrack, a multi-object tracking algorithm, is employed to link the detected objects across consecutive frames. It uses a two-stage association process—first associating high-confidence detections and later linking the remaining detections based on their Intersection over Union (IoU) score. The combination of YOLOv8's precise detection capabilities with ByteTrack's robust tracking ensures the system can maintain accurate tracking across frames and cameras, even in challenging conditions.

Figure 5.1 illustrates the overall workflow of our tracking system, which includes input handling, detection processing, track prediction using Kalman filtering, association of detections with tracks, and track management.

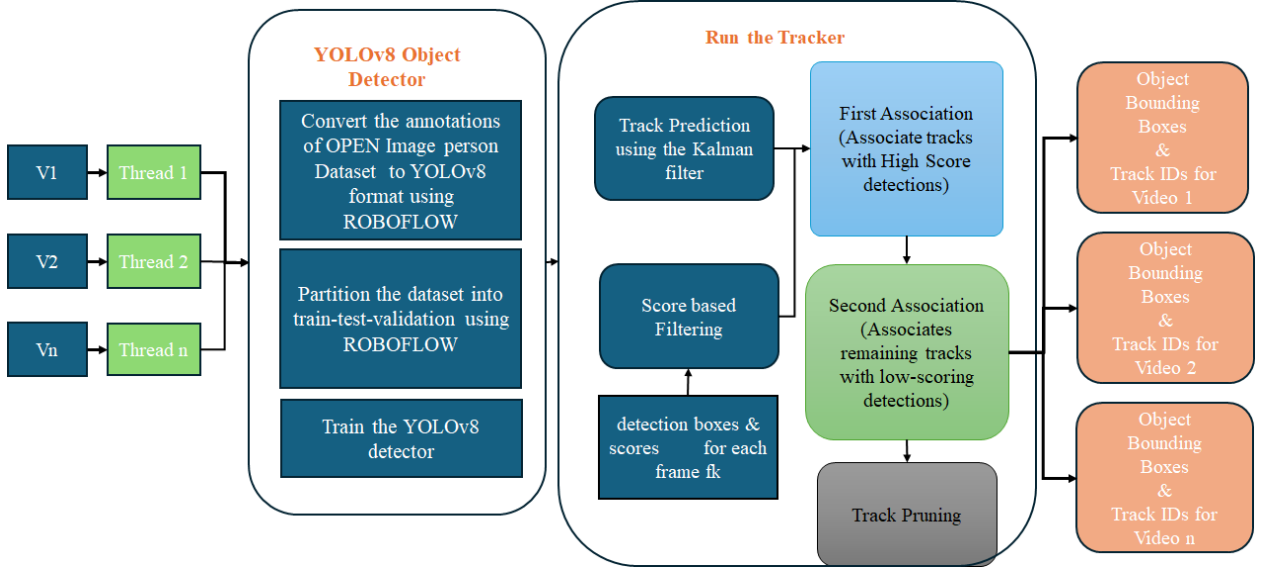


FIGURE 5.1 Proposed YOLOv8 and ByteTrack for Multi-Camera Object Tracking

We implemented a multithreaded tracking system to improve the efficiency of processing multiple video streams. Each thread handles one video stream, ensuring the system can simultaneously process multiple feeds without bottlenecks. Using the threading module in Python, this implementation significantly enhances the tracking system's performance, especially in scenarios with multiple surveillance cameras.

The system receives video inputs from multiple cameras (denoted as V_1, V_2, \dots, V_n). Each video stream is processed in parallel using multithreading, where each thread (Thread 1, Thread 2, ..., Thread n) handles a separate video input. Important steps in this proposed method are explained below:

1. YOLOv8 Object Detector

- The annotations of the input dataset OPEN Image Person Dataset are converted into a format compatible with YOLOv8 using a tool Roboflow. This conversion prepares the data for training the YOLOv8 detector.

- The dataset is partitioned into training, testing, and validation sets, also managed using Roboflow [53], to ensure proper model evaluation and training.
- Once the dataset is prepared, YOLOv8 is trained to detect objects, specifically pedestrians in this setup, within the video frames.

2. *Tracker Prediction:*

After object detection, the system enters the tracking phase:

- For each frame of the video (denoted as f_k), the Kalman filter predicts the new position of each object (track) based on its previous movement.
- The YOLOv8 detector provides detection boxes and corresponding confidence scores for each object. These scores are used to filter detections into two categories: high-confidence and low-confidence detections.

3. *Association:*

- **First Association:** The system performs the first association of detected objects and tracks by matching high-confidence detections with existing tracks. This ensures that the most confident detections are linked to their corresponding object tracks.
- **Second Association:** The remaining unmatched detections (typically low-scoring) are then matched with tracks using a second association. This ensures that even uncertain detections are considered, minimizing track loss.

4. *Track Pruning*

- After both association steps, the system prunes or removes tracks that are not matched with any detections. This helps eliminate false positives or tracks that are no longer valid.

For each video feed (Video 1, Video 2, ..., Video n), the system outputs *Object Bounding Boxes* and *Track IDs* for the detected objects. The bounding boxes specify each object's position, while the Track IDs are unique identifiers assigned to each object to maintain consistency across frames and cameras.

5.2.2 Multi-threaded Tracking

We implemented a multithreaded tracking system to improve the efficiency of processing multiple video streams. Each thread handles one video stream, ensuring the system can simultaneously process multiple feeds without bottlenecks. Using the threading module in Python, this implementation significantly enhances the tracking system's performance, especially in scenarios with multiple surveillance cameras.

5.2.3 Dataset Description and Preprocessing

The datasets used in this study include the Open Images Dataset v6 [36], the Multi-camera Pedestrian Dataset by EPFL [42], and the Real-time Multi-camera Person Dataset. Each dataset presents unique characteristics that aid in evaluating the performance of object detection and tracking algorithms, specifically YOLOv8 and ByteTrack.

The Open Images Dataset v6 is a widely used dataset for object detection tasks, focusing on the category of "Person" in this study. It contains a total of 1,000 images, which are split into 70% for training, 20% for validation, and 10% for testing. A total of 4,036 annotations are available for the "Person" category across the dataset. To adapt the dataset for use with YOLOv8, the annotations were converted to the YOLOv8 format using Roboflow [53]. This tool also facilitated the division of images and annotations into the appropriate sets for training, validation, and testing.

The Multi-camera Pedestrian Dataset by EPFL is a comprehensive dataset comprising two sequences: the Laboratory Sequence [42] and the Passageway Sequence [42], both of which feature overlapping camera views. In the Laboratory Sequence, four cameras are strategically positioned about 2 meters above the ground, capturing video footage of persons entering in sequence and moving around a laboratory environment. Each recording lasts for approximately 2.5 minutes. The video is recorded at a frame rate of 25 frames per second (fps) and encoded using the MPEG-4 codec, providing high-quality data for pedestrian detection and tracking. The Passageway Sequence presents a more challenging scenario, as it was filmed in an underground passageway leading to a train station, a location characterized by poor lighting. This sequence was captured using four DV cameras, each recording at 25 fps, with

the footage encoded using the Indeo 5 codec. The low-light conditions of the passageway make this sequence particularly difficult for tracking algorithms, as it adds complexity to the task of maintaining accurate object detection.

Lastly, the Real-time Multi-camera Person Dataset was recorded in a laboratory setting using two cameras positioned in non-overlapping views. This dataset features individuals entering in sequence and walking around the room for one minute. One camera focuses on the main entrance of the premises, while the other covers a laboratory passageway that is subject to poor lighting. The non-overlapping nature of the camera views provides an additional challenge for object-tracking systems, as individuals must be tracked across distinct areas without the benefit of continuous visual coverage.

The datasets represent real-world challenges such as occlusions, and varying lighting, which are critical for evaluating the robustness of tracking algorithms.

5.2.4 Proposed Algorithm for Multi-camera Object Tracking

Figure 5.2 shows an algorithm represented as a flowchart, detailing the process for **multi-camera object tracking** using YOLOv8 and ByteTrack.

Multiple video streams from different cameras are fed into the system as input. YOLOv8 processes each video frame, detecting objects (likely pedestrians) and generating bounding boxes around the detected objects. It also provides confidence scores for each detection. The system applies a Kalman Filter [46] to predict the new positions of the detected objects (tracks) in subsequent frames based on their previous positions. Detected objects (bounding boxes) are associated with their corresponding predicted tracks. This is done by matching the current detection with the expected location from the Kalman Filter. Two association stages seem to occur:

- **First Association Stage:** High-confidence detections are linked to existing tracks.
- **Second Association Stage:** Remaining detections (likely those with lower confidence scores) are matched to tracks using a secondary matching mechanism, possibly based on Intersection over Union (IoU) [24].

Algorithm: Pseudo-code of YOLOv8 and ByteTrack-based Algorithm for Multi-Camera Object Tracking**Input:**

- Object Detector (YOLOv8) with detection threshold θ
- Video sequence V_1, V_2, \dots, V_n
- Number of input videos n

Output: Tracks T_1, T_2, \dots, T_n for each video

```

1  tracker_run(DetV8,  $\theta$ ,  $V$ )
2  Initialize Track  $T = \emptyset$ 
3  For each frame  $f_i$  in the video  $V$ :
4      Obtain detection boxes and scores using the YOLOv8 model:
6       $D_k = \text{DetV8}(f_i)$ 
7      Initialize two sets:  $D_H = \emptyset, D_L = \emptyset$ 
8      For each detection  $d$  in  $D_K$ :
9          If  $d.\text{score} > \theta$ , add  $d$  to  $D_H$ 
10         Otherwise, add  $d$  to  $D_L$ 
11  Update Tracks:
12  For each track  $t$  in  $T$ , predict its new position using a Kalman Filter:
13   $t = \text{KalmanFilter}(t)$ 
14  First Association:
15  Associate objects in  $T$  with detections in  $D_H$ , using Re-ID feature distances.
16  Track Unmatched Detections:
17   $D_{\text{remain}} = \text{Remaining object detections form } D_H$ 
18   $T_{\text{remain}} = \text{Remaining object detections form } T$ 
19  Second Association:
20  Match the remaining tracks in  $T_{\text{remain}}$  with detections in  $D_L$ , using IoU similarity.
21   $T_{\text{re-remain}} = \text{Remaining object detections form } T_{\text{remain}}$ 
22  Update Unmatched Tracks:
23  Remove unmatched tracks from the set:  $T = T / T_{\text{re-remain}}$ 
24  Return the updated tracks  $V$ .
25  tracker-thread-1  $\rightarrow$  tracker_run(Det,  $\Theta$ ,  $V_1$ )
26  tracker-thread-2  $\rightarrow$  tracker_run(Det,  $\Theta$ ,  $V_2$ )

```

FIGURE 5.2 Proposed Algorithm for Multi-Camera Object Tracking using YOLOv8 and ByteTrack

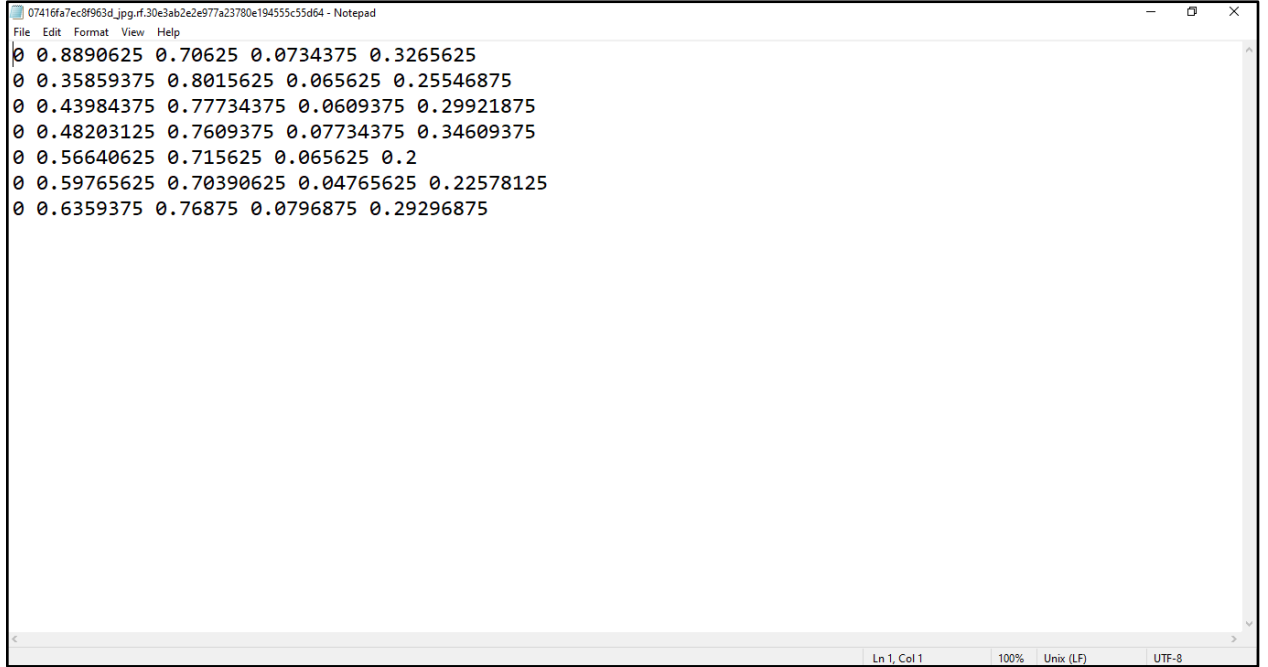
The system updates the status of existing tracks, such as confirming or terminating a track if it can no longer find a match in the current frame. Each video stream is processed in a separate thread to ensure parallel execution, improving processing efficiency. Figure 5.2 provides an overview of the YOLOv8 and ByteTrack pipeline for tracking objects across multiple camera feeds, leveraging Kalman Filters for motion prediction and a multi-stage matching process to maintain track consistency across frames.

5.3 Results

The performance of the YOLOv8 and ByteTrack system was evaluated using several metrics, including Precision, Recall, mean Average Precision (mAP), Multi-Object Tracking Accuracy (MOTA), and Multi-Object Tracking Precision. These metrics provide insights into the system's detection and tracking capabilities, with MOTA and MOTP measuring the overall tracking quality [20]. Figures 5.3 shows the image with a bounding box and annotations of sample image taken from ROBOFLOW. Figure 5.4 shows the object class and bounding box coordinates shown in figure 5.4.



FIGURE 5.3 Sample annotated image taken from Roboflow

**FIGURE 5.4 Annotations for person in YOLOv8 format**

All modules were developed using Python version 3.10.12. The deep learning models were implemented with the PyTorch framework (torch-2.2.1, cu121) and executed on a Tesla T4 GPU, which has 15102 MiB of memory. Both YOLOv8 and ByteTrack were implemented in the Google Colab PRO environment. YOLOv8 was specifically trained on the "Person" category from the Open Image Dataset, with initial weights derived from the COCO pre-trained model. The YOLOv8 model consists of 225 layers, 3,011,238 parameters, 3,011,222 gradients, and performs 8.2 GFLOPs. For ByteTrack, a high association threshold of 0.5 was used for the initial frame-to-frame link, while a lower threshold of 0.2 was applied for subsequent associations.

TABLE 5.1 Performance evaluation of Fine-tuned YOLOv8 on Open Image Dataset

Size	GFLOPs ₇₄	Precision	Recall	<u>mAP@0.5</u>
640	8.1	0.64	0.521	0.55

TABLE 5.2 Performance evaluation ByteTrack on Multi-camera Pedestrian Dataset and Real-time Multi-camera person dataset with non-overlapping camera view

Method	Dataset	MOTA	MOTP
K-Shortest Paths [45]	EPFL Passageway Sequence	68%	82%
	EPFL Laboratory Sequence	70%	83%
POM [42]	EPFL Passageway Sequence	68%	70%
	EPFL Laboratory Sequence	72%	78%
Proposed YOLOv8 + ByteTrack	EPFL Passageway Sequence	70.25%	83.3%
	EPFL Laboratory Sequence	72.14%	84.6%
	Real-time Multi-camera person dataset with non-overlapping camera view	75.60%	86.8%

Table 5.1 displays the results of fine-tuning YOLOv8 on the Open Image Dataset, providing detailed performance metrics for the model after optimization. In Table 5.2, a comparison is made between the proposed YOLOv8 and ByteTrack system and other object tracking methods, including K-Shortest Paths and Probability Occupancy Map (POM). This comparison was conducted using the EPFL Passageway and Laboratory sequences, which present challenging tracking scenarios. The proposed system outperformed the other methods in terms of tracking accuracy, particularly in environments with difficult conditions such as low lighting and occlusions, demonstrating its ability to maintain object tracking even in complex settings.

Additionally, the system was evaluated on a real-time multi-camera person dataset, which features non-overlapping camera views. In this evaluation, the system achieved a Multi-Object Tracking Accuracy (MOTA) score of 75.60% and a Multi-Object Tracking Precision (MOTP) score of 86.8%. These results underscore the robustness and effectiveness of the YOLOv8 and ByteTrack integration, highlighting its ability to handle complex multi-object tracking tasks

across multiple camera perspectives. The high MOTA score indicates the system's capability to accurately track objects across various frames with minimal errors, while the MOTP score reflects the precision of the bounding boxes, ensuring accurate object localization even in challenging tracking conditions. Overall, these results validate the proposed system's effectiveness in real-world applications, particularly for multi-camera tracking in dynamic environments.

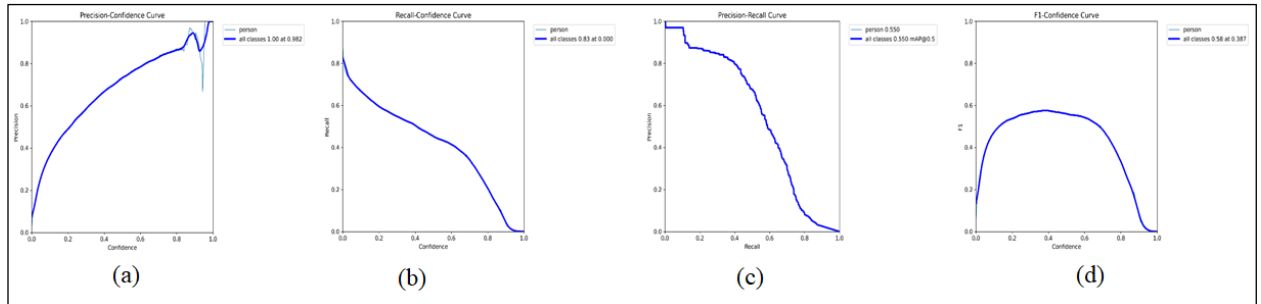


FIGURE 5.5 (a) the precision (P) plotted against confidence (C) (b) The recall plotted against confidence. (c) the mean average precision, which is calculated by comparing the ground truth bounding boxes with the detected bounding boxes. Additionally (d) the IDF1 score

The figure presents four evaluation plots for object detection on the "person" class: (a) illustrates how precision improves with confidence, reaching nearly 1.0 at higher confidence levels. (b) depicts recall starting at 0.83 and declining as confidence increases. (c) demonstrates a trade-off between precision and recall, with the mAP@0.5 for all classes at 0.550. Finally, (d) shows the F1 score peaking at 0.58 around a confidence value of 0.4 before dropping at higher confidence levels. Together, these plots highlight the trade-offs between precision, recall, and confidence for the detection model.

**FIGURE 5.6 Sample video file and annotations**

The image above displays a sample frame along with its corresponding annotation file for tracked objects. The video frame is saved as video_name_frame_number.jpg, while the annotation file is named video_name_frame_number.txt. For Example, the above file names are 4p_c0_168.jpg and 4p_c0_168.txt. Each line in the annotation file describes a single object instance, where the first value indicates the class, the next four values represent the object's bounding box coordinates, and the sixth value specifies the object's identity number.

Table 5.3 Data Format for Evaluation of Object Tracking

Position	Name
1	Class
2	Bounding box left
3	Bounding box top
4	Bounding box width
5	Bounding box height
6	Identity number

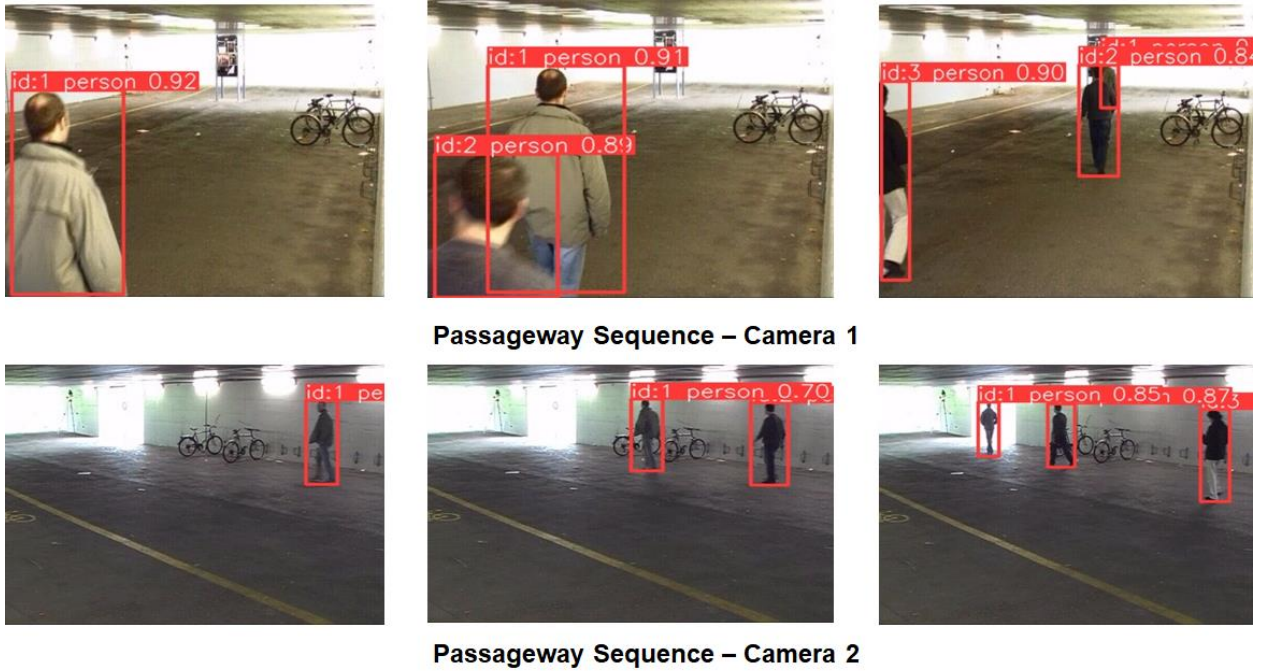


FIGURE 5.7 Sample video frames for the passageway sequence

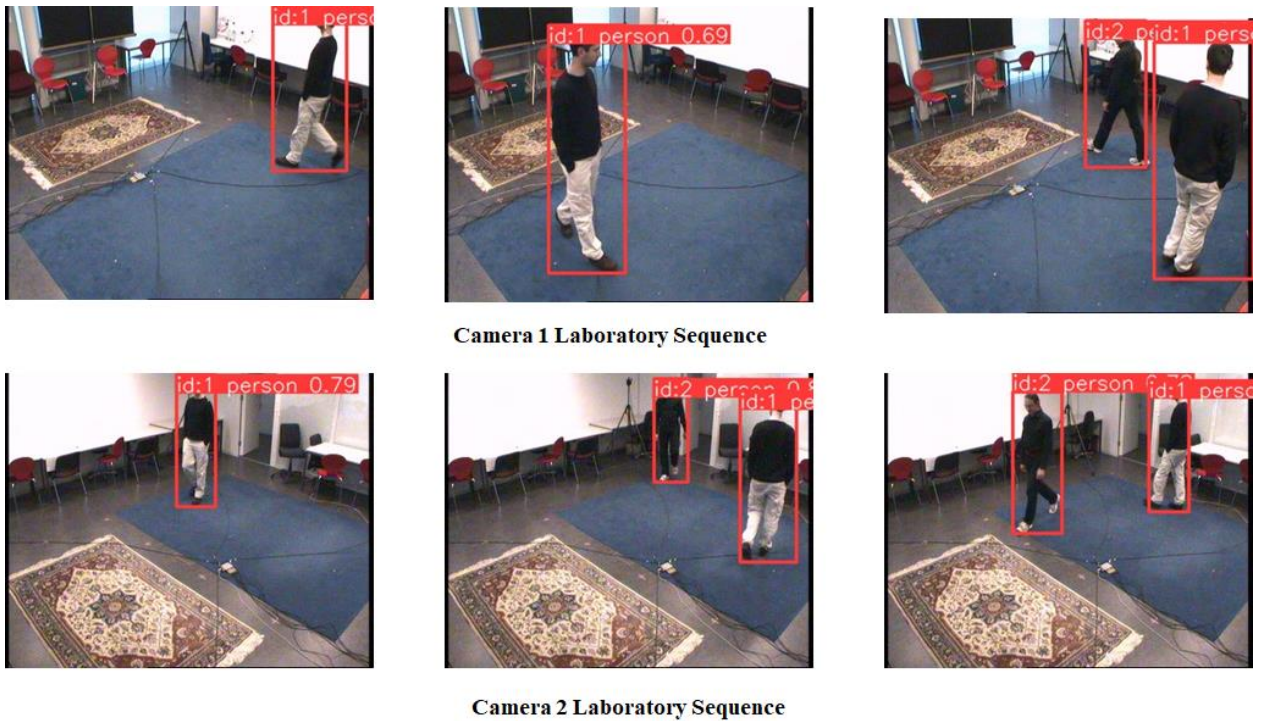


FIGURE 5.8 Sample video frames for the laboratory sequence



FIGURE 5.9 Sample video of multi-camera person dataset with non-overlapping camera view – Laboratory Sequence



FIGURE 5.10 Sample output of multi-camera person dataset with non-overlapping camera view – Main Entry

Figure 5.7 presents the sample output of video frames from the passageway sequence captured by two cameras. Figure 5.8 displays sample frames from the laboratory sequence. Figure 5.9 illustrates the sample output of the multi-camera person dataset featuring non-overlapping camera views in the laboratory sequence. Figure 5.10 showcases the sample output of the multi-camera person dataset with non-overlapping camera views in the main entry sequence.

5.4 Conclusion and Discussion

In this work, we presented an approach to multi-camera object tracking by integrating YOLOv8 and ByteTrack. YOLOv8 served as the primary object detector, offering high detection accuracy and speed, while ByteTrack complemented it with robust tracking

capabilities across frames and camera views. By leveraging a multithreaded implementation, our system efficiently processed video streams from multiple cameras, ensuring scalability and real-time performance.

The evaluation using diverse datasets, including the EPFL multi-camera pedestrian dataset and a real-time multi-camera person dataset, demonstrated the effectiveness of the proposed method. The system achieved superior tracking accuracy and precision in challenging scenarios such as occlusions, low-light environments, and non-overlapping camera views. Specifically, our approach outperformed existing methods like K-Shortest Paths and Probability Occupancy Maps, achieving notable improvements in Multi-Object Tracking Accuracy (MOTA) and Precision (MOTP).

These results underline the potential of combining advanced object detection algorithms like YOLOv8 with robust tracking mechanisms such as ByteTrack for real-world applications. This approach holds promise for various domains, including surveillance, autonomous systems, and traffic monitoring. Future work may explore enhancing cross-camera re-identification and adapting the system for more diverse and complex environments.

CHAPTER – 6

Conclusion and Future Scope

6.1 Conclusion

This thesis has addressed the challenges and advancements in multi-view object detection and tracking, focusing on integrating cutting-edge deep learning models and tracking algorithms to develop robust solutions for real-world applications. The research combined state-of-the-art object detection models such as YOLOv7, and YOLOv8, and tracking algorithms like DeepSORT and ByteTrack to handle complex scenarios in multi-view environments, delivering significant improvements in performance and adaptability.

The initial part of this study highlighted the critical role of image data augmentation in enhancing the performance of deep learning models for object detection tasks. Various augmentation techniques, such as altering object perspectives and adding noise, were applied to diversify the training data, leading to substantial improvements in mean Average Precision (mAP) across several detection models, including Centernet, EfficientDet, SSD, and FasterRCNN. These results, tested on the Open Image Dataset v6, demonstrate that data augmentation can help models generalize better to different scenes and lighting conditions, making the models more adaptable to real-world conditions. The importance of data augmentation in improving model robustness cannot be overstated, especially in situations where acquiring diverse and labeled data is challenging.

This research also extended its exploration of object detection to autonomous driving applications, where the YOLOv8 model was trained on the Udacity Self-Driving Car Dataset. The study found that YOLOv8, combined with multi-view detection techniques, performs exceptionally well in recognizing objects in various driving conditions, including complex lighting and weather scenarios. Accurate and timely detection of objects such as pedestrians, vehicles, and road signs are paramount for the development of reliable autonomous vehicle systems. By training models to detect objects from different angles and under varying conditions, this research has contributed to improving the performance and safety of autonomous driving technology. These findings demonstrate that deep learning models like YOLOv8 can be

instrumental in pushing forward the capabilities of autonomous systems, particularly in real-world applications where detection accuracy is vital for decision-making processes.

In addition to object detection, this thesis explored the potential of multi-camera tracking systems by integrating DeepSORT and ByteTrack with YOLOv7 and YOLOv8 respectively. These systems were tested in dynamic environments with overlapping and non-overlapping camera views, such as in surveillance and monitoring systems. One of the main challenges addressed in this study was the accurate tracking of objects across multiple camera feeds, especially under challenging conditions like occlusions and poor lighting. The experiments demonstrated that combining powerful object detectors like YOLOv7 and YOLOv8 with robust tracking algorithms like DeepSORT and ByteTrack allows for consistent and accurate tracking of objects, even in environments with complex interactions. This research showed that such systems are highly adaptable and suitable for a range of applications, from public safety surveillance to smart city monitoring.

Another important finding was the models' ability to perform well under challenging real-world conditions, such as low-light environments and occlusions adaptability of these methods. The research revealed that even in difficult scenarios, the proposed object detection and tracking systems maintained high levels of accuracy and reliability, highlighting their practical applicability in real-world environments, such as traffic monitoring and security surveillance. This makes the proposed approach particularly relevant for applications that require constant monitoring of public spaces, transportation hubs, or sensitive areas.

Overall, this thesis contributes to the broader field of computer vision by providing valuable insights into the advantages of multi-view object detection and tracking, particularly through the use of fine-tuned deep learning models and advanced tracking algorithms. The study has demonstrated how the combination of robust object detection with multi-view tracking techniques can significantly enhance the effectiveness of systems used in surveillance, autonomous driving, and other areas requiring high levels of accuracy in object detection and tracking.

However, addressing the identified limitations will be crucial for further improving system performance. Future research should focus on enhancing detection in crowded environments,

improving tracking consistency across multiple camera views, optimizing computational efficiency for real-time applications, and integrating multimodal sensor data (e.g., LiDAR, thermal imaging) to improve robustness under challenging conditions. Despite these challenges, the proposed methodologies provide a foundation for future object detection and tracking advancements, paving the way for more efficient, scalable, and adaptable vision-based systems for real-world applications.

6.2 Future Scope

Looking forward, there are several promising directions for future research. One key area involves further optimization of these models for real-time performance. Techniques like model pruning and quantization could enhance the computational efficiency of these models without sacrificing accuracy. Another area of future exploration includes handling even more complex environments, such as those with heavy occlusions, motion blur, or cluttered backgrounds, which will require the development of new algorithms or the refinement of existing ones. Additionally, integrating multimodal data—such as infrared, LiDAR, or radar—can enhance the detection and tracking capabilities of these models, particularly in challenging conditions like fog, rain, or low visibility.

Furthermore, long-term tracking and re-identification across large time gaps or between different camera views remains an open challenge. Future work could improve the consistency and accuracy of tracking objects that temporarily leave the field of view and reappear later or in different cameras. This would be crucial for applications like smart city surveillance, where the ability to track objects over extended periods and across multiple locations is essential.

In summary, this thesis comprehensively examines multi-view object detection and tracking, highlighting the potential of deep learning models and advanced tracking algorithms in real-world applications. The findings pave the way for further innovations in the field, ensuring that future systems will be more reliable, efficient, and capable of addressing increasingly complex challenges in object detection and tracking.

List of Publications

1. Pandya, N. A., & Chauhan, N. (2022). Survey Paper on Multi-view Object Detection: Challenges and Techniques. In *ICT Infrastructure and Computing: Proceedings of ICT4SD 2022* (pp. 1-10). Singapore: Springer Nature Singapore. (Scopus Index)
2. Pandya, N. A., & Chauhan, N. C. (2024). Multi-Camera Person Tracking: Integrating YOLOv8 with ByteTrack. *International Journal of Electrical and Electronics Engineering*, 11(10), 53–60. (Scopus Index)
3. Pandya, N. A., & Chauhan, N (2024). Application Of Multi-View Object Detection in Autonomous Driving Using Deep Learning Approach. *International Journal of Engineering Applied Sciences and Technology*, 8(12), 227–233.

References

- [1] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE access*, 7, 128837-128868.
- [2] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I).
- [3] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893).
- [4] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).
- [5] Girshick, R. (2015). Fast r-cnn. *arXiv preprint arXiv:1504.08083*.
- [6] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137-1149.
- [7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.
- [8] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781-10790).
- [9] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6569-6578).

- [10]Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [11]Simonyan, K. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [12]Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.
- [13]Ross, T. Y., & Dollár, G. K. H. P. (2017, July). Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2980-2988).
- [14]Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- [15]Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7464-7475).
- [16]Redmon, J. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [17]Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [18]Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2778-2788).
- [19]Hou, Y., Zheng, L., & Gould, S. (2020). Multiview detection with feature perspective transformation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow*,

- UK, August 23–28, 2020, *Proceedings, Part VII 16* (pp. 1-18). Springer International Publishing.
- [20] Zhang, F., Ma, Y., Yuan, G., Zhang, H., & Ren, J. (2021). Multiview image generation for vehicle reidentification. *Applied Intelligence*, 51(8), 5665-5682.
- [21] Zhou, Y. (2018). *Deep Visual Feature Learning for Vehicle Detection, Recognition and Re-identification* (Doctoral dissertation, University of East Anglia).
- [22] Yang, A. Y., Maji, S., Christoudias, C. M., Darrell, T., Malik, J., & Sastry, S. S. (2009, August). Multiple-view object recognition in band-limited distributed camera networks. In *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* (pp. 1-8). IEEE.
- [23] Farfade, S. S., Saberian, M. J., & Li, L. J. (2015, June). Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 643-650).
- [24] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., ... & Wang, X. (2022, October). Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision* (pp. 1-21). Cham: Springer Nature Switzerland.
- [25] Milan, A. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.
- [26] Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2007). Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2), 267-282.
- [27] Liu, W., Camps, O., & Sznai, M. (2017). Multi-camera multi-object tracking. *arXiv preprint arXiv:1709.07065*.
- [28] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)* (pp. 3464-3468). IEEE.

- [29] Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645-3649). IEEE.
- [30] Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-time flying object detection with YOLOv8. *arXiv preprint arXiv:2305.09972*.
- [31] Dendorfer, P. (2020). Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*.
- [32] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.
- [33] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [34] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
- [35] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.
- [36] Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., ... & Murphy, K. (2017). Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from <https://github.com/openimages>*, 2(3), 18.
- [37] Tang, C., Ling, Y., Yang, X., Jin, W., & Zheng, C. (2018). Multi-view object detection based on deep learning. *Applied Sciences*, 8(9), 1423.

- [38] Farfade, S. S., Saberian, M. J., & Li, L. J. (2015, June). Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 643-650).
- [39] Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). Sharing visual features for multiclass and multiview object detection. *IEEE transactions on pattern analysis and machine intelligence*, 29(5), 854-869.
- [40] Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1907-1915).
- [41] Roig, G., Boix, X., Shitrit, H. B., & Fua, P. (2011, November). Conditional random fields for multi-camera object detection. In *2011 International Conference on Computer Vision* (pp. 563-570). IEEE.
- [42] Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2007). Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2), 267-282.
- [43] Alahi, A., Jacques, L., Boursier, Y., & Vandergheynst, P. (2011). Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision*, 41, 39-58.
- [44] Bae, S. H., & Yoon, K. J. (2017). Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 595-610.
- [45] Berclaz, J., Fleuret, F., Turetken, E., & Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9), 1806-1819.
- [46] Li, Q., Li, R., Ji, K., & Dai, W. (2015, November). Kalman filter and its application. In *2015 8th international conference on intelligent networks and intelligent systems (ICINIS)* (pp. 74-77). IEEE.

- [47] Li, Q., Li, R., Ji, K., & Dai, W. (2015, November). Kalman filter and its application. In *2015 8th international conference on intelligent networks and intelligent systems (ICINIS)* (pp. 74-77). IEEE.
- [48] Pérez, P., Hue, C., Vermaak, J., & Gangnet, M. (2002). Color-based probabilistic tracking. In *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I 7* (pp. 661-675). Springer Berlin Heidelberg.
- [49] Ristani, E., & Tomasi, C. (2018). Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6036-6046).
- [50] Sinha, S. N., Pollefeys, M., & McMillan, L. (2004, June). Camera network calibration from dynamic silhouettes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (Vol. 1, pp. I-I). IEEE.
- [51] Li, Y. L., Li, H. T., & Chiang, C. K. (2022). Multi-Camera Vehicle Tracking Based on Deep Tracklet Similarity Network. *Electronics*, 11(7), 1008.
- [52] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11), 1330-1334.
- [53] Zhang, S., Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2016). How far are we from solving pedestrian detection?. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1259-1267).
- [54] Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow (Version 1.0) [Software]. Available from <https://roboflow.com>. computer vision.
- [55] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

- [56] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [57] Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, 13(1), 89.
- [58] Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, 13(1), 89.
- [59] Fiaz, M., Mahmood, A., & Jung, S. K. (2018). Tracking noisy targets: A review of recent object tracking approaches. *arXiv preprint arXiv:1802.03098*.
- [60] Li, P., Wang, D., Wang, L., & Lu, H. (2018). Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76, 323-338.
- [61] Verma, R. (2017). A review of object detection and tracking methods. *International Journal of Advance Engineering and Research Development*, 4(10), 569-578.
- [62] Soleimanitaleb, Z., Keyvanrad, M. A., & Jafari, A. (2020, October). Improved MDNET Tracker in Better Localization Accuracy. In *2020 10th International Conference on Computer and Knowledge Engineering (ICCCKE)* (pp. 221-226). IEEE.
- [63] Soleimanitaleb, Z., & Keyvanrad, M. A. (2022). Single object tracking: A survey of methods, datasets, and evaluation metrics. *arXiv preprint arXiv:2201.13066*.
- [64] Yao, R., Lin, G., Xia, S., Zhao, J., & Zhou, Y. (2020). Video object segmentation and tracking: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(4), 1-47.
- [65] Schubert, F., Casaburo, D., Dickmanns, D., & Belagiannis, V. (2015). Revisiting robust visual tracking using pixel-wise posteriors. In *Computer Vision Systems: 10th*

- International Conference, ICVS 2015, Copenhagen, Denmark, July 6-9, 2015, Proceedings 10* (pp. 275-288). Springer International Publishing.
- [66] Bibby, C., & Reid, I. (2008). Robust real-time visual tracking using pixel-wise posteriors. In *Computer Vision—ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II 10* (pp. 831-844). Springer Berlin Heidelberg.
- [67] Milan, A., Leal-Taixé, L., Schindler, K., & Reid, I. (2015). Joint tracking and segmentation of multiple targets. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5397-5406).
- [68] Sun, Z., Chen, J., Chao, L., Ruan, W., & Mukherjee, M. (2020). A survey of multiple pedestrian tracking based on tracking-by-detection framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5), 1819-1833.
- [69] Brown, M., Funke, J., Erlien, S., & Gerdes, J. C. (2017). Safe driving envelopes for path tracking in autonomous vehicles. *Control Engineering Practice*, 61, 307-316.
- [70] Laurence, V. A., Goh, J. Y., & Gerdes, J. C. (2017, May). Path-tracking for autonomous vehicles at the limit of friction. In *2017 American control conference (ACC)* (pp. 5586-5591). IEEE.
- [71] Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3061-3070).
- [72] Colchester, A. C., & Hawkes, D. J. (Eds.). (1991). *Information Processing in Medical Imaging: 12th International Conference, IPMI'91, Wye, UK, July 7-12, 1991. Proceedings* (Vol. 511). Springer Science & Business Media.
- [73] Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. (2002, September). The intelligent ASIMO: System overview and integration. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 3, pp. 2478-2483). IEEE.

- [74] Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. (2002, September). The intelligent ASIMO: System overview and integration. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 3, pp. 2478-2483). IEEE.
- [75] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kämäräinen, J. K., ... & Fernández, G. (2021). The ninth visual object tracking vot2021 challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2711-2738).
- [76] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kämäräinen, J. K., ... & Fernández, G. (2021). The ninth visual object tracking vot2021 challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2711-2738).
- [77] Wang, T., & Ling, H. (2017). Gracker: A graph-based planar object tracker. *IEEE transactions on pattern analysis and machine intelligence*, 40(6), 1494-1501.
- [78] Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(1), 3-19.
- [79] Aghajan, H., & Cavallaro, A. (Eds.). (2009). *Multi-camera networks: principles and applications*. Academic press.
- [80] Amosa, T. I., Sebastian, P., Izhar, L. I., Ibrahim, O., Ayinla, L. S., Bahashwan, A. A., ... & Samaila, Y. A. (2023). Multi-camera multi-object tracking: a review of current trends and future advances. *Neurocomputing*, 552, 126558.
- [81] Pei, Y., Biswas, S., Fussell, D. S., & Pingali, K. (2019). An elementary introduction to Kalman filtering. *Communications of the ACM*, 62(11), 122-133.
- [82] Li, Q., Li, R., Ji, K., & Dai, W. (2015, November). Kalman filter and its application. In *2015 8th international conference on intelligent networks and intelligent systems (ICINIS)* (pp. 74-77). IEEE.

- [83] Leal-Taixé, L. (2015). Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*.
- [84] Ferryman, J., & Shahrokni, A. (2009, December). Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance* (pp. 1-6). IEEE.
- [85] Milan, A. (2016). MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.
- [86] Tang, C., Ling, Y., Yang, X., Jin, W., & Zheng, C. (2018). Multi-view object detection based on deep learning. *Applied Sciences*, 8(9), 1423.
- [87] Farfade, S. S., Saberian, M. J., & Li, L. J. (2015, June). Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 643-650).
- [88] Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). Sharing visual features for multiclass and multiview object detection. *IEEE transactions on pattern analysis and machine intelligence*, 29(5), 854-869.
- [89] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 211-252.
- [90] Yang, L., Fan, Y., & Xu, N. (2019). Video instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 5188-5197).
- [91] Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., & Ling, H. (2021). Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), 7380-7399.
- [92] Dave, A., Khurana, T., Tokmakov, P., Schmid, C., & Ramanan, D. (2020). Tao: A large-scale benchmark for tracking any object. In *Computer Vision—ECCV 2020: 16th European*

- Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16* (pp. 436-454). Springer International Publishing.
- [93] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kämäräinen, J. K., ... & Fernández, G. (2021). The ninth visual object tracking vot2021 challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2711-2738).
- [94] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2013). Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7), 1442-1468.
- [95] Wang, T., & Ling, H. (2017). Gracker: A graph-based planar object tracker. *IEEE transactions on pattern analysis and machine intelligence*, 40(6), 1494-1501.
- [96] Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, 13(1), 89.
- [97] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4), 18-28.
- [98] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [99] Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- [100] Liao, Y., Xie, J., & Geiger, A. (2022). Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3), 3292-3310.
- [101] Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow (Version 1.0) [Software]. <https://roboflow.com>. [computer vision](#).

- [102] Feng, D., Harakeh, A., Waslander, S. L., & Dietmayer, K. (2021). A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 9961-9980.
- [103] Everingham, M. (2007). The pascal visual object classes challenge,(voc2007) results. <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/index.html>.
- [104] Yang, F., Odashima, S., Yamao, S., Fujimoto, H., Masui, S., & Jiang, S. (2024). A unified multi-view multi-person tracking framework. *Computational Visual Media*, 10(1), 137-160.