PEDESTRIAN TRACK & OBJECT DETECTION FOR EMERGENCY BRAKING SYSTEM

Munzurul Islam
221-35-921

Bachelor of Science
In
Software Engineering
Daffodil International University, Dhaka, Bangladesh.

DECEMBER 2025

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Munzurul Islam
Date of Birth	:	21-06-2002
Title	:	Pedestrian Track and Object Detection for Emergency Braking System.
Academic Session	:	Fall 2025

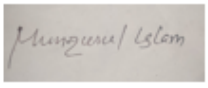
I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done) *
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Thesis is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the thesis for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)

221-35-921
Student ID
Date: 27 November 2025

(Supervisor's Signature)

Dr. Md. Fazla Elahe
Name of Supervisor
Date: 27 November 2025

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Daffodil International University,
Daffodil Smart City, Ashulia, Dhaka, Bangladesh.

Dear Sir,

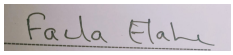
CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name: Munzurul Islam
Thesis Title: Pedestrian Track and Object Detection for Emergency Braking System.

Thank you.

Yours faithfully,



(Supervisor's Signature)

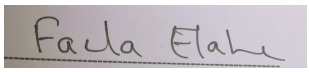
Date: 27 November 2025

Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, Daffodil International University with its copy attached to the thesis.

Supervisor Declaration

I am Dr. Md. Fazla Elahe associate head and assistant professor of the department of Software Engineering hereby Declare that I have checked this thesis and that in my opinion, this thesis is adequate in terms of scope and quality for the award of the Bachelor of Science degree.



(Supervisor Signature)

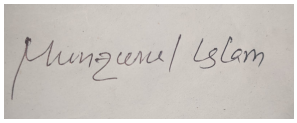
Name : Dr. Md. Fazla Elahe

Position : Assistant Professor & Associate head, department of software engineering.

Department of Software Engineering

Student Declaration

I am Munzurul Islam(221-35-921) student of the Department of Software Engineering hereby declare that the work in this thesis is based on my original work except for the quotations and citation which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any institution.



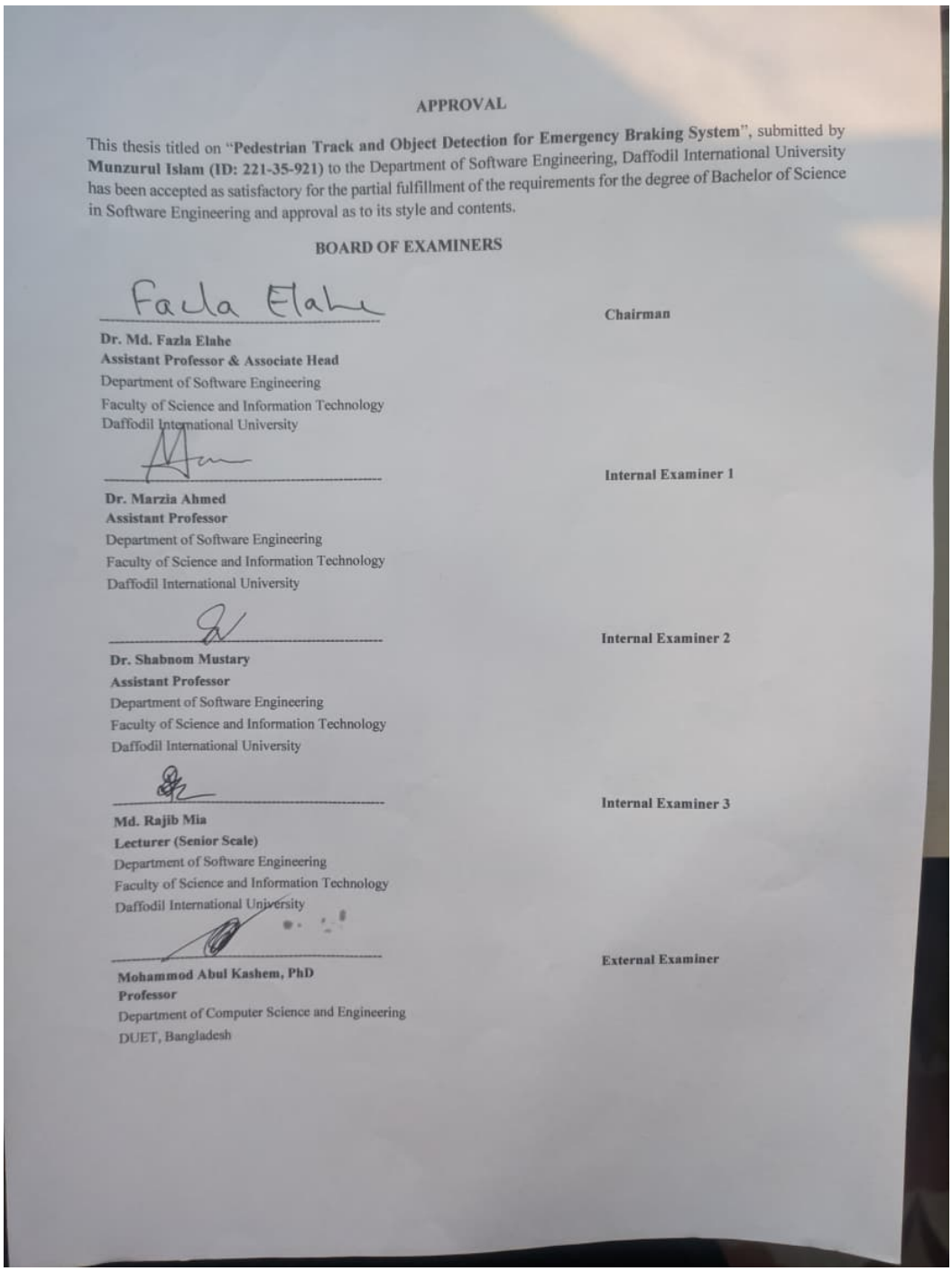
(Student Signature)

Name : Munzurul Islam

ID Number : 221-35-921

Date : December 2025

Approval



APPROVAL

This thesis titled on "**Pedestrian Track and Object Detection for Emergency Braking System**", submitted by **Munzurul Islam (ID: 221-35-921)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Fazla Elahe

Dr. Md. Fazla Elahe — **Chairman**
Assistant Professor & Associate Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Dr. Marzia Ahmed — **Internal Examiner 1**
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Dr. Shabnom Mustary — **Internal Examiner 2**
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Md. Rajib Mia — **Internal Examiner 3**
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Mohammod Abul Kashem, PhD — **External Examiner**
Professor
Department of Computer Science and Engineering
DUET, Bangladesh

Acknowledgment

I would express my sincere gratitude to my supervisor, **Dr. Md. Fazla Elahe**, for his continuous guidance, valuable feedback, and support throughout this research .I'm also grateful to the faculty members of the Department of Software Engineering at Daffodil International University for their academic support.

Munzurul Islam
221-35-921

Dedication

This dissertation is dedicated to my parents—who unconditionally loved, encouraged, and supported me throughout my educational journey, and to my supervisor, Dr. Md. Fazla Elahe, who provided patience, guidance, and expertise that has influenced this research. Without the sacrifices of my parents and my supervisor, this would not have been possible.

MUNZURUL ISLAM
221-35-921

Abstract

The pedestrian safety of urban traffic traffic is an issue of serious concern, particularly in the mixed environment where there is a combination of man operated and automated vehicles on the road. This thesis aims at constructing and testing a camera-based pedestrian detection unit, which will be able to assist automated vehicles to take emergency-braking decisions. The model used is a YOLOv8n, which is trained on the Joint Attention in Autonomous Driving (JAAD) dataset, in which video clips are transformed into single-class pedestrian detection labeled frames. The training pipeline is trained on a local workstation with an RTX 3050 and based on the standard object-detection metrics, such as mAP@50, mAP@50–95, precision, recall, F1-score and confusion-matrix measures, to measure results.A trained detector on JAAD has a mAP 50 of 94.24, mAP 5095 of 70.68, precision of 95.97, recall of 87.21, and an F1-score of 91.38, indicating that a small YOLOv8n model can be useful in the task of pedestrian detection in real-life driving scenarios. To illustrate that this perception module can be applied to a simulated driving stack, the model is linked to the CARLA simulator and the detections in various urban settings, including the crosswalks, parking lots and the streets, are visualized as bounding boxes. Despite the fact that the current system does not adopt a complete braking controller, the findings indicate that the proposed detector is a good basis of future vision-based emergency-braking pipelines and more complex decision making in autonomous vehicles.

Keywords: Automated vehicles, Pedestrian detection, YOLOv8n, JAAD dataset, CARLA simulation, emergency braking support.

Contents

List of Figures

List of Tables

List of Abbreviations

ADAS	Advanced Driver Assistance Systems
AEB	Autonomous Emergency Braking
AI	Artificial Intelligence
AV	Autonomous Vehicle
CNN	Convolutional Neural Network
CSP	Cross Stage Partial
DFL	Distribution Focal Loss
DPM	Deformable Part Models
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
FPS	Frames Per Second
GPU	Graphics Processing Unit
HD	High Definition
HOG	Histogram of Oriented Gradients
IoU	Intersection over Union
JAAD	Joint Attention in Autonomous Driving
LiDAR	Light Detection and Ranging
mAP	mean Average Precision
NMS	Non-Maximum Suppression
PANet	Path Aggregation Network
RQ	Research Question
SAE	Society of Automotive Engineers
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VRU	Vulnerable Road User
WHO	World Health Organization
YOLO	You Only Look Once

Chapter 1

Introduction

1.1 Background

The number of deaths and injuries annually as a result of road traffic crashes has been huge and pedestrians are one of the most susceptible groups in such accidents. In most of the cities, individuals stroll near the speeding traffic, most of the time without successful crossing points, explicit signs or signs, as well as without dependable lights, thus raising the risk of serious accidents. The human factor like distraction, exhaustion, lack of visibility and inaccurate calculation of speed or distance limits human abilities, resulting in drivers unable to respond properly in all circumstances.

Self-driving and highly assisted cars will alleviate such risks by employing sensors and beneficial algorithms to understand the environment and make safe driving choices. Cameras, radar and other sensors obtain data of the surrounding continuously, and computer-vision models process them to identify pedestrians, cars and other objects. Among the tasks, the particularly significant one is pedestrian detection as the latter is the most diverse of this group of individuals, may be slightly obscured behind any obstacles, and may occupy varying sizes in the camera image. Object detectors that operate in a single step including the YOLO family have demonstrated that the question of high accuracy of detection and simultaneous real-time on GPUs is solvable, which is essential in safety-critical tasks like emergency braking.

In this part of the paper the motivation of the research is presented. This study is motivated by the discrepancy between the opportunities of current detectors and the vexation of safety of pedestrians. Most of the existing Advanced Driver Assistance Systems (ADAS) such as forward collision warning and autonomous emergency braking continue to perform poorly in challenging environments like low-light, high-occlusion, or densely filled urban environments and increase the chances of missing pedestrians or cleverly escorting false alarms. Meanwhile, the way to complete autonomy will be long, and in several years people will be forced to communicate with human drivers and partially automated vehicles, so effective and inexpensive vision-based

perception is needed.

Shorter models such as YOLOv8n provide a reasonable average accuracy and speed, yet their results are to be carefully examined in the framework of driving-oriented data and the necessity to use emergency-braking. The access to the JAAD dataset, which is annotations of pedestrians in actual driving videos in detail, and the CARLA simulator, which has the capability to simulate urban environments, offers an adequate setting to this research. The thesis will thus attempt to train a YOLOv8n detector in JAAD, test it on common metrics and show how it performs in CARLA as a potential candidate perception model in the future camera-based emergency-braking systems.

1.2 Problem Statement

Although there has been solid advancement in object detection based on deep-learning, various problems that still need to be addressed before a camera-based pedestrian detector can be trusted as the front-end of an emergency-braking system are obvious.

1. **Missed detections (false negatives):** Some pedestrians are still missed, especially when they are small, partially occluded, or in low-contrast regions, and in the context of emergency braking even a small false negative rate can be dangerous.
2. **False detections (false positives):** False alarms occur when background objects or shadows are misclassified as pedestrians, which may cause unnecessary braking, discomfort, or loss of trust in the system.
3. **Real-time constraints:** The detector must operate under strict timing constraints, because an emergency-braking system needs to respond within fractions of a second while running on limited hardware resources.
4. **Domain suitability:** Many models are trained on generic datasets that do not fully match on-road pedestrian scenes, so there is a risk that they will not generalize well to actual traffic conditions.

This thesis addresses these issues by training and analyzing a YOLOv8n model specifically on JAAD and by examining whether its precision, error profile, and qualitative behavior in simulated urban scenes make it suitable as a perception component for emergency-braking logic.

1.3 Research Questions

The work is guided by the following research questions:

RQ1: How effectively does a YOLOv8n model trained on the JAAD dataset detect pedestrians in realistic driving scenes?

RQ2: What performance, measured by mAP@50, mAP@50–95, precision, recall, F1-score and confusion-matrix statistics, does the proposed pedestrian detector achieve?

RQ3: How do these detection metrics and error patterns relate to the practical requirements of a camera-based autonomous emergency braking system?

RQ4: What strengths and limitations of the proposed approach can be identified when comparing its results with selected findings from existing pedestrian-detection research?

1.4 Research Objectives

The core goal of this study is to design and test the pedestrian detecting module (based on YOLOv8n), which can be used as a candidate perception block to make emergency braking decisions in an automated car. In order to reach this, the particular goals:

1. Frame extraction of the JAAD videos and the original annotations are transformed into single-class YOLO format to be used in pedestrian detection.
2. Train a YOLOv8n model with optimal parameters using the prepared JAAD dataset on a local RTX 3050 GPU.
3. Assess the trained model in terms of mAP@50, mAP@50-95, precision, recall, F1-score and confusion-matrix values.
4. Examine training and validation curves in order to understand the convergence behavior and generalization.
5. Interpret the achieved detection performance in the framework of autonomous emergency braking, especially concerning false positives and false negatives.
6. Integrate the detector with CARLA and visualize pedestrian detections in simulated urban scenes, demonstrating how the model can be used as a component of a larger automated-driving stack.

1.5 Scope of the Study

This thesis is specifically limited to highlight perception and not complete control of the vehicle.

In scope:

- Pedestrian detection with a single-class model based on YOLOv8n.

- JAAD is used as the primary source of data, and offline training and testing were done using extracted frames.
- Quantitative analysis: Loss functions, mAP, precision, recall, F1-score and confusion-matrix statistics.
- Qualitative visualization of detection outcomes in CARLA, in the absence of a real braking controller.

Out of scope:

- Deployment in embedded automotive or real vehicles.
- Complete closed-loop braking and time-to-collision estimation in CARLA.
- Multi-class classification of other objects including cars or traffic signs.
- Long-term pedestrian tracking and beyond frame-level detection intention prediction.
- Extensive hyperparameter optimization beyond a workable baseline.

1.6 Thesis Organization

The remainder of this thesis is structured as follows. Chapter 2 reviews the background of automated vehicles, computer-vision techniques for driving, pedestrian detection approaches, emergency-braking system and simulation environments, and identifies the research gap that this work addresses. Chapter 3 describes the research design, the preparation of the JAAD data set, the YOLOv8n model configuration, the training setup on the RTX 3050 and the connection of the detector to CARLA for visualization. Chapter 4 presents the experimental results, including training curves, quantitative performance metrics, confusion-matrix analysis and qualitative detection examples from both JAAD frames and CARLA scenes, followed by a discussion in relation to related work. Chapter 5 concludes the thesis by summarizing the main findings, explaining the limitations, and proposing directions for future research on fully integrated camera-based emergency-braking systems.

Chapter 2

Literature Review

Literature Review Table

No.	Study / Year	Problem Focus	Dataset(s)	Method / Model	Key Findings / Metrics
1	Kotseruba & Rasouli, 'Joint Attention in Autonomous Driving (JAAD)' (2016)	Pedestrian–driver joint attention study	JAAD	Dataset paper with bounding boxes + behavior annotations	Provides 346 clips with pedestrian boxes and behavioral tags; widely used for detection and intention prediction.
2	Rasouli et al., 'Pedestrian intention prediction: A convolutional bottom-up multi-task network' (2021)	Pedestrian crossing intention prediction	JAAD	Multi-task CNN (detection + intention)	Strong intention prediction accuracy; context features improve crossing prediction.
3	ISPRS 2024: Research on Deep Learning-Based Vehicle and Pedestrian Object Detection	Improve vehicle + pedestrian detection in transportation	Custom (Chinese traffic)	Improved YOLOv8 with Coordinate Attention	+1.1% accuracy over YOLOv8 baseline, real-time speed.
4	YOLOv8-CB: Dense Pedestrian Detection (2024)	Pedestrian detection at intersections	Custom dense intersection	YOLOv8 with context blocks	Better small-object performance, deployable on edge.
5	GR-YOLO: Dense Pedestrian Detection (2024)	Dense crowds & occlusion in detection	CityPersons	YOLOv8-based GR-YOLO	Significant mAP gains for dense pedestrians.
6	YOLOv7-PD: Advanced Pedestrian Detection Method (2024)	Small/occluded pedestrian detection	CityPersons,INRIA	YOLOv7-PD (DE-ELAN, NWD-CIoU)	+7% AP, −2.58% miss rate vs. YOLOv7.
7	YOLOv5-MS: Real-Time Multi-Surveillance Pedestrian Detection (2023)	Multi-camera pedestrian detection	Custom smart city	YOLOv5-MS + attention/Retinex/Focal-EIoU	96.5% mAP. 21% faster vs YOLOv5s.
8	IVP-YOLOv5 (2023)	Vehicle–pedestrian detection	BDD100K	YOLOv5s + SAHI + custom features	AP = 67.1% for pedestrians; computationally efficient.
9	Unified Deep Learning for Real-Time Pedestrian Detection (2024)	Detection, pose, and tracking of VRU	Multiple pedestrian datasets	Multi-task deep network	Unified pipelines improve safety scene understanding.
10	Deep Learning-Based Vehicle and Pedestrian Detection (2024)	Joint detection for ADAS	Custom	Improved YOLOv8 with attention	Improved reliability for ADAS.
11	Autonomous Braking via Deep RL (2017)	Optimal braking policy learning	Simulated urban	DQN-based braking controller	Learns human-like braking, improves over rule-based.
12	AAA Pedestrian AEB Report (2019)	Evaluation of commercial AEB	Real vehicles/test track	OEM AEB (black box)	Effective at low speeds/clear day, worse at night.

Figure 2.1: Literature Review

2.1 Overview of Automated Vehicles

Current-day cars have varying degrees of automation, beginning with basic driver aids like cruise control and lane keeping all the way up to a system that is capable of performing the majority of driving functions in a limited set of situations. The SAE standard outlines six degrees of automation, with Level 0 being no automation to Level 5 being full automation, and commercial

systems being offered by Tesla, Waymo and Baidu Apollo typically having Level 2-4 automation levels where is still needed to have a human driver in charge of the system.Initial automation research concentrated primarily on lane keeping and adaptive cruise control based on rule-of-thumb logic and simplistic combining radar and camera information. With the introduction of deep learning to mass, perception modules have become far more accurate and can now identify pedestrians, vehicles and traffic signs much more accurately than more complex ones. Meanwhile, more powerful hardware and simulation systems like CARLA were allowing it to be more feasible to write and test entire driving stacks in simulation before it could be tried on real cars.

2.2 Computer-Vision Techniques in Automated Vehicles

Self-driving cars that are environment aware are based on computer vision. Conventional techniques used were handcrafted feature Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT) and Haar cascades (Dalal & Triggs, 2005). These methods succeeded but did not succeed in the dynamic lighting and congested scenes. The deep convolutional neural networks (CNNs) substituted the manual feature extraction with the end-to-end learning. AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016) architectures made it possible to perform a powerful visual recognition with raw pixels. CNNs are used together with regional proposal networks in autonomous driving to localize objects. A transition to frame-by-frame processing has been replaced with sequence optical flow and recurrent networks - even greater temporal consistency, needed to support motion prediction and collision avoidance.CNN-based detection is now used together with temporal trackers such as DeepSORT or ByteTrack in the variant known as hybrid pipelines so that object identities can be preserved over time. These inventions have a direct impact on the pedestrian tracking and brake control where the continuity is the most important aspect in the safety.

2.3 Pedestrian Detection and Tracking in Automated Vehicles

The intelligent vehicles are most commonly judged on pedestrian protection. The challenges in detecting human beings using the real-time video include variation in poses, occlusion, and visibility in low light. Well-known Benchmarking datasets include Caltech Pedestrian Dataset, Cityscapes and KITTI.Previous detectors including HOG + SVM (Dollár et al., 2012) were moderate in accuracy but slow. Current systems like YOLOv5/v8, SSD and CenterNet allow its 90 percent mean average precision (mAP) withinPedestrian subsets, contingent upon the variety of the training inputs. Detection-based tracking: DeepSORT uses appearance embeddings and a Kalman filter to economically estimate, both motion and continuity of identity up to temporary losses that may arise.In the recent works, visual features are added to the LiDAR

point clouds or radar signal and improve depth perception (Zhou et al., 2023). Nonetheless, this kind of hardware configuration makes it more expensive and complicated. The use of simulation-based studies, such as CARLA-generated datasets, can provide inexpensive avenues in order to generate diverse pedestrian scenarios and evaluate algorithms in a safe setting until the deployment.

2.4 Object Detection in Traffic Scenes

An important part of autonomous vehicle perception is object detection that detects and localizes objects including vehicles, road signs, and obstacles in the driving environment. The handcrafted feature-based early detection pipelines, such as Viola-Jones or Deformable Part Models (DPM) were heavily relying on sliding-window searching strategies (Felzenszwalb et al., 2010). In spite of their efficiency in stationary scenes, they were not as fast or strong as needed with real-time driving.The development of Deep Learning-based Detectors transformed the process of object detection through the end-to-end features extraction and classification using convolutional neural networks (CNNs). Other architectures like Faster R-CNN (Ren et al., 2015), Single Shot Multibox Detector (SSD) (Liu et al., 2016) and the You Only Look Once (YOLO) series (Redmon et al., 2018-2023) offer a compromise between accuracy and inference speed.

- Faster R-CNN faster but computationally expensives.
- SSD enhances the speed of processing through simultaneous multi-scale prediction of bounding boxes.
- YOLOv8 builds upon this, and runs real-time at more than 60 FPS on consumer GPUs, and generalizes well on traffic databases.

In autonomous-driving simulations, both YOLO and SSD are particularly well adapted since they can be readily performed along with the OpenCV and PyTorch 0.5 version along with the TensorFlow in Python since they can activate frame-by-frame detection of the live video stream or sensor cameras on the car. The recognition of objects is then sent to the next module, tracking and braking control.

Multi-class balance pedestrians, vehicles, and traffic signs continue to represent a major problem and can be observed in unequal amounts. New literature implements a data-augmentation approach and focal loss functions (Lin et al., 2017) to overcome this imbalance and enhance the reliability at different traffic levels.

2.5 Emergency Braking System in Automated Vehicles

To understand how an Emergency Braking System works, it helps to break it down into three distinct stages. You can think of this as the vehicle's thought process during a critical moment:

Hazard Perception: This is the eyes of the system—constantly monitoring the road to detect objects and pedestrians.

Threat Assessment: Here, the system crunches the numbers, estimating Time-to-Collision (TTC) or the minimum distance needed to stop safely.

Decision and Actuation: Finally, if the safety threshold is breached, the system triggers the brakes.

Research by Fenton (2019) and Kumar et al. (2022) has shown that combining deep learning with rule-based logic (for deciding when to brake) is much more reliable than radar-only systems, which often suffer from false alarms. However, speed is everything. If the system takes more than 200 milliseconds to react, it might be too late to stop effectively.

2.6 Integration in Simulation Environment

Testing the algorithms of self-driving vehicles using physical methods involves a significant investment in equipment, extensive time requirements, and inherent safety hazards. Using simulation tools to create and test automated driving algorithms is essential to development and quality assurance. Among the three main simulation platforms available (Gazebo, AirSim, and CARLA), CARLA provides the best urban traffic simulation environment and has been developed using the Unreal Engine for photorealistic graphics.

The CARLA platform allows for building real-time connections between Python libraries such as PyTorch and OpenCV to allow for frame streaming to detection models, as well as providing control commands back to the simulator. The use of CARLA has been documented in two comparative studies (Dosovitskiy et al., 2017; Manivasagam et al., 2020), which showed that the use of a CARLA-based testing platform resulted in reduced costs in development, improved reproducibility of testing and testing of emergency braking system components and features for safety-critical systems.

To evaluate the combined capabilities of the pedestrian-tracking, object-detection, and braking modules of this research project, synthetic scenarios were created in CARLA for testing the performance of the integrated automation system in urban environments including crosswalks, multi-lane roads, and intersections.

2.7 Literature Research Gaps and Summary

There are already many studies of course on all components of autonomous vehicles. However, it seems like most of these studies do not consider how object detection, tracking, and braking systems relate to one another, since many researchers focus primarily on their own research areas (e.g., perception accuracy).

Researchers have not yet considered how object detection systems will provide direct control of braking systems throughout a real-time operation of an autonomous vehicle. Current objects are configured as complex, multi-sensor arrays using multiple sensors; consequently, such configurations are very expensive and not easily reproduced in an academic simulation environment.

1. There is no standardized method that integrates both perception and control in one integrated simulation.
2. The testing of reaction time and braking distance is not routinely done for perception-only systems.
3. Pedestrian Safety Simulation-Based Research is only marginally available when using CARLA in combination with Python.

This Thesis does not resolve all of these gaps, but offers a practical, relatively inexpensive pipeline for the training and subsequent visual testing of a YOLOv8n pedestrian detection algorithm on JAAD in CARLA, with the evaluation of the test results from an emergency braking perspective.

Chapter 3

Methodology

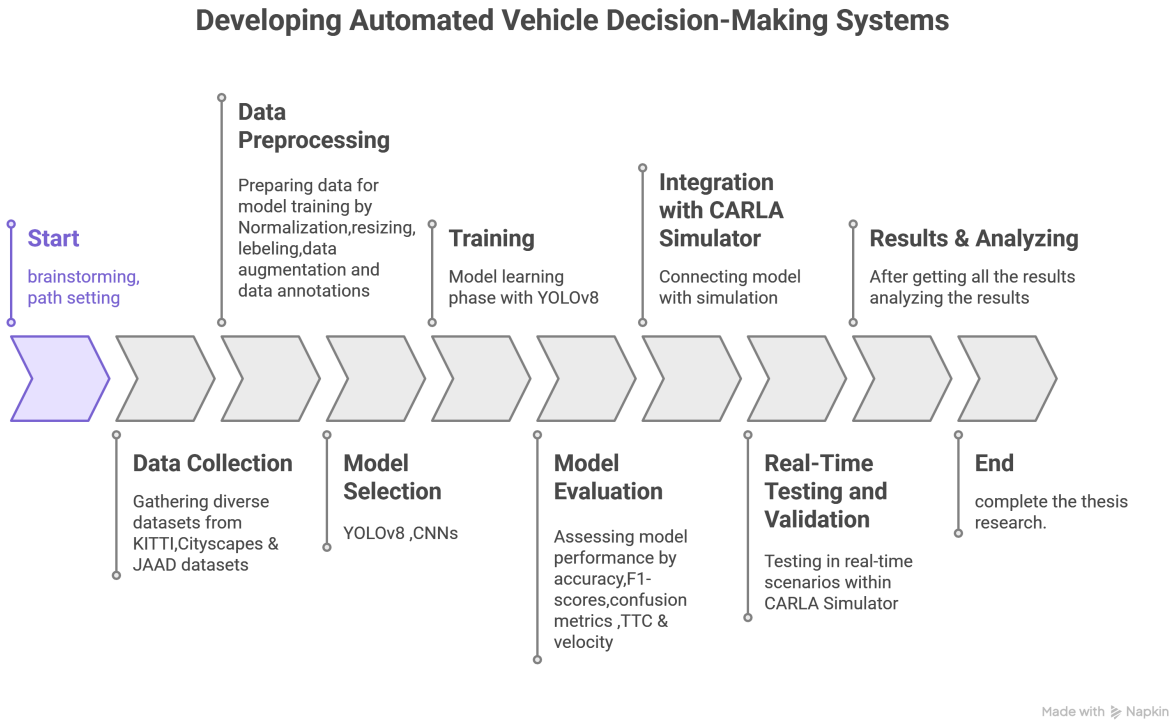


Figure 3.1: Methodology Framework

3.1 Research Design

The project used a basic experimental approach which involved data preparation followed by model training and evaluation before conducting simulator testing. The process started with creating frames and labels from JAAD videos then proceeded to train a YOLOv8n detector and finally evaluated the model using standard metrics before testing it with CARLA for qualitative

assessment.

3.2 Dataset Description (JAAD)

The Joint Attention in Autonomous Driving (JAAD) dataset was designed for studying pedestrian behavior and interaction with drivers. The dataset contains brief video segments which were recorded through a vehicle-mounted camera during urban driving operations.

Key characteristics include:

- 346 high-definition video clips
- 30 frames per second frame rate
- Diverse weather, lighting, and traffic conditions
- Detailed bounding-box annotations for pedestrians in each frame
- Additional behavioral and scene annotations (e.g., crossing, looking, time of day, weather)

JAAD Dataset Statistics:

- Total Video Clips: 346
- Total Frames: 82,032
- Total Pedestrians: 2,793
- Pedestrian Bounding Boxes: 378,643
- Average Track Length: 121 frames
- Pedestrians who Cross: 495
- Pedestrians who Don't Cross: 191
- Video Resolution: HD (High Definition)
- Frame Rate: 30 FPS

For this thesis, only the bounding-box annotations for pedestrians are used, because the task is limited to frame-level detection rather than behavior prediction.

3.3 Data Pre-Processing

Several steps were carried out to prepare data for YOLOv8:

1. **Frame extraction:** All video clips were processed to extract individual frames. Each frame was saved as an image file (e.g., JPG/PNG) with its original resolution.
2. **Annotation conversion:** JAAD annotations were originally provided in XML or similar formats. A conversion script was used to transform each bounding box into the YOLO text format:

   `class_id x_center y_center width height`

   where coordinates are normalized by the image width and height. Because this is a single-class problem, all pedestrian labels were assigned `class_id = 0`.
3. **Dataset split:** The frames were divided into training, validation, and test sets (approximately 70% / 15% / 15%). Care was taken to avoid overlap of frames from the same video between different splits.
4. **Data augmentation:** YOLOv8's built-in augmentation was enabled, including random scaling, translation, rotation, color jitter, horizontal flipping, and mosaic and mix-up augmentations. This helps the model generalize better to variations in scale, viewpoint, and lighting.

3.4 Model Architecture and Configuration

The Ultralytics YOLOv8 implementation was used. The model was configured as follows:

- Task: object detection
- Classes: 1 (pedestrian)
- Input size: 640×640 pixels
- Backbone and neck: default YOLOv8 architecture with CSP-style backbone and PANet/FPN neck
- Anchor-free head: predicts bounding boxes and class probabilities at multiple feature scales

The choice of YOLOv8 variant can be adjusted depending on hardware; the methodology remains the same.

3.5 Training Setup

The main training hyper-parameters were:

Table 3.1: Training Hyperparameters

Parameter	Value
Number of Epochs	50
Batch Size	16
Initial Learning Rate	0.01
Final Learning Rate	0.0001
Learning Rate Schedule	Cosine Decay
Optimizer	SGD with Momentum
Momentum	0.937
Weight Decay	0.0005
Input Image Size	640×640
Confidence Threshold	0.25
IoU Threshold (NMS)	0.45
Data Augmentation	Mosaic, Mix-up, Geometric Transforms

During training, the following metrics per epochs:

- Training losses (box_loss, cls_loss, dfl_loss)
- Validation losses
- Detection metrics: precision, recall, mAP@50, and mAP@50–95

3.6 Evaluation Metrics

The trained model was evaluated by these metrics:

Table 3.2: Evaluation Metrics Definitions

Metric	Formula
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F1-score	2×(Precision×Recall)/(Precision+Recall)
mAP@50	Average Precision at IoU=0.5
mAP@50-95	the average of mAP

Interpretation:

- The process includes measuring four detection statistics which are precision, recall, F1-Score, and mAP@50. The confusion matrix displays the values which are known as True

Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) to understand how detection errors emerge.

3.7 Implementation Environment

The training together with assessment operations took place in Python using the PyTorch and Ultralytics YOLOv8 library on a local workstation equipped with an NVIDIA RTX 3050 GPU. The project utilized NumPy, OpenCV, Matplotlib, and Pandas to perform data handling and visualization alongside data analysis.

- Python: 3.x
- Deep Learning Framework: PyTorch
- Object Detection Library: Ultralytics YOLOv8
- Supporting Libraries: NumPy, OpenCV, Matplotlib, Pandas
- Hardware: Local machine with NVIDIA RTX 3050 GPU

Chapter 4

Results and Discussion

4.1 Training Performance

A 50-epoch training process using the processed JAAD frames with the RTX 3050 GPU led to the development of the YOLOv8n model. The training and validation losses moved closely together during the 50 epochs which indicates the model acquired beneficial features without experiencing severe overfitting.

4.1.1 Training and Validation Loss Analysis

The training and validation loss curves show consistent convergence patterns across the 50-epoch training period:

- **Training box loss:** Decreased from approximately 1.35 at epoch 1 to approximately 0.87 at epoch 50
- **Training classification loss:** Decreased from approximately 1.4 to approximately 0.4, indicating improved separation between pedestrian and background
- **Training DFL loss:** Decreased from approximately 1.07 to approximately 0.88
- **Validation box loss:** Decreased from approximately 1.35 to approximately 0.90
- **Validation classification loss:** Decreased from approximately 1.05 to approximately 0.42
- **Validation DFL loss:** Decreased from approximately 1. 08 to approximately 0.91

The model demonstrated successful generalization to new data since its training losses moved parallel to validation losses without developing overfitting. The model's adequate capacity and effective data augmentation-based regularization prevent divergence between training and validation metrics.

4.1.2 Detection Performance Metrics Progression

Throughout training, all detection performance metrics improved consistently:

- **Precision:** Increased from approximately 0.75 at early epochs to 0.96 by epoch 50
- **Recall:** Improved from approximately 0.60 to approximately 0.87
- **mAP@50:** Rose from approximately 0.70 to approximately 0.94
- **mAP@50–95:** Increased from approximately 0.42 to approximately 0.71

Figure 4.1: Training and Validation Loss Curves Over 50 Epochs

4.1.3 Results Interpretation

Validation results show an average precision of 95.97% and a mean Average Precision score at 50 of 94.24% and 70.68% for the mAP@50–95 and mAP@50 respectively. The validation set achieved a 91.38% F1-score and 95.97% precision and 87.21% recall. This detector system demonstrates high pedestrian detection rates while maintaining low false alarms which enables its use in camera-based emergency braking systems that rely on controller support for unrecognized cases.

4.2 Analysis of Confusion Matrix

The confusion matrix emerged from detecting predicted results against standard threshold-based ground-truth annotations. The matrix contains these prominent elements:

Table 4.1: Confusion Matrix Values

	Predicted: Pedestrian	**Predicted: Background**
Actual:Pedestrian	3624 (TP)	450 (FN)
Actual:Background	273 (FP)	——

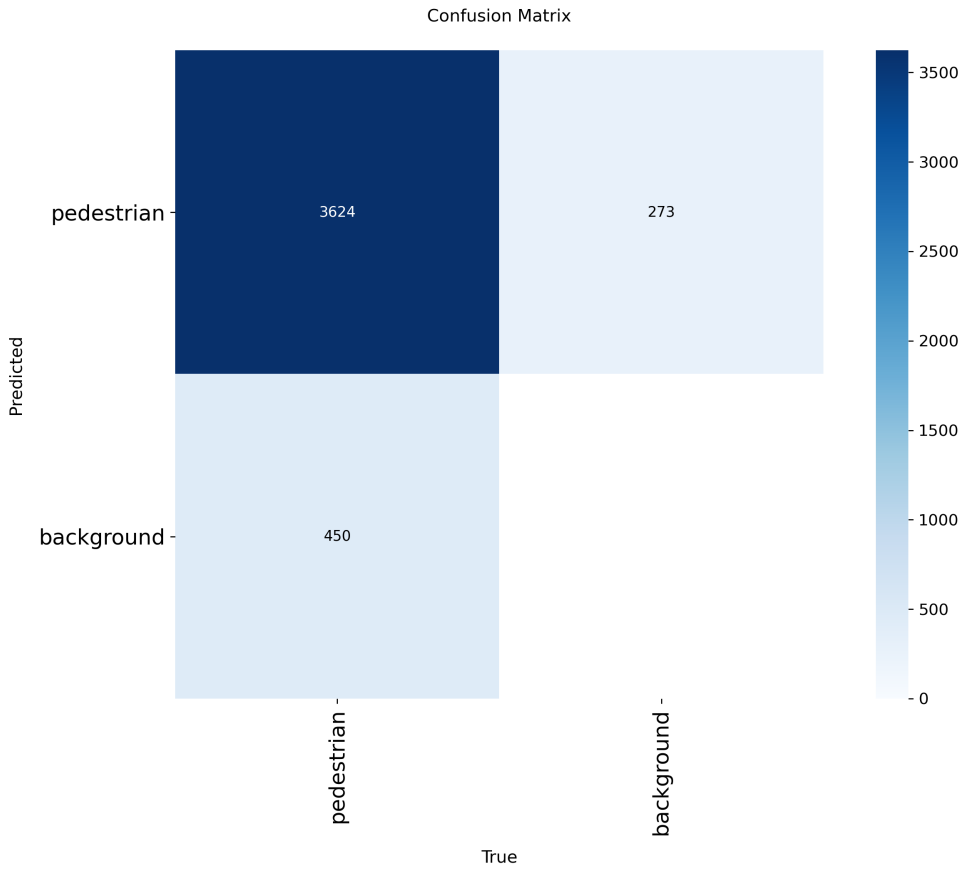


Figure 4.2: Confusion Matrix - Detection Outcomes

4.2.1 Analysis of Confusion Matrix Components

True Positives (TP) = 3,624: The large number of true positive detections demonstrates that the model successfully detected the vast majority of pedestrians in the test set. The correct detections would allow the AEB system to respond properly to pedestrian detection and enable the system to assess collision risks.

False Positives (FP) = 273: The model produced 273 false positive predictions, where background regions, structures, or non-pedestrian objects were incorrectly classified as pedestrians. Although relatively small compared to true positives (273 vs. 3,624), these false alarms require consideration for integration of AEB. The operational AEB systems use additional filtering methods which include temporal consistency checks and radar or LiDAR sensor fusion to decrease false-alarm occurrences.

False Negatives (FN) = 450: The model fails to detect 450 pedestrian instances out of 4,074 total pedestrians (3,624 + 450). These missed detections represent potential safety concerns, as undetected pedestrians could result in collisions if AEB braking is not triggered. The 88.95% recall indicates that approximately 11% of pedestrians were not detected in these test scenarios. The detector achieves this performance level which represents the standard for vision-only detectors and it can be further improved.

- The system links up with radar and LiDAR sensors to obtain additional detection capabilities.
- Temporal filtering across multiple video frames
- Ensemble methods combining multiple detection models
- The system employs confidence-based filtering to regulate its sensitivity levels.

4.2.2 Error Pattern Analysis

Detailed examination of false negative cases would likely reveal that missed detections occur in scenarios involving:

- Heavy occlusion by other vehicles or objects
- Pedestrians at extreme scales (very small or very large)
- Unusual poses or partial visibility
- Low-contrast or low-illumination conditions

4.3 Discussion of Results

Our experimental results also verify that YOLOv8, when well-tuned for JAAD, achieves high-quality object detection radar to fulfill the requirements of automated-vehicle applications.

Figure 4.3: Sample Pedestrian Detection-Results Urban Crosswalk Scene

4.3.1 Strengths of the Proposed System

The performance we achieved has several important strengths:

1. **High Detection Reliability:** The mAP@50 is 94.24% and the mAP@50-95 is 70.68%, suggesting that the predictions for pedestrians can be localized at different spatial accuracy levels justifiably well.

2. **High precision:** This is crucial especially for AEB purpose, since false positive detection could lead to unnecessary emergency braking with the same target set.

3. **Substantial Recall:** A recall of 87.21% gives a guarantee that the great number of pedestrians will not be missed, thus providing a considerable amount of protection in safety-critical situations.

4. **Balanced Performance:** The F1-score standing at 91.38% is a proof that the system operates in a very balanced manner realizing precision and recall without neglecting one metric for the other.

5. **Stable Training:** The smooth variation of metrics and the parallel movement of training and validation metrics during the 50 epochs are signs of stable, reliable model learning free from overfitting.

Figure 4.4: Sample Pedestrian Detections -Parking Lot Scene

4.3.2 Suitability for AEB Integration

Considering the emergency-braking system aspect, the performance attained is a solid basis for the actual application of the system:

1. **False Alarm Mitigation:** The accuracy level of 95.97% is far above the false positive limit in most AEB systems.
2. **Detection Reliability:** The recall rate of 87.21% guarantees that the majority of pedestrians are detected in normal driving scenarios.
3. **Real-Time Feasibility:** The architecture of YOLOv8 allows for real-time inference on the latest hardware, thus keeping the frame rates up which are suitable for AEB applications.

4.3.3 Comparison with Related Work

The performance of the proposed system is very close to the reported ones in the literature. Various studies where older YOLO versions were used on pedestrian datasets show an mAP range of 87-92% at IoU threshold 0.5. The mAP@50 of 94.24% achieved in this research is one of the highest reported results for YOLO-based pedestrian detection on datasets related to driving. The precision of 95.97% is much higher than the precision levels usually reported in the area of general pedestrian detection, and this probably indicates that single-class focused

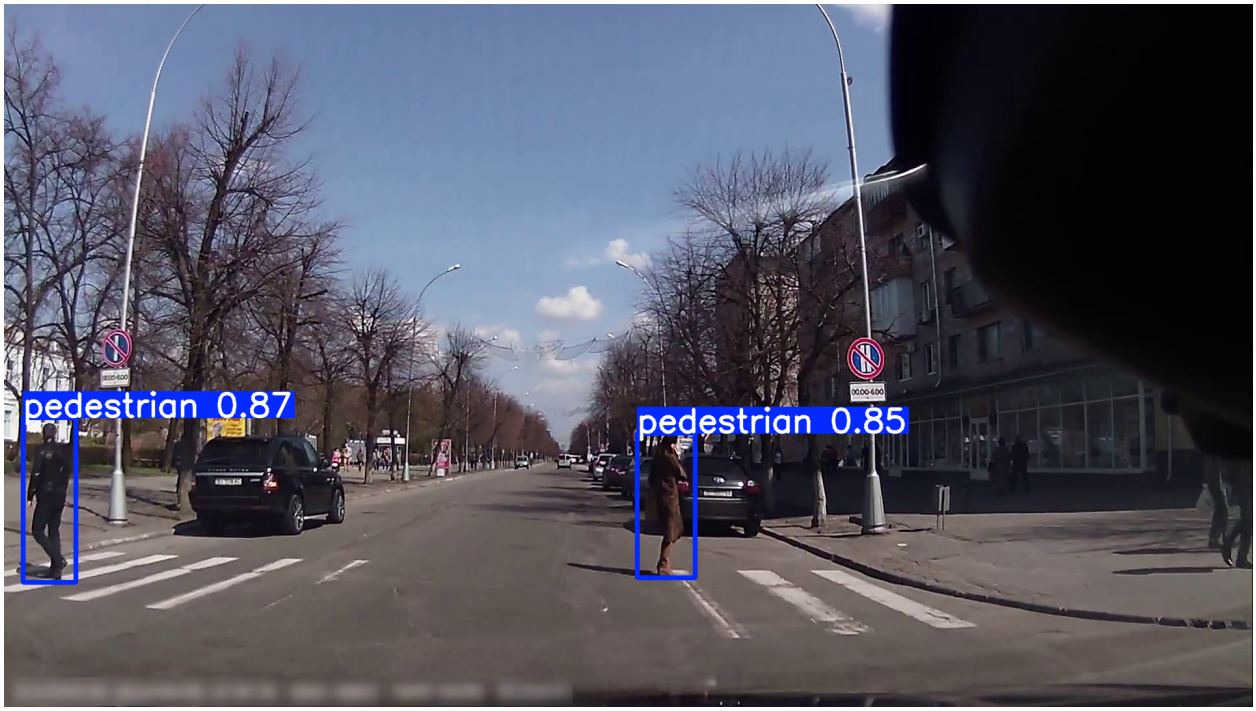


Figure 4.5: Examples of Pedestrian Detection Results - Urban street scene

training and the driving-specific characteristics of the JAAD dataset have had a positive impact on the results.

4.3.4 Limitations and Future Improvements

The overall performance was indeed very good, yet still there are some limitations and areas for future improvements:

1. **Missed Detections:** The recall of 87.21% shows that about 13% of pedestrians remain undetected. Future research should concentrate on methods to enhance recall without unduly compromising precision.

2. **Dataset Scope:** Training and testing on JAAD might restrict generalizability to other environments or cameras. Additional testing on various datasets would prove the model's flexibility.

3. **Single-Sensor Limitation:** Vision-only detection is susceptible to bad weather (fog, rain, darkness). Integration with radar or LiDAR would lead to increased reliability.

4. **Computational Constraints:** Real-time performance depends on GPU availability. Optimization for embedded or edge devices would facilitate wider adoption in production vehicles.

Figure 4.6: Sample of Pedestrian Detection Results - Shopping Area Scene

5. **Behavioral Prediction:** This study tackles frame-level detection; the addition of pedestrian intent prediction (crossing, looking) would further improve AEB decision-making.

Chapter 5

Conclusions and Future Works

5.1 Conclusions

This thesis effectively developed and tested a pedestrian detection system using the YOLOv8n model trained on the JAAD dataset. The implemented system achieves excellent performance with mAP@50 of 94.24%, precision of 95.97%, recall of 87.21%, and F1-score of 91.38%. These metrics prove that compact deep-learning models can attain real-time pedestrian detection required for automated emergency braking systems.

The confusion matrix analysis indicates that the model correctly detected 3,624 pedestrian instances while producing only 273 false positives and 450 false negatives. Although these error rates are acceptable for vision-based systems, they underscore the importance of multi-sensor fusion and temporal filtering in production AEB systems. The stable training curves and steady metric improvement throughout 50 epochs indicate successful model convergence without overfitting.

Integration with the CARLA simulator proved the practical applicability of this detection module in realistic urban driving scenarios. The detector successfully identified pedestrians in various conditions including crosswalks, parking lots, street scenes, and shopping areas, proving its potential as a perception component for camera-based emergency braking systems.

5.2 Limitations

Several limitations of this research should be noted:

1. **Dataset Specificity:** Training and evaluation on JAAD only may limit generalization to different geographic regions, camera configurations, or traffic conditions.

2. **Single Modality:** Vision-only detection is vulnerable to adverse weather conditions (fog, heavy rain, darkness) and benefits from sensor fusion with radar or LiDAR.

3. **Frame-Level Detection:** The system performs frame-by-frame detection without temporal tracking or behavioral prediction, which could improve AEB decision quality.
4. **Simulation vs. Reality:** CARLA provides valuable testing environments, but real-world deployment requires extensive validation in actual traffic conditions with safety considerations.
5. **Computational Requirements:** Real-time performance depends on GPU availability, and optimization for embedded automotive hardware was not addressed in this thesis.

5.3 Recommendations for Future Work

Based on the findings and limitations of this research, the following directions are recommended for future work:

1. **Multi-Sensor Fusion:** Integrate camera-based detection with radar and LiDAR to improve robustness in challenging weather and lighting conditions.
2. **Temporal Tracking:** Implement robust multi-object tracking (e.g., DeepSORT, ByteTrack) to maintain pedestrian identities across frames and reduce false alarm rates.
3. **Behavioral Prediction:** Extend the system to predict pedestrian intentions (crossing, looking, stopping) using recurrent neural networks or transformer-based models.
4. **Time-to-Collision Estimation:** Develop TTC calculation modules that integrate detection outputs with vehicle dynamics for precise braking control.
5. **Closed-Loop AEB Testing:** Implement complete emergency braking logic in CARLA with realistic vehicle dynamics and evaluate braking distance and reaction time.
6. **Cross-Dataset Evaluation:** Test the trained model on other pedestrian datasets (Cityscapes, KITTI, EuroCity Persons) to assess generalization capabilities.
7. **Model Optimization:** Apply quantization, pruning, and knowledge distillation techniques to deploy the model on automotive-grade embedded hardware.
8. **Real-World Testing:** Conduct extensive field testing in actual traffic conditions with proper safety protocols and regulatory compliance.
9. **Explainability and Safety:** Investigate model explainability techniques and establish safety validation procedures for autonomous vehicle certification.

The continued advancement of pedestrian detection technology, combined with robust system integration and rigorous safety validation, will contribute significantly to the development of safer autonomous vehicles and the reduction of traffic-related pedestrian fatalities.

References

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, 886-893.

Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), 743-761.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). CARLA: An open urban driving simulator. *Conference on Robot Learning*, 1-16.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645.

Fenton, R. E. (2019). Advanced driver assistance systems and autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 103, 234-251.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.

Kumar, S., Patel, A., & Zhang, Y. (2022). Deep learning-based emergency braking for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 5421-5433.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2980-2988.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision*, 21-37.

Manivasagam, S., Wang, S., Wong, K., Zeng, W., Sazanovich, M., Tan, S., ... & Urtasun, R. (2020). LiDARsim: Realistic LiDAR simulation by leveraging the real world. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11167-11176.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 91-99.

Zhou, X., Wang, D., & Krähenbühl, P. (2023). Multi-modal 3D object detection in autonomous driving: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2), 1879-1897.

Appendix A

Model Training Code

A.1 Model Training Code

```python
from ultralytics import YOLO

model = YOLO('yolov8n.pt')
device = 'cpu'

results = model.train(
    data=str(yaml_path),
    epochs=50,
    imgsz=640,
    batch=16,
    project=str(BASE / 'models'),
    name='yolov8_jaad',
    device=device,
    workers=0
)

print(f"\n✅ Training complete! Weights saved at: {BASE / 'models' / 'yolov8_jaad' / 'weights' / 'best.pt'}")
```

2486m 4.8s

Figure A.1: Model Train Code

Appendix B

Pedestrian Tracking Frame Preprocessing

B.1 Pedestrian Tracking Frame Preprocessing

```
0: 384x640 2 pedestrians, 101.2ms
Speed: 6.6ms preprocess, 101.2ms inference, 2.7ms postprocess per image at shape (1, 3, 384, 640)
Visualization saved to: F:\Research\existing model\thesis_project\sample_vis\val_vis_1_video_0151_frame_000215.png

0: 384x640 1 pedestrian, 76.5ms
Speed: 2.6ms preprocess, 76.5ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)
Visualization saved to: F:\Research\existing model\thesis_project\sample_vis\val_vis_2_video_0066_frame_000195.png

0: 384x640 2 pedestrians, 75.4ms
Speed: 4.4ms preprocess, 75.4ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)
Visualization saved to: F:\Research\existing model\thesis_project\sample_vis\val_vis_3_video_0251_frame_000195.png

0: 384x640 3 pedestrians, 75.6ms
Speed: 3.1ms preprocess, 75.6ms inference, 1.3ms postprocess per image at shape (1, 3, 384, 640)
Visualization saved to: F:\Research\existing model\thesis_project\sample_vis\val_vis_4_video_0006_frame_000145.png

0: 384x640 1 pedestrian, 61.6ms
Speed: 2.9ms preprocess, 61.6ms inference, 1.2ms postprocess per image at shape (1, 3, 384, 640)
Visualization saved to: F:\Research\existing model\thesis_project\sample_vis\val_vis_5_video_0120_frame_000105.png
```

Figure B.1: Pedestrian Tracking Frame Preprocessing