Detection of Dangerous Driving Behaviors Using Multi-Dimensional Data-Driven Methodology

Bao Ma

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

July 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Bao Ma**

Entitled: **Detection of Dangerous Driving Behaviors Using Multi-Dimensional Data-Driven Methodology**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Name of the Chair

______________________________ External Examiner
Dr. Name of External Examiner

______________________________ Examiner
Dr. Name of Examiner One

______________________________ Supervisor
Dr. Subhash Rakheja

______________________________ Co-supervisor
Dr. Hamid Taghavifar

Approved by ______________________________________
Martin D. Pugh, Chair
Department of Mechanical and Industrial Engineering

____________ 2025

____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Detection of Dangerous Driving Behaviors Using Multi-Dimensional Data-Driven Methodology

Bao Ma

Transportation technologies are currently experiencing rapid advancements in the context of global development. The rapid increases higher in traffic volume, however it has contributed to higher, and severity of traffic are continuously rising. The preservation of human lives and attributed to road traffic accident has become vital worldwide and the public is an urgent requirement. However, owing to frequent and widely diverse dangerous driving behaviors of drivers, there are significant potential road safety risks, which will seriously affect the healthy development in the transportation industry. To mitigate such safety hazards, there is an urgent need for precise detection and warning of dangerous driving behaviors of human drivers under driving condition. This dissertation research focuses on a multi-scale data-driven method for detecting drivers' dangerous driving behaviors. The multi-dimensional data basis provided the essential providing theoretical basis and technical support platform is proposed for formulating of a comprehensive driver warning system.

The main work of this paper includes the following five parts:

(1) Analyzing the danger of a driving behavior and its relationship with traffic accidents from the perspectives of public safety and system engineering is developed considering a visual recognition detection system architecture for dangerous driving behaviors based on using the reported research methods, and deep learning neural network models and detection algorithms used for dangerous driving behavior recognition, laying the theoretical foundation for subsequent research.

(2) An efficient identification method is subsequently proposed for dangerous driving behaviors based on the improved YOLOv8 (You Look Only Once version 8) neural network platform. To improve the recognition efficiency and accuracy of dangerous driving behavior detection, a Multi-Head Self-Attention (MHSA) attention mechanism module is further adopted to enhance efficiency

and accuracy, enabling the model focus on global target within a larger receptive field, thereby enhancing the recognition capability of targets. This global modeling capability helps reduce false positive and false negative rates in target detection, improving the overall performance and robustness of the model. Meanwhile, inserting a driver's driving emotion detection module in the network layer shares ROI (Region Of Interest) features in the target detection algorithm, effectively increasing the detection of driver's driving emotion recognition function without significantly increasing the complexity of network computation demand.

(3) Proposing a method for detecting driver's brake pedal and accelerator pedal operations using the Mask-RCNN instance segmentation deep learning network. It recognizes and evaluates the operation of driver's brake pedal and accelerator pedal through video frames collected using a driver's leg camera. The use of ROI Align method to effectively align pixels during downsampling enhances the feature extraction capability of the entire detection network with increasing the computational efficiency and complexity of the network model, thereby improving the efficiency and accuracy of the detecting driver's brake pedal and accelerator pedal operations.

(4) A data-driven detection method is subsequently proposed using the filtering and sliding windows. By collecting brake signals and throttle signals from the vehicle CAN bus data and performing low-pass and median filtering, the computational feature extraction capability of the driving signal data is enhanced. The sliding window method compares and judges brake signals and throttle signals in the sampling interval with the corresponding thresholds, achieving detection and evaluation of driver's dangerous driving behaviors such as sudden acceleration and deceleration.

(5) Building a dangerous driving behavior detection system, integrating neural network training weights and detection algorithms through Python and Qt Designer software systems, and designing a visual real-time detection front-end UI interface. At the same time, designing real vehicle verification experiments, based on real-time experiments in actual driving environments, to verify the effectiveness and superiority of the driver's dangerous driving behavior detection method proposed in this study, providing theoretical basis and technical support for the widespread application of dangerous driving behavior recognition technology.

Keywords: Dangerous driving behavior; Deep learning; YOLOv8; Mask-RCNN; Sliding window method

Acknowledgments

First and foremost, I would like to express heartfelt greatitude to my supervisors, Prof. Subhash Rakheja and Prof. Hamid Taghavifar. I would like to thank the two professors for giving me the opportunity to study and live in a place 13,000 kilometers away from my hometown. I admire Prof. Rakheja's persistence in scientific research for more than 40 years and Prof. Taghavifar's patient answers and serious work attitude. They set an example for me and let me know the direction of my efforts. I am very grateful to the two professors for their patient guidance on my thesis, which has benefited me a lot. What I need to thank more is that the two professors are willing to give me the opportunity to continue to engage in scientific research so that I can continue my PhD degree here. In addition, I would like to thank a lady named Sarita. In my life in a foreign country, Ms. Sarita gave me meticulous care in life. I still remember that when I first came here, she drove to pick me up I couldn't take the metro and always cared about my daily life. This gave me great confidence to survive in this strange country. I would also like to thank all the friends I met here, Masih, Abolfazl, Jules, Zhipeng, Negar, Ashwathy, Keywan, Neda, Erfan, because of them, my life has become colorful.

Furthermore, I want to thank my parents. I never thought about studying abroad. I always felt that tuition and living expenses were expensive, but my parents always encouraged me and gave me confidence. They saved money for my academic development and used their savings for many years to send me to Concordia University to continue my studies. I will keep working hard with their expectations until one day I can take good care of them.

Yang Jiang said: Flowers have their blossoming periods, and people have their fortunes at different times. You should make an effort, but do not be anxious. Both blooming like a brocade and bearing abundant fruit require a process. I will continue to learn from Prof. Subhash Rakheja and Prof. Hamid Taghavifar and spread my wings in the sky of scientific research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

In recent years, the rapid developments in economy and technology, the automotive industry has undergone significant advancements leading to driver assist technologies for enhancing road safety. Automatics cars have also become an integral part of people's daily lives. The sharp increases in the volume of vehicles, however, have contributed to numerous road safety concerns worldwide. The frequent occurrences of traffic accidents have contributed to unreasonable losses in human life, prompting more and more countries and governments to pay closer attention to traffic safety issues (Kampen et al., 2022).

Figure 1.1 illustrates the estimated trend in global road traffic fatalities from 2000 to 2021, alongside an insert showing the change in road traffic fatality rates per 100,000 population from 2010 to 2021 (World Health Organization, 2023). Overall, despite the continuous growth in global vehicle ownership and the expansion of transportation infrastructure, the number of road traffic fatalities has remained relatively stable. This trend is particularly evident during the "Decade of Action for Road Safety 2011–2020," a period focused on enhancing road safety through legislation, education, and enforcement efforts aimed at reducing traffic injuries and fatalities. While some progress has been made, the overall data indicates that this initiative did not lead to a substantial reduction in global road traffic fatalities. This suggests that there remains significant potential for more efforts in reducing traffic related deaths and enhancing road safety measures.

The inset chart presents the trend in road traffic fatality rates per 100,000 population from 2010 to 2021. The data show an overall decline in fatality rates, gradually decreasing from a higher level in 2010 to a lower level in 2021. This indicates that, the implementations of global road safety policies and stronger enforcement of traffic regulations, the risk of fatalities can be reduced. It is worth noting that this declining trend contrasts with the stable pattern observed in the total number of global road traffic fatalities. This discrepancy may be attributed to factors such as population growth and increased vehicle ownership, which offset the reductions in individual risk levels. Therefore, continued efforts to mitigate high-risk driving behaviors remain a critical focus for future global road safety initiatives.

This work focuses on detecting dangerous driving behaviors of drivers. The keys of motivations, objectives and an outline of this thesis are summarized in this chapter.

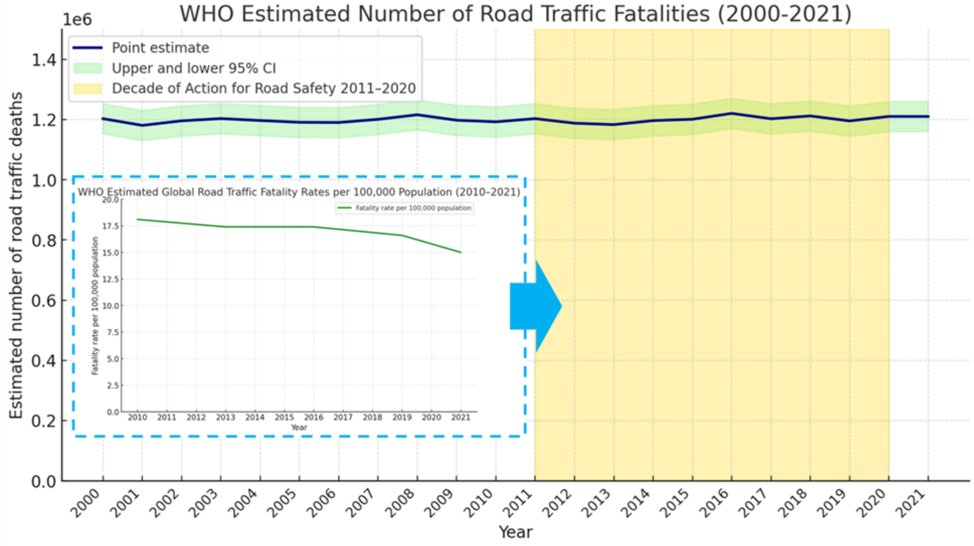


Figure 1.1: Global Road Traffic Fatalities and Fatality Rates per 100,000 Population from 2001 to 2021 reported by WHO (2023). Source: World Health Organization (World Health Organization, 2023).

1.1 Motivation of the Study

In order to better analyze the causes of road traffic accidents and improve the safety of road users, many studies around the world have reported a series of scientific investigation in order to identify primary causal factors. Overall, the factors contributing to road traffic accidents mainly revolve around the interactions of the human drivers with the vehicles, and the roads, and environment. Among these factors, the vast majority of road safety accidents are caused by dangerous driving behaviors of the human. According to reported studies (National Bureau of Statistics of China, 2021), driver behavior accounts for over 90% of traffic accident causes. Over 90% of dangerous driving behaviors typically include violations of traffic rules, lack of safety awareness, and risky driving practices such as fatigued, distracted driving, and sudden acceleration or deceleration. Fatigued driving occurs when the driver experiences a decline in physical and mental functions due to either prolonged concentration or distractions while operating a vehicle, leading to road safety risks (Song, 2023; Li et al., 2010).

At the same time, aside from fatigued driving, reported studies have also highlighted the significance of other dangerous driving behaviors during road traffic participations, which should not be overlooked (Province, 2021). These include the use of a phone, drinking water, smoking, and abrupt acceleration or deceleration. Moreover, the driver's psychological and emotional state while driving is also critical (Petridou and Moustaki, 2000). To reduce road safety accidents caused by human factors and to eliminate such safety hazards as much as possible, a real-time detection and warning system for monitoring driver fatigue, dangerous driving behaviors, emotions, and vehicle handling becomes crucial. The implementation of such a system is vital for protecting human lives, health, and property in commerce drivers, road users, and the public (Kaplan et al., 2015; Wei et al., 2021; McManus et al., 2016).

1.2 Literature Review

Detection of dangerous or distracted driven behaviors encompasses numerous challenges associated with human factors, vehicle dynamics and driven-vehicle-road interactions. Reported studies have employed widely different measures for detecting driver fatigue. These include methods based on physical measures; those based on facial movements and expressions; and those based on vehicle dynamics measures.

1.2.1 Studies Reporting Fatigue Detection via Physiological Measure

International research on fatigued driving mainly focuses on real-time monitoring and evaluation of driver's physiological and psychological fatigue mechanisms during driving (Hayashi et al., 2018). Based on the types of data used in fatigued driving detection, focusing on the studies generally fall into three categories: changes in physiological measures (e.g., EEG, ECG), changes in physical actions (e.g., eye and facial movements), and vehicle operation data (e.g., steering wheel angle, driving posture, and driving trajectory, and acceleration and deceleration). These three types of data are used to assess the driver's fatigue state. The data types for detecting driving behavior are also illustrated in Figure 1.2.

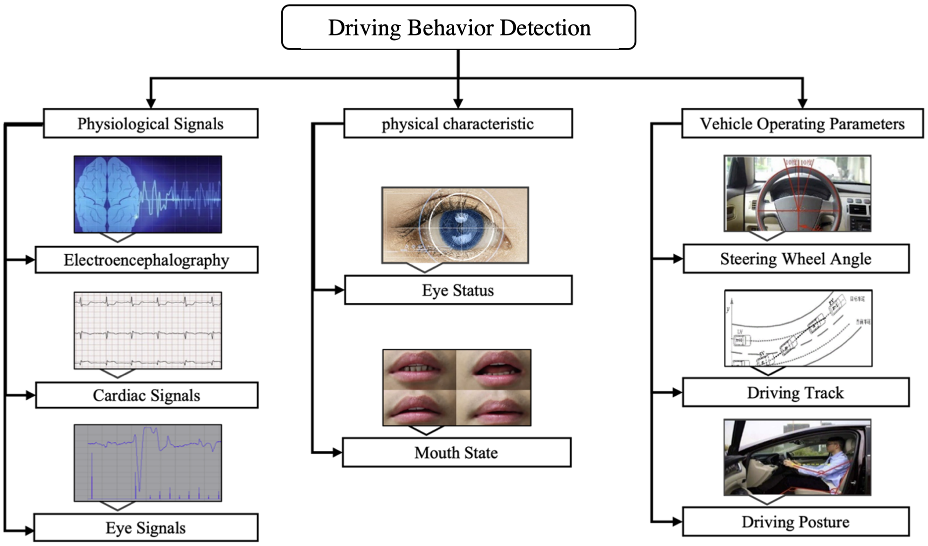


Figure 1.2: Driving Behavior Detection Methods Data Types

Reported studies on these subjects are thus reviewed and summarized in following subsections in order to formulate a synthesis of these studies and to gain knowledge of effective methods and important factors. Studies employing physiological measures monitoring subtle changes in the driver's physiological signals, real-time detection of these indicators allow for assessment of fatigue. A number of studies have employed physiological measures for detecting driver fatigue and risky behaviors. There have used widely different physiological indicators for fatigue detection, such as

Electroencephalogram (EEG) (Sikander and Anwar, 2019; Yin, 2008; Lan, 2021), Electromyography (EMG) (Guo, 2011; Jiao, 2019; Huang, 2020), and Electrocardiogram (ECG) (Cao, 2022; Ye, 2018; Jin, 2017) signals. Research findings have shown that EEG signals exhibit a strong correlation with fatigued driving behavior, making EEG measure the ”Gold Standard” (Shi, 2019) for detecting driving fatigue.

In one study, Saroj and Craig (Lal and Craig, 2001) conducted experiments on 35 non-professional participants and established baseline EEG averages from the participants while awake inferred EEG changes across five different states: awake, mildly fatigued, severely fatigued and startled (Lal and Craig, 2000). Similarly, Zong et al. (Zong et al., 2024) analyzed physiological characteristics using the EEG signals related to driver fatigue in the frequency as well as time domain analyses, in order to other topological features. Authors used these measures to define a fatigue detection model to monitor, generate and issue warnings in a driven-assist system (DAS).

In recent years, the advancements in machine learning, neural network algorithms have been applied for EEG signal processing. For instance, Wang Jie (Wang, 2023) used the Non-Hair-Bearing (NHB) method to acquire EEG signals from three channels on different locations of the driver's forehead. The data were analyzed the data using the Catboost classification algorithm and Tree-of-Parzen-Estimators (TPE) hyperparameter optimization algorithm subsequently applied the GLU-Oneformer quantitative assessment model was to dynamically evaluate driver's fatigue level. The study showed the model could achieve promising results.

In studies focused on detecting and evaluating driver fatigue through vehicle operation parameters, studies have employed parameters closely related to fatigued driving in real driving environments to assess fatigue levels. For example, Jia (Jia, 2019) developed a fatigue detection method based on real-vehicle steering operation characteristics. By collecting experimental data related to steering wheel angle, and vehicle speed, the method simulated various driver fatigue states. Considering both the time-domain and frequency-domain analyses, fatigue features were effectively detected using a support vector machine (SVM) classifier during driver distraction experiments.

In terms of driver posture, studies have mainly focused on head position data. Zhang (Zhang, 2024a), for instance, collected driver head posture data such as pitch, yaw, and roll angles. Using a Long Short-Term Memory (LSTM) neural network model, the study classified the head posture data to detect the drivers' fatigue state.

Furthermore, in studies on driver actions, machine learning and visual processing algorithms have also been applied for fatigue detection. Wang (Wang, 2024) combined the OpenPose posture estimation model with deep learning to analyze six types of driver behaviors or distraction, namely: driving, drinking, smoking, phone usage, lowering the head, and looking around. By tracking key skeletal points during these actions, the study developed a real-time visual detection method to estimate driver posture during the driving process.

In studies aimed at detecting driver body movements, certain facial movements, vehicles are also effectively used as indicators for evaluating driven fatigue, such as the number and duration of eye closures, driving posture, and yawn frequency (Zhang and Ye, 2022). The reliability of the method, however, has been widely questioned considering broad variation in the data. These are key signals that reflect a driver's level of fatigue. Generally, these methods focus on evaluating the driver's eyes and facial movements using machine vision and object detection algorithms to extract critical feature values. For example, Feng et al. (Feng et al., 2024) developed a method based on computer image processing that tracks the facial expression and detects eye closure when driving speeds exceed 70 km/h.

In the context of deep learning applications, researchers have applied fatigue detection in more complex driving environments. Due to variations in driving conditions and environment, such as cabin lighting, head movement, and facial expressions, the accuracy of algorithms becomes even more critical. Zhao (Zhao, 2021) designed a driver fatigue detection system based on facial features, utilizing a Multi-Task Cascaded Convolutional Network (MTCNN) for facial detection and a CNN+LSTM dynamic fatigue recognition model. The system reported algorithm for detecting and evaluating driver fatigue under various driving conditions by analyzing 68 different facial feature points and classifying six key indicators, namely: Focus, Feedback, Engagement (FFE), Maximum Duration Fatigue Expression (MDFE), Percentage of Eye Closure (PERCLOS), Blinking Frequency (BF), Yawning Frequency (YF), and Nodding Frequency (NF).

In summary, the methods for detecting driver fatigue mentioned in this chapter mainly focus on monitoring changes in physiological signals, body movements, and driving operation parameters. These methods provide a comprehensive approach to evaluating fatigue. However, despite their theoretical promise, practical applications may face issues such as false positives and false negatives. Additionally, there exists a lack of standardized criteria for assessing driver fatigue, which does not permit comparisons, relative analyses and validations of results across different studies.

1.2.2 Vision-Based Methods for Detecting Distracted Driving Behavior

The technological advancement of technology and the proliferation of in on-board sensors together with entertainment systems in modern smart vehicle cabins and the widespread use of smartphones have led to an increasing number of factors contributing to distracted driving. In real-world driving, seemingly minor “small actions” or “habits” can cause devastating accidents (Yang, 2023). As a result, a series of scientific studies have focused on detecting distracted driving behaviors and emotions during the driving process.

In recent years, Advanced Driver Assistance Systems (ADAS) have emerged as sophisticated platforms for driver assistance, including monitoring of abnormal driving behaviors and providing various forms of warnings (Martinez et al., 2018). Implementing these systems, however, is highly challenging due to the need for accurate and real-time identifications. Studies have attempted to use Vehicle Ad Hoc Networks (VANETs) to detect one or more driver behaviors (Hasrouny et al., 2017). Al-Sultan et al. (Al-Sultan, 2013) extended VANET by integrating a real-time probabilistic model based on a Dynamic Bayesian Network (DBN). This approach combined contextual information related to the driver, the vehicle, and the environment to infer dynamic driver behaviors and detect four different indicators, namely, intoxication, fatigue, aggression, and normal behavior. Given the large number of parameters and considering the VANET operation as a Mobile Ad-hoc Network (MANET), the transmission of contextual information to the real-time probabilistic model via Dedicated Short-Range Communications (DSRC) imposed significant computational demands. The reduction in the number of parameters was thus considered crucial for maintaining reasonable computational efficiency of the network.

Advances in machine vision and image processing technologies have significantly enhanced the recognition of negative emotions and abnormal driving behaviors. Seyhan and Oguchi (Uçar and Oguchi, 2021) used distance and path deviation metrics to detect distracted driving behaviors.

Zheng et al. (Zheng et al., 2020) proposed an improved CornerNet Sade method to identify distractions caused by smoking or eating while driving. The study (Wang et al., 2021) developed a machine vision-based distracted driving detection method using a Fast Region-based Convolutional Neural Network (R-CNN) model. The method employed class activation mapping to analyze key driving behavior features.

The real-time object detection network, You Only Look Once (YOLO) (Jiang et al., 2022), was considered a breakthrough in object detection, as it treated the problem as a straightforward regression task. YOLO operated significantly faster than the popular two-stage detectors like Faster R-CNN, although it may compromise some accuracy. Various alternate YOLO networks with different architectures have also been developed to enhance detection precision. For instance, Qin et al. (Qin et al., 2022) developed a method called ID-YOLO that detects distracted driving behaviors by identifying key objects observed by the driver. Hnewa et al. (Hnewa and Radha, 2023) introduced an integrated Multi-Scale Domain Adaptive YOLO (MS-DAYOLO) framework for efficient real-time object detection. The proposed network addresses domain shift issues encountered in many deep learning applications with improved speed.

In summary, this chapter discussed research methods for recognizing distracted driving behaviors and emotions, primarily employing technologies such as Advanced Driver Assistance Systems (ADAS), Vehicle Ad Hoc Networks (VANETs), and machine vision. These methods span various technical fields, including communication technology, machine vision, and deep learning to address the problem of distracted driving from multiple perspectives. Advanced technologies and algorithms, such as Dynamic Bayesian Networks and improved object detection models, have been used to enhance the accuracy and reliability of detecting distracted driving behaviors. These methods, however, require substantial computing resources and complex algorithms for real-time recognitions, which may lead to issues with computational efficiency or delays. Additionally, technologies like ADAS and machine vision necessitate advanced hardware and software support, potentially increasing the cost burden for vehicle manufacturers and drivers.

1.2.3 Methods Based on Vehicle Dynamic Responses

Studies on dangerous driving behaviors typically categorize these risky behaviors as ’three sudden actions and speeding (sudden braking, sudden turning, sudden acceleration, and speeding) (Sathyanarayana et al., 2008). Such driving data can directly reflect the driver’s behaviors. For instance, Zhang (Zhang et al., 2019) defined an absolute acceleration threshold of 3 m/s^2 and suggesting that acceleration greater than 3 m/s^2 indicates a dangerous driving scenario, while deceleration less than -3 m/s^2 indicates dangerous sudden brakeing. Vehicle data, including instantaneous speed and acceleration over time, could be analyzed and classified using algorithms such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), decision trees, and random forests, combined with the k-means++ clustering algorithm. Such data are then input to a Backpropagation (BP) neural network model for detecting and evaluating dangerous driving behaviors (Zhang, 2021).

Zhang (Zhang, 2024b) utilized vehicle-to-everything (V2X) technology and OBD (On-Board Diagnostics) smart vehicle terminals to collect driving data. Algorithms were employed to mine and analyze this data to identify instances of sudden acceleration and deceleration. In a vehicle-based analysis platform study, Lu (Lu, 2016) used a strong classifier (AdaBoost) to compare features such as vehicle speed, RPM, throttle position, and engine load (OBD-II data). These features were used in an experimental study, and a BP neural network algorithm was employed to evaluate a risky driving behavior.

In summary, these studies discussed methods for detecting dangerous driving behaviors, related to sudden acceleration and deceleration. These methods primarily involve analyzing driving data to detect and evaluate risky driver behaviors. Using the actual vehicle data, which provides objectivity and clarity in defining driver actions. However, some methods required vehicles to be equipped with specialized devices or utilize vehicle-to-everything (V2X) technology to gather driving data, potentially increasing the cost and complexity of data collection. Additionally, algorithms like BP neural networks impose high computational complexity, requiring significant resources and time for model training, which may affect the real-time performance and efficiency of these methods.

1.3 Objectives of the thesis

The primary objective of this thesis is to develop an efficient and robust multi-scale identification method for detecting dangerous driving behaviors by leveraging multivariate data and providing both the theoretical and technical support for the formulation of a comprehensive driver warning platform. Given the critical role that dangerous driving behaviors play in traffic accidents and their severe impact on road safety, developments in accurate detection and early warning mechanisms are vital. To achieve this overarching goal, this thesis focuses on the following specific objectives:

1) Develop a dangerous Driving Behavior Recognition Architecture: The reported architectures for dangerous driving behavior recognition are reviewed and an enhanced recognition method based on the YOLOv8 (You Only Look Once version 8, a modified YOLO) neural network model is proposed. By integrating the Multi-Head Self-Attention (MHSA) module, to facilitate global target modeling within a larger receptive, thereby, improving target recognition accuracy. This enhancement aimed to reduce false and missed detections and increase the overall performance and robustness of the detection system. Furthermore, a driving emotion detection module was incorporated into the network to share the Region of Interest (ROI) feature, thereby adding emotion recognition functionality without significantly increasing computational complexity.

2) Development of Driver Pedal Operation Detection: Another key objective is to detect driver brake pedal and accelerator pedal operations using video frame data captured by a leg camera. This is accomplished using the Mask-RCNN instance segmentation deep learning network. By employing ROI Align during down-sampling, the thesis focused on enhancement of the network's feature extraction capability, thereby improving the accuracy of pedal operation recognition while maintaining the model's computational complexity.

3) Detection and Evaluation of Driving Data Based on CAN Signals: The thesis also focuses on detecting dangerous driving behaviors through analyses of brake and accelerator signals obtained from vehicle CAN (Controller Area Network) data. By applying low-pass and median filtering of the signal data and using a sliding window method to compare and evaluate extracted signals against with known thresholds, the method aimed to detect rapid acceleration and decelerations, enabling a comprehensive evaluation of dangerous driving behaviors.

4) Development and Integration of a Real-Time Detection System: The final objective is to integrate the neural network training weights and detection algorithms using Python and the Qt Designer software system. This will involve building a comprehensive detection system for dangerous driving behaviors, complete with a visually accessible real-time front-end UI. Additionally, a real vehicle experimental verification scheme will be designed to validate the effectiveness and superiority of the proposed detection methods in actual driving scenarios, demonstrating their applicability in real-world environments.

1.4 Thesis Outline

The rest of this thesis is organized as follows:

In Chapter 2, the developments in history of the YOLO series of target detection algorithms are presented in Chapter 2. The PERCLOS algorithm for the detection of fatigue driving behaviors is also introduced together; secondly, the example segmentation algorithms are described in detail, and the relative advantages and disadvantages of the various example segmentation algorithms are discussed.

In Chapter 3, the neural network modeling algorithm and associated data processing techniques utilized in this study are presented as the core components of the proposed methodology. Firstly, the YOLOv8 target detection algorithm based on the MHSA Attention Mechanism Module and the Driver Emotion Detection Module is introduced so that its Region of Interest (ROI) data can be shared to achieve the detection of driver fatigue, distracted driving behavior, and driving emotion (the above was submitted as a regular paper to the IEEE Access and accepted); secondly , as for the construction of instance segmentation algorithm Mask-RCNN is used to detect the driver's operation of vehicle pedals; finally, the detection and evaluation of the brake signal and gas pedal signal in the vehicle driving data are used in a window sliding processing method, to achieve the detection of rapid acceleration and deceleration of hazardous driving behaviors based on signal data processing.

Chapter 4 focuses on the training and evaluation process of the models and algorithms. Firstly, the preparation of the dataset is introduced: the self-constructed dataset of distracted driving behavior, the FER2013 emotion public dataset, and the real vehicle dataset of Yutong Bus Co. Ltd.

(video frame signal data and driving data); secondly, the experimental environment required for the training and evaluation of this model and algorithm and the evaluation indexes used are described in detail; the evaluation of the training results of model of neural network algorithm and the sliding window algorithm validation of the sliding window algorithm. Finally, the formulation of a multivariate data-driven platform for risky driving behavior detection is proposed in this study, aiming to integrate multiple data sources for enhanced detection accuracy and system robustness, as well as the effect of the real-vehicle test. Firstly, it is introduced for the construction process of the real-time visualization front-end UI followed by platform and experiment design; secondly, it gives a detailed introduction to the experimental platform vehicle used in this real-vehicle experiment; finally, it introduces the process of this real-vehicle experiment in detail and shows examples of the effect in the process of real-vehicle experiment.

In Chapter 5, summarizes the highlights of this study, together with the shortcomings, major conclusions and prospects of probable future work.

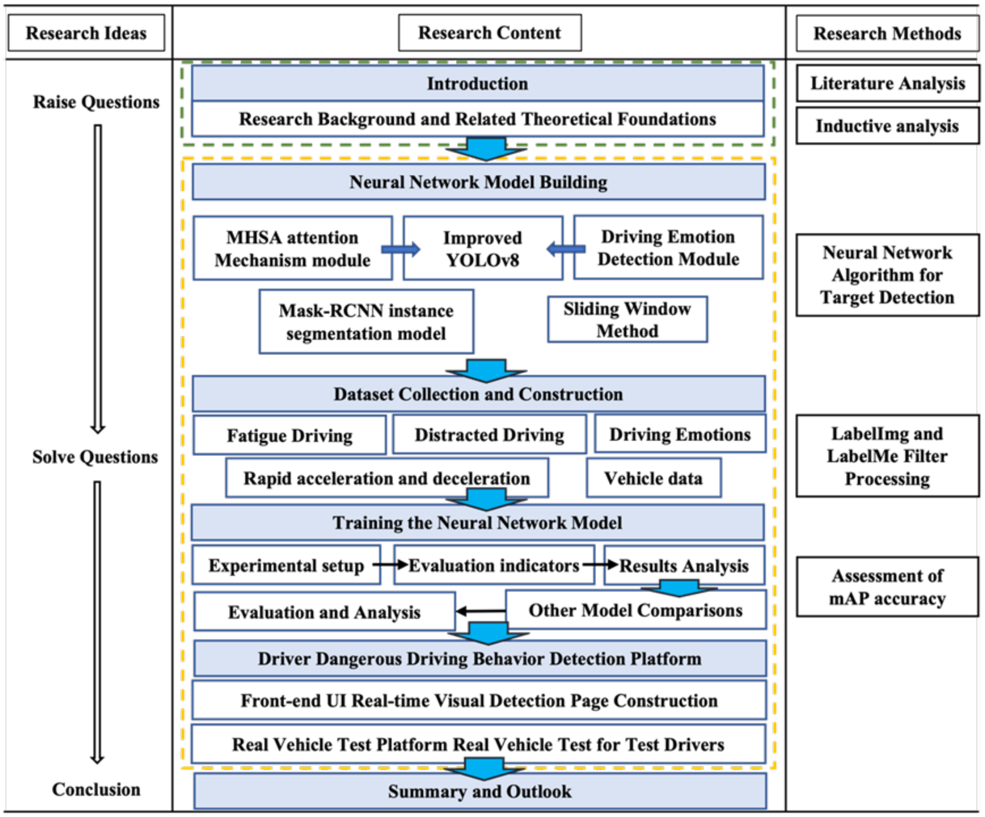

Research Ideas
Research Content
Research Methods
Raise Questions
Solve Questions
Conclusion
Introduction
Research Background and Related Theoretical Foundations
Literature Analysis
Inductive analysis
Neural Network Model Building
MHSA attention Mechanism module
Improved YOLOv8
Driving Emotion Detection Module
Mask-RCNN instance segmentation model
Sliding Window Method
Neural Network Algorithm for Target Detection
Dataset Collection and Construction
Fatigue Driving
Distracted Driving
Driving Emotions
Rapid acceleration and deceleration
Vehicle data
LabelImg and LabelMe Filter Processing
Training the Neural Network Model
Experimental setup
Evaluation indicators
Results Analysis
Evaluation and Analysis
Other Model Comparisons
Assessment of mAP accuracy
Driver Dangerous Driving Behavior Detection Platform
Front-end UI Real-time Visual Detection Page Construction
Real Vehicle Test Platform Real Vehicle Test for Test Drivers
Summary and Outlook

Figure 1.3: Structure of the Technical Route

Chapter 2

Detection of a Risky Driving Behavior Using Deep-Learning Method

This chapter mainly outlines the most relevant concepts and methods of this thesis. In the research of traditional studies focusing on detection methods, the required standards cannot be achieved due to several objective factors and confounders such as high hardware cost, complex network structure, large amount of calculations, and poor real-time performance. With the availability of more and more public datasets on various dangerous driving behaviors, deep learning and machine vision techniques could facilitate. The reposted existing focused on detecting one or several specific types of dangerous driving behaviors rather than providing a comprehensive detection platform that could handle all relevant scenarios dangerous driving behaviors. This chapter will briefly describe the theoretical concepts of neural networks, deep learning, and data processing, as well as the development of an improved YOLO algorithm, while providing a detailed overview of the neural network model used in the study.

2.1 YOLO Target Detection Algorithms

As a single-stage lightweight target detection algorithm, the use of YOLO has grown very rapidly in recent years, with a simpler network architecture and faster detection algorithm. YOLO is mainly used to visually detect specific individuals, animals, or objects and classify and locate these objects in digital images (Zou et al., 2023). More strikingly, YOLO can achieve one-time recognition of multiple objects at the same time and offers a significant advantage, as its acronym suggests. The YOLO series of algorithms have been developed at an astonishing speed in recent years; with meritorious features many versions have appeared. Figure 2.1 depicts the development process of YOLO could thus be found in the literature.

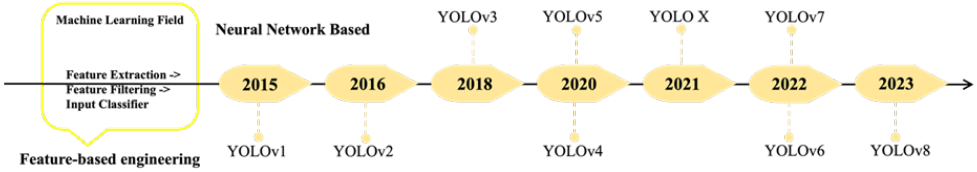


Figure 2.1: YOLO Series Mainstream Algorithm Development Process

YOLOv1 (Redmon et al., 2016) was first introduced in 2015. As the first single-stage target detection algorithm, it exhibits a simpler network architecture and a small size. In addition, it brings a new way of task detection, constructing it as an end-to-end regression problem, which has faster detection speed and higher accuracy than the popular R-CNN network model at the time. With the rapid developments in the YOLO series, from YOLOv2 to YOLOv8, each version has been continuously optimized and improved in terms of running performance, model architecture, and detection accuracy. This iterative process makes these more applicable in the visual detection direction of autonomous driving (Yang and Li, 2023). Table 2.1 summarizes the key improvements in the iterative process of the YOLO series. This study mainly introduces the detection of driver fatigue, distracted driving behavior, and driving emotions based on the improved YOLOv8 target detection network model and compares and analyzes the accuracy of YOLOv8 with other versions of the series (YOLOv4, YOLOv5, YOLOv7, and native YOLOv8) and draws conclusions.

In the study, a more advanced iteration of the YOLO series has been used. In fact, YOLOv8 is based on the network architecture of YOLOv4, further optimizing the network structure and adopting a more effective training scheme to achieve faster detection speeds and higher detection accuracy. In the network architecture, YOLOv8 employs a more efficient, lighter, and more powerful backbone network (such as CSPDarknet53 or CSPDarknet19), thereby improving the efficiency and accuracy of feature extraction. It also utilizes multi-scale training and inference methods to further improve the detection accuracy of targets of different scales and aspect ratios, thereby enhancing

Table 2.1: Summary of YOLO Series of Detection Algorithm and Their Key Features

Version	Main Improvements	Advantages and Limitations
YOLOv1 (Ahmad et al., 2020)	Anchor, frame structure with efficient feature learning	Simple network but poor positioning
YOLOv2 (Huang et al., 2021)	BN layer, DarkNet-19, transfer module	Precise classification with relatively lower accuracy
YOLOv3 (Redmon and Farhadi, 2018)	DarkNet-35, multi-scale feature fusion	Faster speed, lack of strict precision on edges
YOLOv4 (Bochkovskiy et al., 2020)	FPN+PA Net, Mish activation function	Accurate small target detection but relatively greater complexity
YOLOv5 (Jocher et al., 2022)	Performance improvement, SPPF, multiple models	Small model size and fast inspection
YOLOX (Ge et al., 2021)	Decoupled header, anchor-free design	Many framework choices
YOLOv6 (Li et al., 2022)	Efficient Rep, SimSPP	Fast detection and industrial bias
YOLOv7 (Wang et al., 2023a)	Reparameterized modules, dynamic label assignment	High precision, high speed but higher complexity
YOLOv8 (Wang et al., 2023b)	Decoupled header, anchor-free, TAL	Highly extensible and compatible with all frameworks

the robustness and accuracy of detection. In addition, in terms of data processing, the model utilizes a variety of data enhancement techniques, such as random scaling, cropping, and color jittering, and expands the diversity of training data. The iteratively developed YOLOv8 combines the classification error and the bounding box positioning error into a new loss function, which gives it stronger data analysis capabilities during the training process. In the lightweight improvement, YOLOv8 also employs methods such as pruning, quantization, and distillation to achieve model compression and acceleration with acceptable performance loss, making it more suitable for deployment in resource-limited systems (such as vehicle-mounted terminals, etc.).

The network structure diagram of the native YOLOv8 is shown in Figure 2.2. As can be seen from the figure, the network structure of YOLOv8 is mainly divided into two parts: Backbone and Detection Head. Among these, the Backbone network uses CSPDarknet53 or CSPDarknet19. CSPDarknet53 is a 53-layer deep Backbone network consisting of convolutional layers, residual blocks, and Cross Stage Partial (CSP) modules. This network effectively avoids the gradient vanishing problem and enhances information flow through skip connections and residual connections to residual blocks. In the CSPDarknet53 network, the CSP module splits the feature map into two parts. One part inputs the feature data into a series of convolution functions and activation functions for processing and analysis; the other part remains unchanged. The two parts subsequently integrated are connected to accelerate information propagation. In addition, the Detection Head network, another important part of the network consists of a series of convolutional layers, downsampling layers, and convolution kernels. The network consists of the above three network layers, where the convolution layer further processes the feature parameters extracted and input by the backbone network, and the downsampling layer network reduces the spatial dimension of the feature map through convolution or pooling operations with a step size greater than 1. The convolution kernel is responsible for predicting key information, analyzing and evaluating the category probability, bounding box position, and confidence score of the detected target. The entire YOLOv8 network structure significantly improves the speed and accuracy of target detection through end-to-end training and reasoning.

The network architecture of YOLOv8 is illustrated in Figure 2.2. The functional descriptions of each module in the network architecture are presented in Table 2.2.

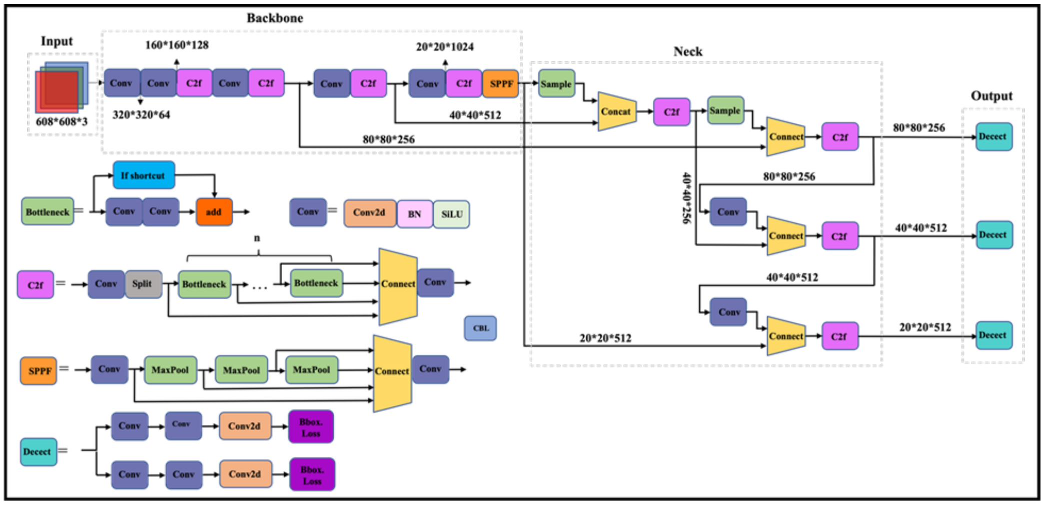


Figure 2.2: Structure of the YOLOv8 Target Detection Algorithm Network

2.2 PERCLOS Fatigue Driving Detection Algorithm

In the detection of driver fatigue, the measurement method based on the percentage of eyelid closure (PERCLOS) is recognized as one of the most effective and validated detection methods.

Table 2.2: Summary of YOLOv8 Architecture Components

Component	**Full Name/Explanation**	**Function Description**
Conv	Convolutional Layer	Standard convolution layer with BN + SiLU activation
C2f	Cross Stage Partial Network with 2 fusions	Improved C3 module with better gradient flow
SPPF	Spatial Pyramid Pooling Fast	Fast multi-scale feature pooling
Upsample	Upsampling Layer	Increases feature map resolution
Concat	Concatenation	Merges feature maps along channel dimension
Detect	Detection Head	Decoupled classification and regression outputs
BN	Batch Normalization	Normalizes layer outputs for stable training
SiLU	Sigmoid Linear Unit	$x \cdot \text{sigmoid}(x)$ activation function
DWConv	Depthwise Convolution	Lightweight convolution operation
Bottleneck	Bottleneck Block	Residual block with channel reduction

The PERCLOS measurement method was first proposed and validated by Wierville et al. to detect and evaluate degree of driver fatigue (Wierwille et al., 1994; Wierwille and Ellsworth, 1994). Subsequently, Dinges et al. conducted a validation study based on this in 1998 to evaluate the effectiveness of PERCLOS in detecting errors in the psychomotor vigilance test (PVT) during driver fatigue (Dinges et al., 1998). The results showed that among all detection methods, including EEG and facial movements (such as blinking, etc.), PERCLOS was the most accurate measurement method for detecting PVT errors. Since then, PERCLOS has been widely validated as the optimal measurement method for detecting driver fatigue in various situations, including simulated driving in laboratory environments and actual road driving (Sparrow et al., 2019; Cori et al., 2019; Cai et al., 2021; Abe et al., 2014).

Research on driver fatigue detection based on other data types has emerged in an endless stream. The most typical ones are direct detection methods and indirect detection methods. The direct detection methods judge and evaluate the driver's eyelid opening and closing movements, yawning, and head movements (such as nodding), while the indirect detection methods focus more on the analysis and evaluation of the driver's driving trajectory deviations. It was later confirmed that the above detection indicators are derivative indicators related to reliability and strong situational dependence (Zhihu, 2023).

The PERCLOS measurement assesses fatigue by the percentage of time the eyes are closed during a specific period (usually a defined period). PERCLOS has three measurements: P70: the percentage of time the eyelids cover more than 70

The formula for calculating PERCLOS is as follows:

$$\text{PERCLOS} = \frac{\text{Time eyes are at least 80\% closed}}{\text{Total observation time}} \times 100\% \tag{2.1}$$

The PERCLOS detection method uses video frame data extracted from the vehicle's driver monitoring system (DMS) camera for evaluations. That is, the video frame data are extracted by the camera and input into the controller of the facial recognition algorithm for analyses and evaluations. Following the analyzing frame by frame analyses in which the eyelid covers more than a preset proportion of the eyeball surface, the PERCLOS detection method is used to divide the number of frames identified as fatigue by the total number of frames in each period, and finally the PERCLOS value is calculated and output.

The Federal Highway Administration (FHWA) and the National Highway Traffic Safety Administration (NHTSA) in the United States have conducted experiments and tests on fatigue driving in a laboratory environment, mainly focusing on the evaluation and comparison of nine different fatigue detection methods in simulated driving. The experimental results showed that, although the other eight fatigue driving detection methods were able to detect the driver's fatigue state to varying degrees, the PERCLOS detection method provided best correlation with fatigue driving detection (CSDN, 2016).

2.3 Instance Segmentation Algorithm

The instance segmentation visual detection algorithm is also one of the target detection algorithms of machine vision. It mainly focuses on segmenting each detected target instance in the detection image. Unlike the target detection algorithms mentioned above, the instance segmentation visual detection algorithm can depict the precise boundary of the target and perform better in the detection task of target details. The development history of the instance segmentation algorithm is shown in Figure 2.3.

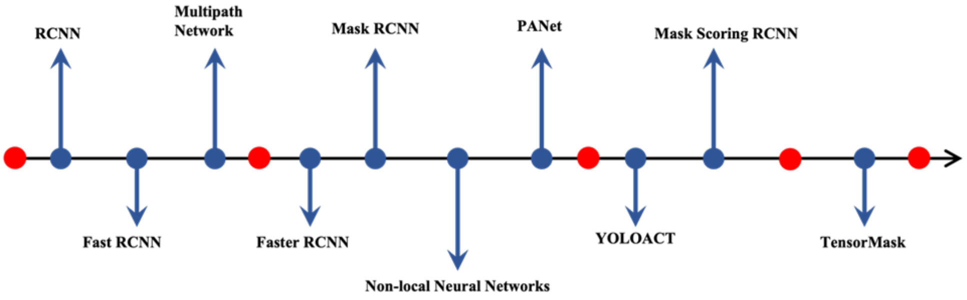


Figure 2.3: Segmentation Algorithm Development History Diagram (Zhang, 2009)

The instance segmentation algorithm is mainly divided into four main network architecture methods: Mask Proposal Classification, Detection-Then-Segmentation, Pixel Labeling and Clustering, and Dense Sliding Window. The main technical methods mentioned above are shown in Table 2.3. Among these, the second classification method (detection-segmentation method) has been most widely used in the field of autonomous driving and advance manufacturing because its data set format is more standardized, and the training conditions and requirements are relatively simple. The specific algorithm flow of the detection-segmentation method is shown in Figure 2.4.

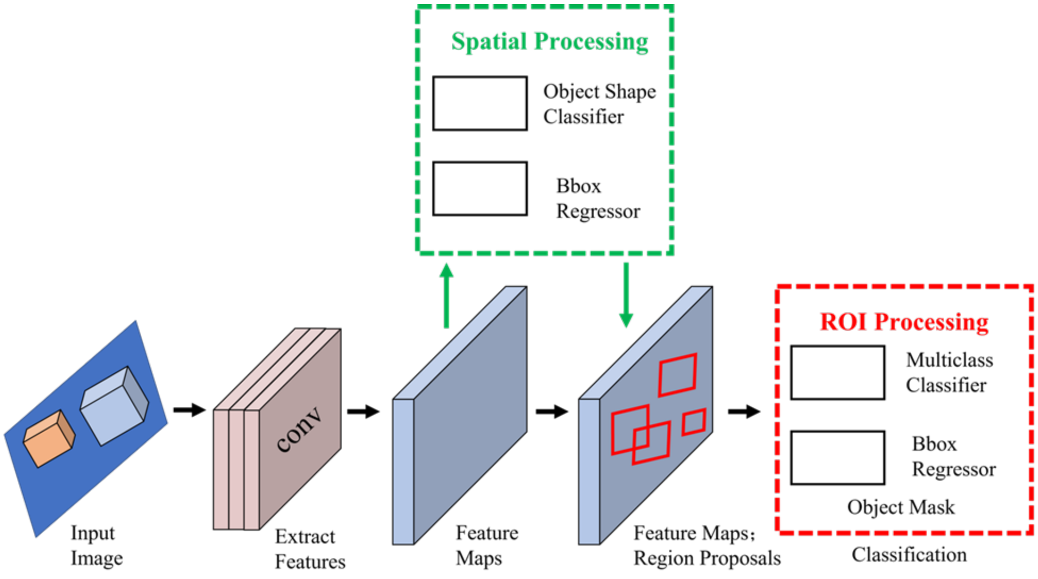


Figure 2.4: Network Flowchart of the Detection-Then-Segmentation Method

Table 2.3: Instance Segmentation Primary Network Architecture Approaches

Category	Representative Algorithms
Mask Proposal Classification	R-CNN, Fast R-CNN, Faster R-CNN
Detection Followed by Segmentation	HTC, PANet, Mask R-CNN, Mask Scoring R-CNN, MPN, YOLACT
Pixel Labeling Followed by Clustering	Deep Watershed Transform, Instance Cut
Dense Sliding Window	Deep Mask, Instance FCN, TensorMask

The following is a detailed description of the image processing steps of the real-time segmentation algorithm: Image input: First, the video frame data is output from the real-time camera, and each frame of image data is passed as input to the real-time segmentation algorithm model. Since the algorithm is not very sensitive to resolution and quality, these images can be in JPEG, PNG, or other formats. Object detection: The model performs target detection on the input image, searches for the target object of interest, and outputs a bounding box that tightly surrounds these objects after searching for the target. Commonly used object detection algorithms include Faster R-CNN, YOLO, or SSD, which accurately detect objects of interest after multiple trainings on many data sets. Instance segmentation: After the object is detected in the previous step, the algorithm model continues to segment each detected object instance separately and accurately depicts the boundaries within the bounds of each object instance. Mask generation: The main purpose of instance segmentation algorithms is to generate a corresponding mask for each detected object instance. These masks are binary arrays of the same size as the input image, and each of their pixel parameters is marked as to whether it belongs to the pixel within the bounding box of the object instance. Refinement: After the previous step is completed, the instance segmentation algorithm will use refinement techniques to depict the segmentation boundaries in more detail to improve segmentation accuracy, such as refining mask boundaries, reducing false positives, or handling overlapping instances. Output: The final output of the instance segmentation algorithm is subsequently a set of segmented object instances and a set of masks corresponding to them. When this set of segmented object instances and masks is superimposed on the original image, the system can detect the segmentation results very intuitively. Owing to its special detection method, instance segmentation algorithms are widely used in many fields such as autonomous driving, robotics, and medical image analyses. These fields exhibit a common feature. They all require very accurate and fitting target object detection.

Chapter 3

Modification

This chapter introduces in detail three dangerous driving behavior detection methods based on different data types: driving behavior and driver emotion detection based on the improved YOLOv8 model combined with the attention mechanism; driver pedal operation detection based on video processing; dangerous driving behavior detection based on vehicle data. These methods will provide important technical support for the research and development of real-time driving monitoring systems.

3.1 Detection of Driver Distracted Driving Behavior Based on Target Detection

3.1.1 Multi-head Self-attention

As a newer version of the target detection algorithm series, YOLOv8's detection speed and accuracy have always been the focus of research in various fields (Zhou et al., 2023). This paper innovatively integrates the multi-head self-attention (MHSA) module into the backbone network of YOLOv8 to improve detection speed and accuracy. MHSA is one of many attention mechanisms. Due to its special information exchange mechanism, it can efficiently simulate the dependencies between each different position in the input sequence. It splits the input sequence into different head information, calculates the attention weight for each group of head information, and outputs it weightily. It is worth noting that each group of head information has its own set of queries, keys, and values, and these queries, keys, and values are derived from the input sequence through linear transformation (usually a fully connected layer).

This unique design structure can more effectively capture the relationship between the head information of each group of different positions in the input sequence and can generate more information exchanges within a certain period, thereby improving the detection speed of the model. In this experiment, by introducing the MHSA module, the model can focus on the target globally in the receptive field to a greater extent by using the MHSA module. This strategy effectively reduces false positives and missed detections in target detection, and thereby, greatly improves the accuracy of model detection leading to improved performance and robustness of the model. The working principle of MHSA is shown in Figure 3.1, and its main workflow is shown in Figure 3.2.

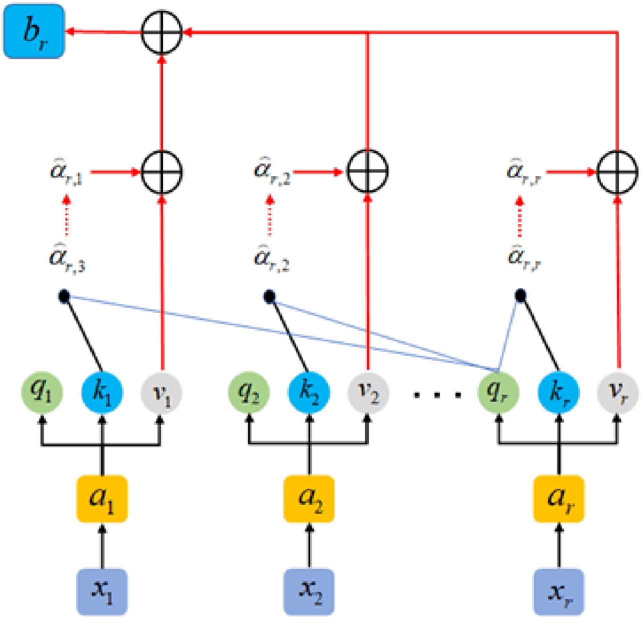


Figure 3.1: MHSA Working Principle Diagram of the MHSA Module

The outcome of the attention mechanism b_r is evaluated from weighted sum of individual attention value score v_i of the i^{th} head; T is the total number of samples, as:

$$b_r = \sum_{i=1}^{T} \hat{a}_{T,i} \cdot v_i \tag{3.1}$$

where, b_r represents the final output of the attention mechanism, computed as the weighted sum of

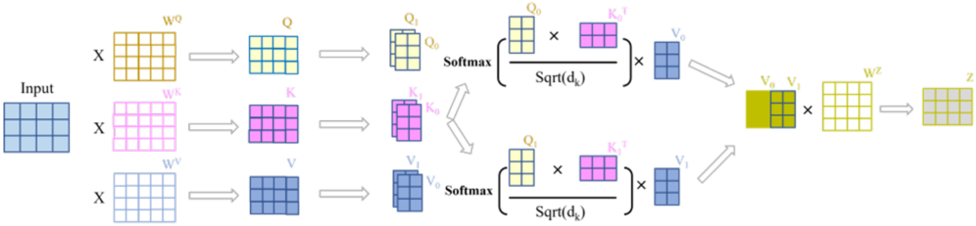


Figure 3.2: MHSA Main Workflow Diagram of the MHSA Module

the value vectors v_i with weights $\hat{a}_{T,i}$. The normalized attention scores are obtained from:

$$\hat{\alpha}_{T,i} = \text{softmax}(\alpha_{T,i}) \tag{3.2}$$

where, $\hat{a}_{T,i}$ denotes these normalized attention scores obtained by applying the softmax function to the unnormalized attention scores $\alpha_{T,i}$.

$$\alpha_{T,j} = \frac{q_T^\top \cdot k_j}{\sqrt{d_{q,k}}}, \quad j = 1, 2, 3, \dots \tag{3.3}$$

The attention score $\alpha_{T,j}$ is calculated as the scaled dot product between the query vector q_T and the key vector k_j, where $d_{q,k}$ is the dimensionality of the query and key vectors used to scale the dot product, such that:

$$q_i = W^Q a_i \tag{3.4}$$

$$k_i = W^K a_i \tag{3.5}$$

where, the query, key, and value vectors q_i, k_i, and v_i are derived from the input sequence a_i using the linear transformation matrices W^Q, W^K, and W^V, respectively.

$$v_i = W^V a_i \tag{3.6}$$

$$a_i = W x_i \tag{3.7}$$

where, a_i is the result of applying the linear transformation matrix W to the input element vector x_i. The input to MHA consists of three vectors: the query vector (q_i), key vector (k_i), and value vector (v_i).

MHA performs a weighted summation of the key vectors, compares the similarity between the key vectors using a given query vector. It subsequently computes, calculates and applies weights to generate the output, as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{3.8}$$

where, Q, K, and V represent the query, key, and value vectors, respectively. The transformation matrix W is used to transform the output of each head ($\text{head}_i; i = 1, \ldots, h$).

The self-attention mechanism computes head_i through the attention function, such that:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3.9}$$

where, QW_i^Q, KW_i^K, and VW_i^V are the query, key, and value transformation matrices for the i^{th} head, respectively. The attention in MHA (Multi-Head Attention module) is computed based on the self-attention mechanism, such that:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3.10}$$

where, d_k is the dimensionality of the key vectors, and the superscript “$\top$” denotes the matrix transpose.

3.1.2 Emotion Detection

As one of the most classic neural network models, the convolutional neural network (CNN) is at a mature stage in terms of global and local feature capture and extraction, data processing, and training schemes. In the facial emotion recognition introduced in this section, the required detection method focuses more on identifying and classifying emotions conveyed by facial expressions. In this case, the focus of YOLO detection is more on dividing the image into equal size grids and classifying and locating the objects in each grid. Since the CNN network itself is particularly well-suited for tasks that require higher sensitivity and fine-grained feature extraction. The CNN network architecture for expression detection could help optimize facial feature extraction and large-area expression classification, aiming to carefully classify facial expressions with less obvious features.

Obviously, in driving emotion detection, the CNN network is more suitable for this task than YOLO after specific detection optimization. The integration of the CNN module for driving emotion detection in the improved YOLOv8, however, ruminates challenge for application towards emotion detection. It is worth noting that the YOLOv8 network contains ROI, which includes information in the region of interest. As mentioned above, the improved YOLOv8 comprises the function of extracting features of 68 key points of the face, so we can promote sharing information contained in its ROI with the CNN module, thereby realizing the simultaneous detection of driving emotions.

The integration of CNN module for driving emotion detection, involves two important steps into the improved YOLOv8 network: Firstly, it utilizes the publicly available use the public dataset, FER2013, to train the model driving emotion detection module and saves the weight file; The driving emotion detection module is subsequently applied in the target area of the improved YOLOv8 face detection. The structures of the two models are finally integrated, shared and combined in terms of the ROI to realize simultaneous detections of fatigue, distracted driving behavior, and driving emotions.

Figure 3.3 shows the network structure designed for driving emotion detection CNN module. Firstly, the improved YOLOv8 detection network inputs the ROI information of facial detection into the input layer of the module and generates 48×48 grayscale image data that are input to the detection layer. The detection layer encompasses a series of convolutional layers, ReLU activation functions and pooling layers that include 3x3 convolutional filters with 32 channels and 64 channels of 3x3 convolutional layers with 128 channels. After each convolutional layer and ReLU activation function, the size of the feature value is further reduced by the pooling layer. Each set of feature values is then flattened into a one-dimensional vector and input to a fully connected layer consisting of 64 units, the number of units in this layer is equal to the number of emotion categories (e.g., anger, disgust, fear, happiness, sadness, surprise and neutrality). Finally, the softmax activation function is applied to the connected layer to generate the driving emotion outcome. The model uses the cross-entropy loss function for multi-label classification and is optimized using the Adam optimizer (Zhang, 2018).

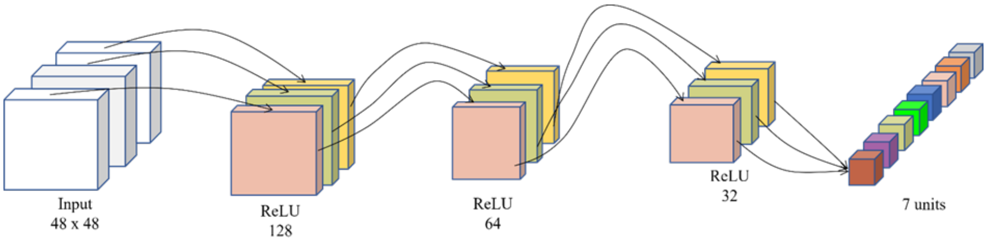


Figure 3.3: Driver Emotion Detection Module Detection Flowchart

3.1.3 Improved YOLOv8

This study proposed a detection model for fatigue, distracted driving behavior and driving emotions based on an improved YOLOv8 network. The model integrates a multi-head self-attention (MHSA) mechanism module and a driving emotion detection module. The MHSA module optimizes the head information of spatial features by cross-updating and sharing ROI information to achieve multi-task processing of driving emotion detection at the same time. These improvements make the improved YOLOv8 network model simultaneously detect various indicators of drivers faster and more accurately.

The network architecture and detection operation principles of the MHSA module have been introduced in the previous chapter. The chapter also described the drivers' emotion detection module. This section will introduce the integration of these two modules into the YOLOv8 network architecture in detail to facilitate simultaneous detections of driving behavior and driver emotion and explain their deployment principles and detection process.

Figure 3.4 shows the architecture of the improved YOLOv8 network model integrating the MHSA module and the driving emotion module. The multiple convolutional layers in the backbone network structure are initially analyzed for extracting basic features from the input image. This part of the network is mainly used for information and feature extractions of the input image. In order to effectively apply the MHSA attention mechanism, the MHSA layer is introduced after the last convolutional layer of the backbone. This ensures that the feature input of the MHSA layer is thoroughly processed and integrated.

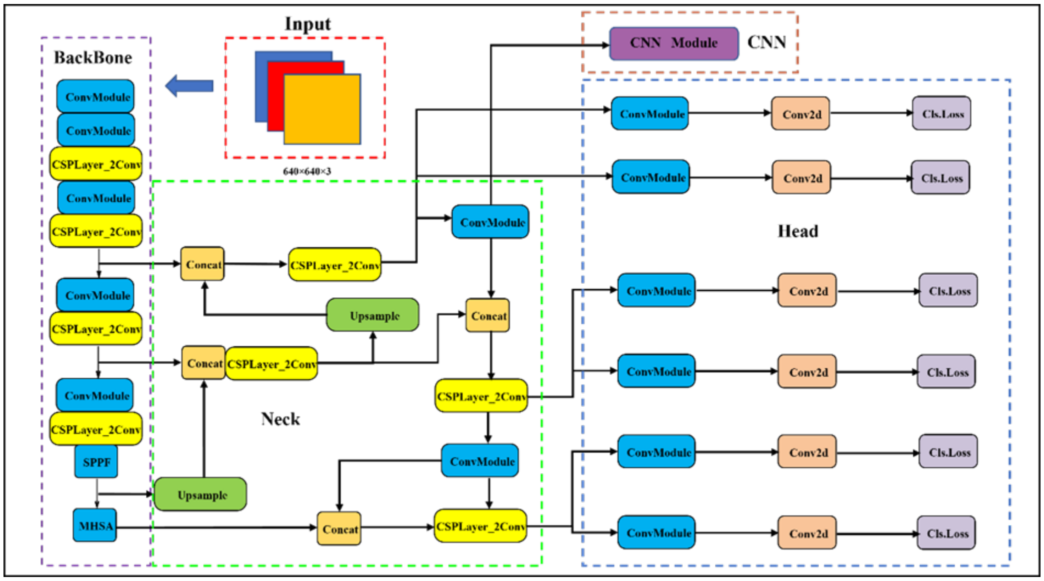


Figure 3.4: Improved YOLOv8 Network Architecture Diagram

From the perspective of network structure, the MHSA module is at the last layer of the backbone network, receiving the thoroughly processed and integrated feature map information and calculating the attention weights. These attention weights are applied to the feature map information received by MHSA to generate weighted feature map information. Subsequently, this weighted feature map information is passed to the Neck network part of YOLOv8 and fused with other feature map information to obtain higher-level feature semantic information. The MHSA layer calculates the attention weights through a special multi-head self-attention mechanism structure and integrates them to the original feature map to emphasize important spatial positions.

The driving emotion detection module belongs to the Head network part of the YOLOv8 network and is located in a dedicated emotion detection layer for emotion classification. According to the network design and relationship between target detection and emotion detection, the position of this emotion detection layer is variable and may be located before or after the target detection layer. This module mainly extracts emotion-related features from the ROI (region of interest) in the improved YOLOv8 network.

In the improved YOLOv8 network structure, ROI is generated by the object detection algorithm layer, which indicates the area where the detected object exists in the input image. In this study, the input to the driving emotion detection module mainly uses ROI information to extract facial emotion features. In short, the driving emotion detection module extracts emotion features from the region of interest of the input image from the YOLOv8 model. Subsequently, the driving emotion detection module analyzes this input ROI information and passes it to the emotion classifier. The emotion classifier maps these processed ROI features to specific emotion categories and outputs the final emotion classification results. .

3.2 Driver Pedal Operation Detection Based on Image Processing

This section introduces implementation of the driver's brake pedal operation detection in the structure implemented. For this purpose, the Mask R-CNN algorithm based on video image processing and instance segmentation target detection network are used as a highly flexible framework in the instance segmentation target algorithm, Mask R-CNN can freely add various branches to perform different detection tasks, such as target detection classification, semantic segmentation, instance segmentation, and human posture estimation. Therefore, the Mask R-CNN network is more suitable for the task of driver's brake pedal operation detection. It was first proposed by He et al.(He et al., 2017) as an advanced instance segmentation model that extended Faster R-CNN. Structurally, Mask R-CNN is a two-stage method. In the first stage, a region proposal network (RPN) is constructed to generate region of interest (ROI) candidates. This is followed by the model to predict the class, bounding box offset, and binary mask of each ROI. In the second stage, the Mask R-CNN also introduces ROI Align, which makes the pixel alignment more efficient and faster during the downsampling process, thereby, enabling faster and more accurate instance segmentation(Zhang, 2018).

In terms of the operation of the Mask R-CNN algorithm, Figure 3.5 shows the main steps of image processing. The image is input to the feature extraction layer and feature map information is generated; the location information of each pixel in the feature map information is set as the region of interest (ROI), also known as anchors; the ROI information is subsequently input to the matching region proposal network (RPN), and refined by binary classification (foreground and background) using coordinate regression methods; the refined ROI is then aligned with the original ROI, so that the pixels in the feature map can be intuitively displayed in the original image; finally, the ROI information is subjected to multi-class classification and bounding box regression methods generate a mask to complete the task of accurate image segmentation using a fully convolutional network (FCN).

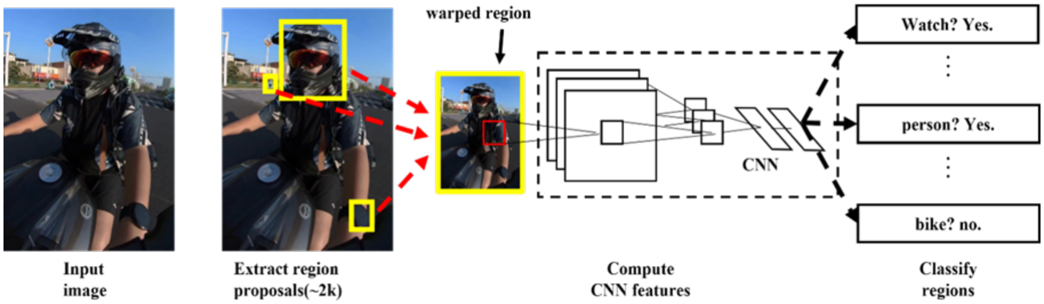


Figure 3.5: Mask R-CNN Algorithm Flowchart

3.3 Dangerous Driving Behavior Detection Based on Vehicle Data

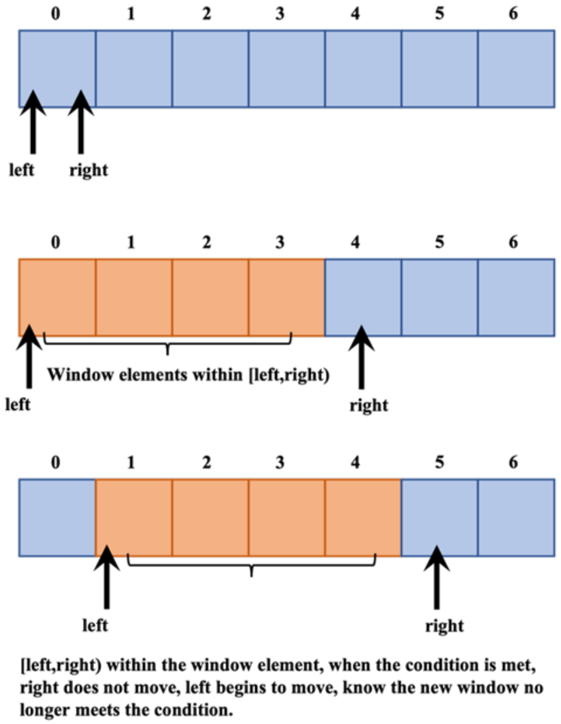
Many existing studies on dangerous driving behavior detection are based on vehicle data analysis. In these studies, speed and time data are usually used to estimate the vehicle acceleration/deceleration within a certain period of time, and acceleration/deceleration is used as the basis for judging and evaluating dangerous driving behaviors (Liang et al., 2025; Xiang et al., 2021). Among them, some of the studies have proposed the threshold and absolute acceleration value based on the vehicle dynamics theory. For example, Zhang et al. set the absolute acceleration threshold limit to 3 m/s^2. The specific setting conditions are: (i) sudden acceleration is judged when $a > 3$ m/s^2; and (ii) is related to sudden braking operation when $a < -3$ m/s^2 (Zhang et al., 2019). This proposed limit contributes a more intuitive detection method that reflects the driver's dangerous operation of the vehicle.

In this study, a sliding window approach is implemented to analyze the handling braking and throttle data for detection, assessment, and evaluations of risky driving attention. The sliding window method, based on a two-point approach, creates a window between two data points. The sliding window is applied to obtain smoothened data, such as the average over a continuous time segment, which helps enhance stability, as seen in applications involving temperature monitoring (Zhang et al., 2025). Figure 3.6 illustrates the general process of executing the sliding window method.

At the initial position, both the ”left” and ”right” data position point to the array element with position information of 0, and a window of [left, right] is created. This is the initial state definition, and the initialized sliding window is a left-closed and right-open interval with an empty value (no elements). Subsequently, the ”right” pointer begins to move, and the algorithm traverses each array element within the window length through the first loop until the traversed moving distance exceeds the array length. The algorithm then exits the loop, and traverses, calculates and counts the data extracted in the window. After the ”right” pointer stops at the end of a loop, the distance between the ”left” and ”right” pointers is exactly the specific length of the sliding window. The ”left” and ”right” pointers start to move sequentially in the chosen direction of sequence increase at the same speed at the same time. At this time, the two pointers maintain a sliding window of a specific length to traverse, calculate and count each array sequence in it. In simple terms, this means that within a certain length of time period, the relationship between the vehicle speed and time change is used to infer the continuous instantaneous acceleration, so that the average value of the acceleration in the continuously updated time interval is compared with the threshold value, and finally the evaluation result of the degree of danger of driving behavior is output.

This study used the sliding window method to identify, analyze and evaluate the driver’s driving behavior of brake and throttle operation using the brake and throttle data signals acquired from the vehicle CAN during driving. As the main communication line for various vehicle sensors and on-board computers, the CAN line has provided high real-time vehicle state data until a sampling interval for brake and throttle data reaches in the order of 10 microseconds. Owing to relatively short sampling interval sliding window method is considered to be more suitable for this type of data processing, thereby improving real-time performance.

Figure **??** illustrates the data process, including analyzes and evaluations of the CAN line output data using the sliding window method acquired. Firstly, the brake and throttle data, input from the CAN line, are preprocessed. Preprocessing method selected here is used to preprocess the data, which is permit treatment of missing values, NaN and infinite values to ensure continuity and readability of the entire data. Assume that there exist two known data points (x_1, y_1) and (x_2, y_2), where $x_1 < x_2$, which means that two data points at two different time scale positions are randomly selected in the data sequence. These two points are used to construct a linear line through linear interpolation to estimate the coordinate values (x, y) of the data points in all data series between x_1 and x_2. The linear line equation constructed by the linear interpolation formula can be expressed as:



[left,right) within the window element, when the condition is met, right does not move, left begins to move, know the new window no longer meets the condition.

Figure 3.6: An Illustrates of the Sliding Window Method Execution Process

$$y = y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)} \cdot (x - x_1) \tag{3.11}$$

where, y is the estimated value of the unknown data point that needs to be estimated, x is the x desires of the unknown data point.

At this point, if only the above method is used to process the data, the generated data will be distorted. The low-pass filter can be used to effectively restore the true value in the data sequence and eliminate unnecessary high-frequency noise in the data. Therefore, after restoring the true value, a low-pass filter is used to restore the data and eliminate the high-frequency noise that is not needed in this study can be eliminated. The low-pass filter function used in the study is given by:

$$Y(n) = a \cdot X(n) + (1 - a) \cdot Y(n - 1) \tag{3.12}$$

where, X is the input signal, and Y is the filtered output. a is the filter coefficient, which controls the threshold value of the sample value passing through the filter. In layman's terms, the smaller the filter coefficient a, the smoother the output will be, while the sensitivity will be reduced; the larger the filter coefficient a, although the sensitivity is increased, the output will be less smooth.

Considering that the input data sequence of the current data processing cannot allow the data sequence to oscillate too much, the median filtering is adopted to process the data so that the data can be effectively extracted. Median filtering is a kind of nonlinear signal processing technique based on statistical sorting theory. Generally, the median filtering is used to process data with large oscillations and eliminate the noise. The median filter can be represented by the following relationship:

$$g(x, y) = \text{med}\,\{f(x - m, y - n), (m, n \in w)\} \tag{3.13}$$

where, $f(x, y)$ represents the original image and $g(x, y)$ represents the processed image.

The data are normalized to fall within the range of 0 to 1. Normalization is a process that transforms data, typically in a specified range and removes dimensions to ensure desired unit seized to unites and scales limit. Typical ranges for normalization are pre-bounded in the [0, 1] or [-1, 1] ranges. The most common normalization method is Min-Max normalization (Ali, 2022). Min-Max normalization, also known as min-max scaling, is a kind of normalization specified to transform action of the original data to mapping to the [0, 1] range, such that:

$$x_{\text{new}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{3.14}$$

where, x_{max} and x_{min} represent the maximum and minimum values of the sampled data.

Following interpolation, filtering, normalization, and detrending of the brake and throttle signal data, subsequent evaluation and analysis can be conducted. The input parameters related to the degrees of brake and throttle (ranging from 0 to 1) are subsequently expressed into two boolean arrays. For the chosen sliding window size, the data analysis is preboned considering sampling rate of 10Hz, throttle threshold of 0.8, and brake threshold of 0.5, as recommended in (Miyajima et al., 2011; Deligianni et al., 2017). The sliding window method is used to assess whether the degrees of brake and throttle input degrees exceed the safe threshold values. If the thresholds are exceeded, a dangerous driving behavior, such as sudden acceleration or sudden deceleration, is judged when the degrees of either brake or throttle input exceed the correspondence threshold values.

Chapter 4

Training and Data Analysis

This section describes the training and verification process of the multivariate data-driven driver dangerous driving behavior detection model proposed in this study. In order to ensure the availability of the training dataset, a variety driving scenarios were considered in a laboratory environment to synthesize the dataset; used the FER2013, public emotion dataset for grayscale image processing (Goodfellow et al., 2013); The publicly available data relevant to drivers' emotion, were also used for training. In addition to these, the datasets collectively integrated and analyzed the available public datasets and the actual vehicle data were obtained from Yutong Bus Company, and finally formed a comprehensive driving behavior dataset of multivariate data. This operation enhanced the generalization ability and robustness of the model during the training process. Secondly, the software and hardware platforms used in this experiment were chosen. These included the settings of relevant deep learning frameworks and auxiliary tools.

In the model performance evaluation, this study employed accuracy, recall and F1 value as evaluation indicators, moreover, average precision (AP) and average accuracy (mAP) were used to evaluate the accuracy of model detection. The above evaluation indicators basically cover the performance level of the model algorithm in detection, classification and output tasks, which greatly improve the credibility of the final detection outcome.

4.1 Dataset

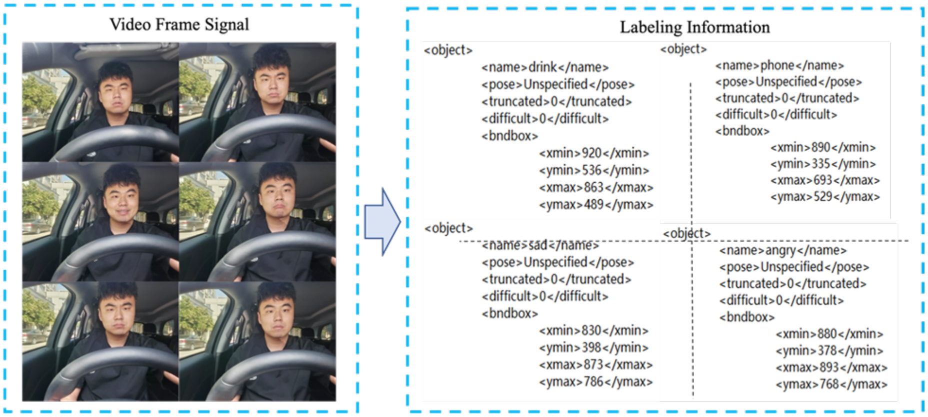
The multi-source data-driven driver dangerous driving behavior detection method proposed and designed in this study offers advantages in detection function diversity and data diversity. In order to comprehensively evaluate the feasibility and advantages of the detection method, it is considered to use a variety of different types of data sets to train, verify and evaluate the detection algorithms of each part. However, the existing available public data sets are limited and cannot meet the needs of this experiment for multiple types of data sets. The data sets covered by most existing detection methods are not consistent with the data sets required by this experiment. In order to solve this problem, this experiment analyzes, classifies and integrates different types of data, and finally forms a data set for the dangerous driving behavior detection method of multi-source data. Specifically, this multi-source data set includes self-made data sets, public data sets and real vehicle data provided by Yutong Bus Company. Among these, the self-made data set can ensure that it maintains high diversity and authenticity while fully meeting the requirements of this experiment. This self-made data set considers a variety of simulated driving behavior scenarios in a laboratory setting, such as fatigue driving, drunk driving, smoking while driving, mobile phone use, etc. in the laboratory environment, and obtains representative driving behavior data through analysis and integration under controlled conditions. The public data set has a more complete data integration, with a rich number of data samples and a variety of types, which makes the coverage of driving behavior type data larger. The real vehicle data set provided by Yutong Bus Co. was real vehicle driving data collected and integrated from the CAN line, which could greatly improve the application ability of the detection method in actual scenarios. This experiment integrated the above data sets and jointly trained the detection models of each part, further improving the performance and stability of the integrated model.

4.1.1 Driving Behavior Datasets

Owing to the limited availability of diverse driver behavior datasets in the reported studies, this paper adopts a homemade dataset in order to meet the experimental needs. This study considered a dataset synthesized in the laboratory under carefully controlled scenarios. Data collection was conducted in a laboratory setting using video streaming and image information from cell phones. A total of 20 subjects participated in the data collection process that involved different distracted driving scenarios, including fatigue driving, beverage consumption, or smoking or phone usage during driving.

In the process of dataset preparation, this study fristly used the software platform described in (Zhang et al., 2017) to analyze and annotate the image data collected for each driving scene, and generates the XML (Extensible Markup Language) files required by the detection method model. The information in these generated XML files reflects the location, classification features and label information of the detection target in each image. Figure 4.1 shows the process of using the LabelImg platform to annotate the collected image dataset to form a dataset. Each XML file describes the location information, classification label and feature data of the driver behavior detection target in the image with detailed data of different values, providing an important data basis for subsequent data processing and analysis. In fact, this dataset basically covers all the dangerous driving behavior characteristics that could occur in the simulated actual driving environment, such as driver fatigue driving and drinking, smoking, and using mobile phones while driving.

This section mainly described the data collection of fatigue and distracted driving behaviors simulated in actual driving scenarios in a laboratory environment and classifies and annotates them to form a self-made driving behavior dataset. The dataset provided an experimental basis for the comprehensive detection method of fatigue and distracted driving behaviors.



```
<object>
        <name>drink</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
                <xmin>920</xmin>
                <ymin>536</ymin>
                <xmax>863</xmax>
                <ymax>489</ymax>
```

```
<object>
        <name>phone</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
                <xmin>890</xmin>
                <ymin>335</ymin>
                <xmax>693</xmax>
                <ymax>529</ymax>
```

```
<object>
        <name>sad</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
                <xmin>830</xmin>
                <ymin>398</ymin>
                <xmax>873</xmax>
                <ymax>786</ymax>
```

```
<object>
        <name>angry</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
                <xmin>880</xmin>
                <ymin>378</ymin>
                <xmax>893</xmax>
                <ymax>768</ymax>
```

Figure 4.1: Sample Image and XML File Generated by LabelImag Software

4.1.2 FER2013 Emotion Datasets

The driver emotion detection datasets used in this study were derived from the publicly available dataset FER2013 (Ko, 2018), a dataset widely used for the training and evaluation of Facial Emotion Recognition (FER) algorithms, as shown as a partial example of the dataset in Figure 4.2. The introduction of the FER2013 datasets provided an important benchmark for emotion recognition together while facilitating the developments in evaluation algorithms. The FER2013 dataset contained a variety of images of human facial expressions, which are considered as indicators of different human emotional states.

The FER2013 dataset mainly contains seven different label information, namely anger, disgust, fear, happiness, neutral, sadness and surprise. These seven different labels basically cover the emotions that drivers may have in daily driving and are highly representative. Each image label file in the FER2013 dataset has a clear emotion classification label, which makes it an ideal dataset for training and evaluating emotion recognition algorithms.

Overall, the FER2013 dataset was an emotionally rich facial emotion recognition base, and its diversity and labeling information provided important support for the research and evaluation of emotion recognition algorithms. The dataset can be used to better understand and solve emotion recognition problems and promote the developments in elective detection algorithm.

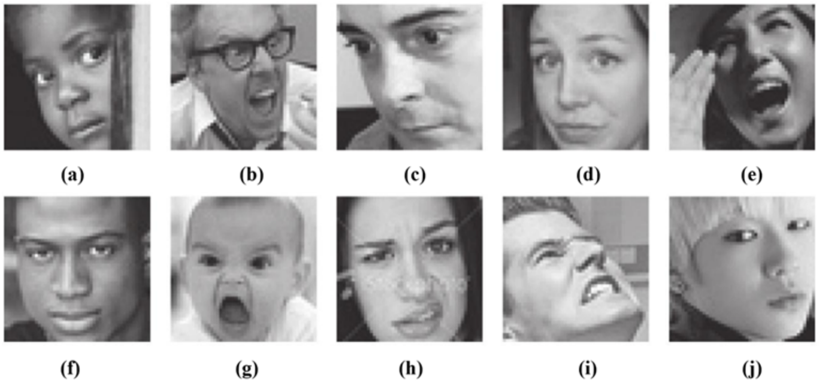


Figure 4.2: Example Diagram of A FER2013 Dataset

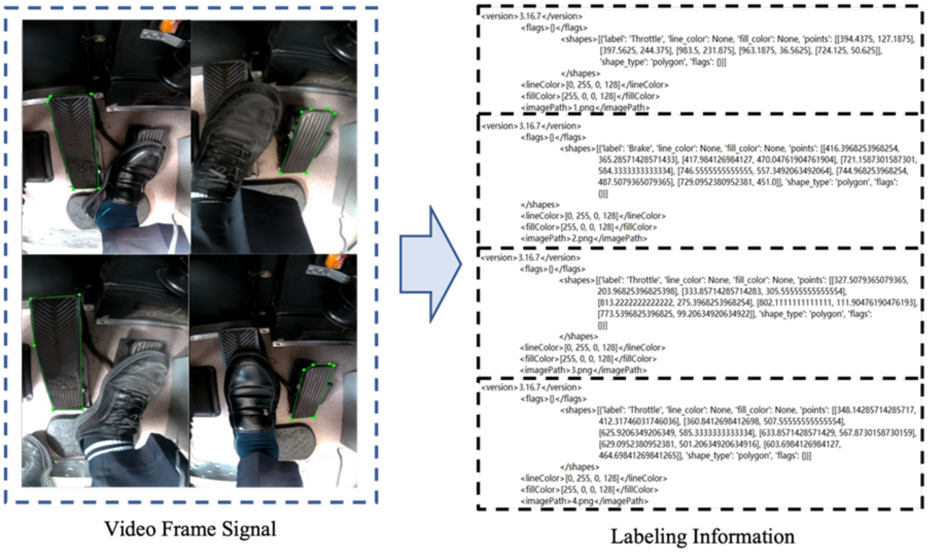
4.1.3 Driver’s Pedal Operation Dataset

The field of automotive safety and driving behavior recognition research is particularly important to analyze drivers’ brake and gas pedal operations. These actions directly reflect the driver’s driving intentions and behavioral patterns and are important for understanding driving behavior, predicting potentially dangerous actions, and improving road traffic safety. Despite the obvious practical value and application prospects, only a few studies have explored drivers’ brake and gas pedal operations in the context of risky driving behavior detection. Moreover, there exists a lack of publicly available driving datasets containing detailed pedal operational data, which further limits the consideration of pedal operations by the driver’s research.

In collaboration with Yutong Bus Co., Ltd., for this purpose, an experimental study has been carried out which enabled us to utilize the company’s facilities and acquire video frames of the positions of driver’s gas pedal and brake pedal in real-time driving situation directly from the company. The proposed acquisition method offered highly accurate and realistic data, while the data were more likely to reflect the driver’s real driving behavior and operating characteristic since the data were collected during the real driving process. The acquired video frame signals were firstly pre-processed with permitted LabelMe (Russell et al., 2008) software to pre-process these video frame signals. LabelMe provided an intuitive interface which enabled the researchers to do the target labeling for every frame of the video.

Figure 4.3 illustrates an example of the labelImg process (assuming there is a diagram or image showing the labeling process using LabelMe). In the Labelling process, the LabelMe platform allows the user to draw the boundaries of the objects by mouse and assign a label to each labelled object. In addition, LabelMe platform also allows the user to modify any necessary adjustments to the labels (e.g. modify the shape of the label) to ensure that the label describes the target object as accurately as possible.

After the labelling process is completed, LabelMe platform will automatically generate a JSON file alter completism the LabelImg process that contains all information of the labelling. This JSON file is very important because the information of the location, shape, and category of each object are described in detail in this JSON file. Specifically, this JSON file will contain various data, such as



```
<version>3.16.7</version>
          <flags>{}</flags>
                    <shapes>[{'label': 'Throttle', 'line_color': None, 'fill_color': None, 'points': [[394.4375, 127.1875],
                              [397.5625, 244.375], [983.5, 231.875], [963.1875, 36.5625], [724.125, 50.625]],
                              'shape_type': 'polygon', 'flags': {}}]
                    </shapes>
          <lineColor>[0, 255, 0, 128]</lineColor>
          <fillColor>[255, 0, 0, 128]</fillColor>
          <imagePath>1.png</imagePath>
<version>3.16.7</version>
          <flags>{}</flags>
                    <shapes>[{'label': 'Brake', 'line_color': None, 'fill_color': None, 'points': [[416.3968253968254,
                              365.28571428571433], [417.984126984127, 470.04761904761904], [721.1587301587301,
                              584.3333333333334], [746.5555555555555, 557.3492063492064], [744.968253968254,
                              487.5079365079365], [729.0952380952381, 451.0]], 'shape_type': 'polygon', 'flags':
                              {}}]
          </shapes>
          <lineColor>[0, 255, 0, 128]</lineColor>
          <fillColor>[255, 0, 0, 128]</fillColor>
          <imagePath>2.png</imagePath>
<version>3.16.7</version>
          <flags>{}</flags>
                    <shapes>[{'label': 'Throttle', 'line_color': None, 'fill_color': None, 'points': [[327.5079365079365,
                              203.96825396825398], [333.85714285714283, 305.55555555555554],
                              [813.2222222222222, 275.3968253968254], [802.1111111111111, 111.90476190476193],
                              [773.5396825396825, 99.20634920634922]], 'shape_type': 'polygon', 'flags':
                              {}}]
                    </shapes>
          <lineColor>[0, 255, 0, 128]</lineColor>
          <fillColor>[255, 0, 0, 128]</fillColor>
          <imagePath>3.png</imagePath>
<version>3.16.7</version>
          <flags>{}</flags>
                    <shapes>[{'label': 'Throttle', 'line_color': None, 'fill_color': None, 'points': [[348.14285714285717,
                              412.31746031746036], [360.8412698412698, 507.55555555555554],
                              [625.9206349206349, 585.3333333333334], [633.8571428571429, 567.8730158730159],
                              [629.0952380952381, 501.20634920634916], [603.6984126984127,
                              464.69841269841265]], 'shape_type': 'polygon', 'flags': {}}]
                    </shapes>
          <lineColor>[0, 255, 0, 128]</lineColor>
          <fillColor>[255, 0, 0, 128]</fillColor>
          <imagePath>4.png</imagePath>
```

Figure 4.3: An Example Illustration of the Labelling Process in the Platform LabelMe

the coordinate of each vertex of the polygon labelled by the user, the name of the label, and other possible metadata information. Because this information is very important for training the deep learning model, all information describes the essential and precise target and context information for the training of deep learning model.

In the final step, Mask R-CNN deep learning model requires the JSON files to be coinvested into the more general or framework-specific COCO data format (Carvalho et al., 2020), so the JSON files could be generated by LabelMe into COCO format. The COCO format is a standard data format that is widely used in computer vision tasks. Especially, this data format is widely used in target detection, segmentation, and classification tasks. All of the information in the original annotation information (such as the coordinate of each vertex of the polygon) will be transformed into the data structure required by the COCO standard, such as object bounding boxes, segmentation masks, and category IDs. We can write the scripts or search the existed tools to transform the data format. Labeled data will be exported and used in Mask-RCNN neural network model training.

4.1.4 Vehicle Data Dataset

The driving behavior dataset used in this study was obtained from real vehicle driving data collected in cooperation with Yutong Bus Co., Ltd., which includes brake and throttle signals on the CAN line. The collected vehicles and equipment are shown in Figure 4.4. For driving behavior detection based on vehicle driving data, real vehicle driving data has higher authenticity and can reflect driving behavior data under real driving conditions to the greatest extent.

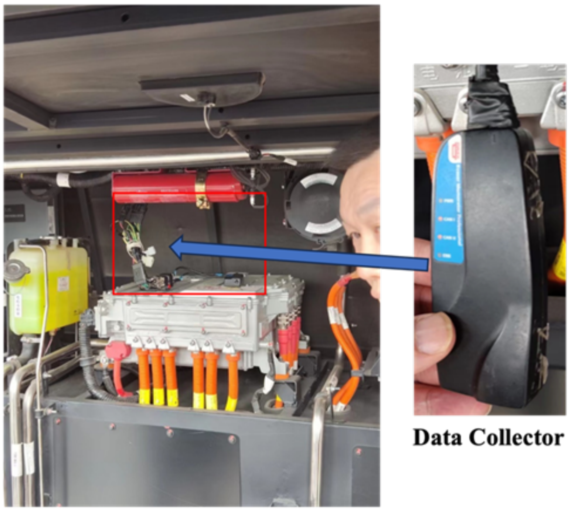


Figure 4.4: Vehicle CAN Data Collector

The production of this dataset is necessary to extract, translate and process the real driving data collected directly from the CAN line, because these data usually have problems such as missing values, outliers and noise (Buccafusco et al., 2021; Purohit and Govindarasu, 2022). In this process, this experiment performed a series of preprocessing operations on its data sequence. The theoretical basis of preprocessing has been introduced in detail in Chapter 2. The specific setting process in the code was subsequently petuned to convert the collected brake and throttle signals into pandas. Series objects for analysis and translation; use linear interpolation to fill the gaps in the data and remove non-numeric values NaN and infinite values in the data; use the scipy.signal.filtfilt function to low-pass filter the signal, restore the data to its true state, and remove unnecessary high-frequency noise in the system; use the scipy.signal.medfilt function to median filter the signal to remove mutations or outliers in the signal, making the data smoother and more stable; finally, normalize and standardize the obtained data series to reduce its range to between 0 and 1. Through the above series of preprocessing steps, high-quality driving data and input it into the detection model was obtained for further judgment and evaluation. Figure 4.5 shows a flowchart of the preprocessing data changes applied to the vehicle CAN bus data, which could intuitively understand the effect of this process on data processing.

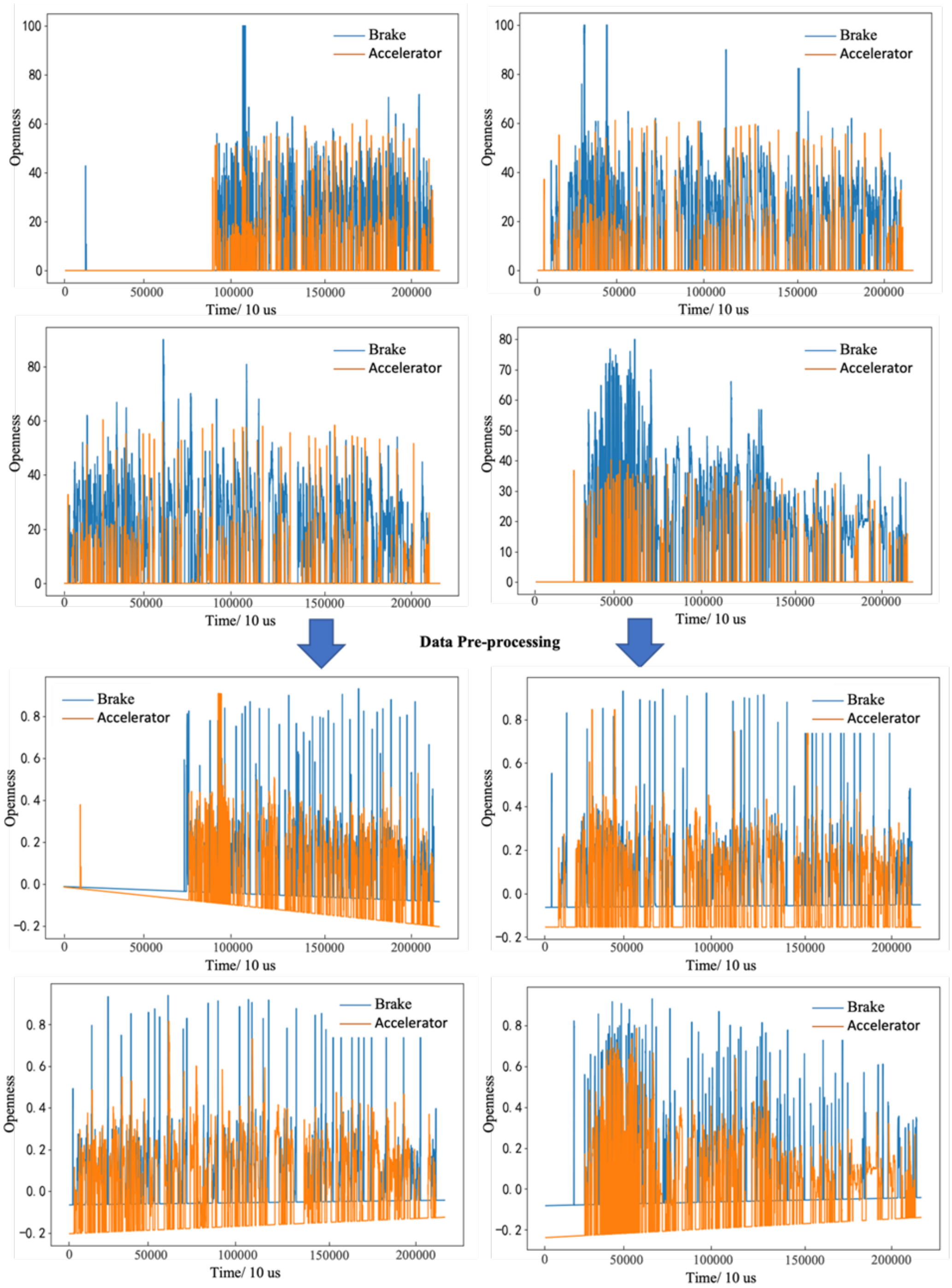


Figure 4.5: Vehicle CAN Data Collector

4.2 Experimental Environment and Evaluation Indicators

This section provides a detailed introduction to the experimental environment configuration and the selection of evaluation indicators for this experiment. These are crucial prerequisites for the training and verification of deep learning network models and play a very important role in the smooth progress of the entire experiment, the generation of results, and the performance evaluation. First, the platform and environment configuration of the hardware and software used in this experiment are introduced. In the training and evaluation of deep learning models, the selection of appropriate hardware equipment could be largely guaranteed for the speed and accuracy of model training. At the same time, the software environment ensures the feasibility of model training. In addition, the selection and setting of initial training parameters and optimizers will directly affect the training effect and performance of the model, so they need to be carefully considered and adjusted. Secondly, the evaluation indicators of the model training results of this experiment are introduced in detail. The results were obtained from model training require appropriate evaluation indicators for analysis and evaluation. These evaluation indicators could intuitively and effectively show the trend of various parameter results of the model during the training process, including the trend of the loss value with the number of training rounds and the performance of the model on the validation set.

4.2.1 Experimental Environment

In this study, a desktop computer with a high level of hardware configuration was used as the primary experimental platform. The required deep learning framework was configured on this experimental platform to meet the hardware and software requirements of the experimental environment, as shown in Table 4.1, from which the relevant configuration of the hardware and software was shown in the experimental conditions, as well as the initialization of the functions required for training settings.

In terms of hardware, the hardware used in this experiment had the following parameters: CPU: i7-13700k, GPU: NVIDIA-RTX-4070Ti, Memory: 16GB; in terms of software, the training platform is PyCharm, the programming language is Python, and the deep learning frameworks are: Pytorch, Opencv-Python (cv2), Tensorflow-GPU, Scipy, Pillow, Keras, Pandas, Numpy, Seaborn;

Table 4.1: An Illustration of The Experimental Environment

Experimental Condition	**Related Configuration**
Hardware	CPU: i7-13700k GPU: NVIDIA-RTX-4070Ti Memory: 16GB
Software	PyCharm, Python, Pytorch OpenCV-Python (cv2), Tensorflow-GPU Scipy, Pillow, Keras Pandas, Numpy, Seaborn
Training Setup	Batch Size: 16 Epoch: 300 Optimizer: SGD Learning Rate: 1×10^{-4} Detector: Valid loss

the training initialization parameters are set to batch size 16, epoch to 300, optimizer to SGD, learning rate to $1{\times}10^{-4}$, and detector to valid loss. The reasonable configuration of the necessary basic was conditioned in the above three aspects provides important guarantees for the subsequent training and evaluation of network models.

4.2.2 Evaluation Indicators

In order to accurately evaluate the accuracy of the model of the driver's dangerous driving behavior detection method proposed in this study, the analyses and evaluations of the model training results in this experiment was used by precision, recall and F1 value as measurement indicators (Chicco and Jurman, 2020). The above evaluation indicators could comprehensively analyze and evaluate the model training process, data set verification and detection effect from different aspects. Specifically, precision could be very intuitively reflected the detection accuracy of the model on the target data set during the training and verification process, which referred to the ratio between the number of correctly classified positive samples and the total number of classified positive samples; recall referred to the performance of whether the model could identify the detection target in the image, which referred to the ratio between the number of positive samples correctly classified by the model and the number of classified positive samples; and F1 value was the harmonic mean of the above two data, which could be very intuitively reflected the true value of the detection target being correctly classified. The three basic evaluation indicators could display the detection performance of the model in real time during the model training and verification process, and timely reflect the problems existing in the model during the training process. This could be enabled to adjust the model structure and training parameters in a timely manner, thereby improving the effectiveness of model training and verification.

The specific calculation method is summarized below:

$$\text{Precision}(P) = \frac{TP}{TP + FP} \tag{4.1}$$

$$\text{Recall}(R) = \frac{TP}{TP + FN} \tag{4.2}$$

$$F_1\text{-score} = \frac{2PR}{P + R} \tag{4.3}$$

where P is the precision rate, R is the recall rate, and F_1 is the harmonic mean of the precision rate and recall rate. TP denotes a correctly detected positive sample, FP denotes a false detection, FN denotes a missed detection, and TN denotes a correctly detected negative sample. The specific representation is shown in Table 4.2.

Table 4.2: Summary of Specific Indicators

Label	**Prediction result**	**Real result**	
		Positive sample	**Negative sample**
Prediction result	**Positive sample**	TP	FN
	Negative sample	FP	TN

$$AP_i = \int_0^1 P_i(R_i)\, dR_i \tag{4.4}$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{4.5}$$

where AP (Average Precision) is the average precision, a metric commonly used in target detection tasks.

In the target detection task, the model generates a series of bounding boxes and predicts for each one whether it contains a target and the class of the target. AP evaluates the performance of the model by calculating the area under the precision-recall curve. Based on the prediction result, they are sorted by confidence level. The precision and recall are calculated for different confidence levels. Finally, the area under the precision-recall curve is integrated and calculated to obtain the AP value. mAP (mean Average Precision) is the mean value of average precision, which are usually used to evaluate the model performance in multi-category target detection tasks. In the multi-category target detection task, each category exhibits a corresponding AP value, and mAP is the average of the AP values of all categories.

Judging from the numerical values of the evaluation indicators, in general, the recognition accuracy of the trained detection model for each category of target objects depends on the size of these two values. Higher AP and mAP values can reflect that the trained model has a stronger accuracy and robustness in detecting target objects, and the model can more accurately identify each category of detection targets.

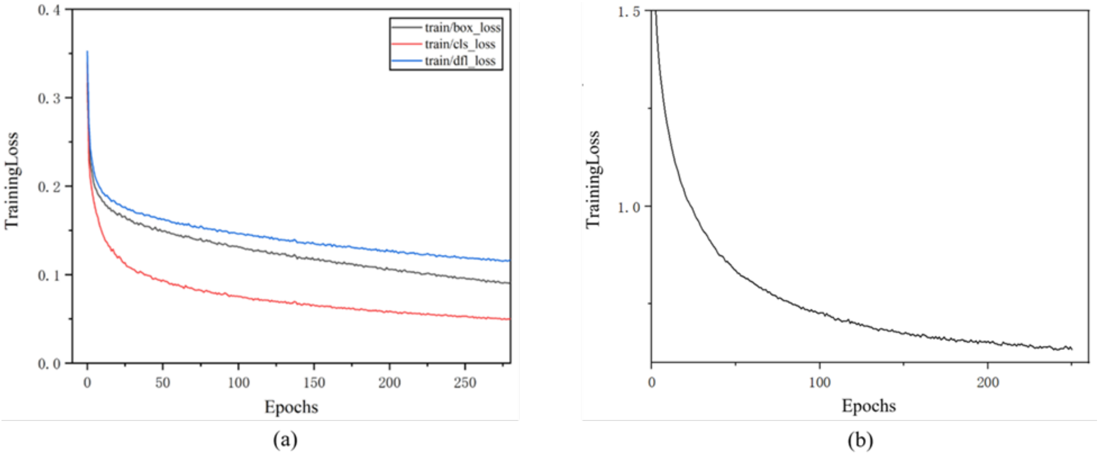
4.3 Experimental Results and Analysis

4.3.1 Improved YOLOv8 Model Training

In this experiment, each category label sample in the data set was randomly assigned to a training set and a test set with a ratio of 9:1. In order to avoid overfitting of the model training effect, the SGD (stochastic gradient descent) optimizer was employed to set the model training learning rate decay mode to cosine to control the model learning rate at different stages of training. In terms of the initialization parameter setting of the model training, the initial learning rate of the model training was set to 0.01, the image input size was adjusted to 640×320 pixels, and the training step size of the batch samples was set to 16 to ensure the stability and convergence of the model. In the model compilation stage, the Adam optimizer and cross-entropy loss function commonly used in model detection classification training tasks were used. The Adam optimizer could dynamically adjust relevant training parameters (such as learning rate) in real time according to different data distributions and training stages during the model training process (Kingma and Ba, 2014). The cross-entropy loss function could avoid the model fitting phenomenon during the training process to the greatest extent, improving the readability of the training data set and the model detection classification performance (Mao et al., 2023). During the model training process, multi-threaded data reading technology was used to improve the speed of data interaction and throughput and model iteration and update during the training process, thereby accelerating the convergence of model training results.

The training results, shown in Figures 4.6, demonstrate the change of loss values with epochs. For the driving behavior detection task, the training loss is decomposed into total loss and sub-losses such as box, cls, dfl, etc., as shown in Figure 4.6(a). From the figure, it could be observed that the individual loss and total loss decrease rapidly with the increase of epochs, which indicated that the improved YOLOv8 model achieves good convergence on the driving behavior detection task.

In addition, during the training of the driving emotion detection module, as the number of iterations increases, the model training loss decreased rapidly and gradually converges. Figure 4.6(b) shows very intuitively that the driving emotion detection module proposed in this study shows strong learning ability, convergence and detection accuracy in the training of the data set.



(a)

(b)

Figure 4.6: Improved YOLOv8 Neural Network Model Loss Function Plot

Figure 4.7 shows that the model accuracy is constantly updated and improved as the number of training iterations increases. For the distracted driving detection task, Figure 4.7(a) also shows that the accuracy increases with the number of iterations (epochs), which indicates that the improved YOLOv8 model achieves good accuracy in the driving behavior detection task. On the other hand, in the training of the emotion data subset, the accuracy also increases with the number of iterations, as shown in Figure 4.7(b). This figure shows the accuracy of the method for facial emotion recognition using the proposed CNN module. Finally, after model fusion, the final accuracy of the improved YOLOv8 neural network-based model proposed in this chapter approached 81.4%, which indicates that the model achieves better results while realizing the detection purpose.

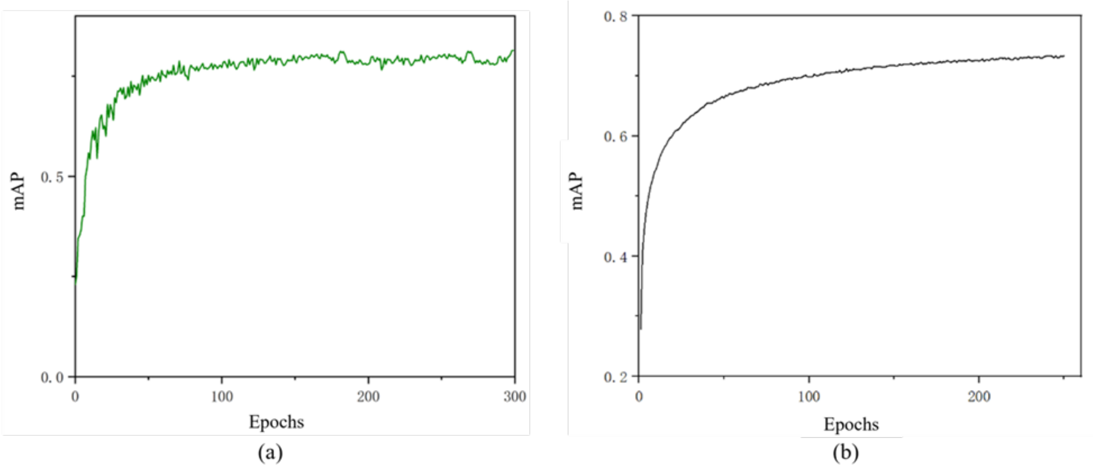
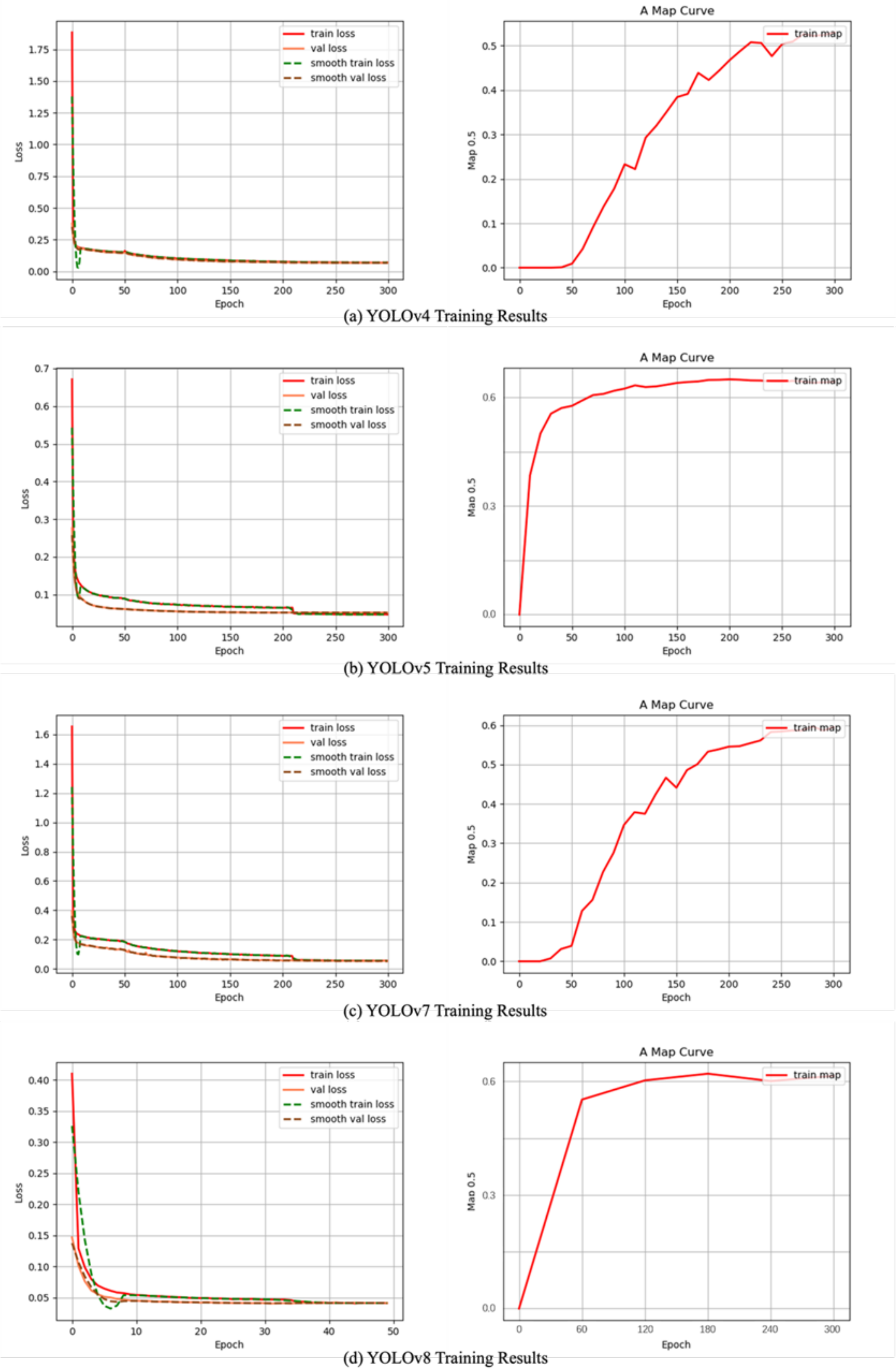


Figure 4.7: Prediction Accuracy of The Improved YOLOv8 Neural Network Model with Increasing Number of Epochs Plot

Figure 4.8(a) shows the training results of the YOLOv4 neural network model. The results show that the loss function image decreases and flattens out with increase in the number of training iterations (Epoch), which indicates the convergence of the YOLOv4 neural network model; moreover, the accuracy increases and flattens out and finally reached 53.1%. Figure 4.8(b) shows the training results of the YOLOv5 neural network model, in which it could be clearly seen that with the increase of the number in training iterations (Epoch), cause graded decrease in the image of the loss function gradually decreases and tends to level off. The results also show the convergence of the YOLOv5 neural network model; and that the accuracy increases gradually and tends to level off, and finally reaches 75.7%, which is a more accurate result. As shown in Figure 4.8(c), the training results of the YOLOv7 neural network model reveal a loss function curve comparable to those of the YOLOv4 and YOLOv5 models under identical conditions, further confirming the convergence capability of YOLOv7; however, the accuracy of YOLOv7 neural network model is poorer, only reaching 59.5%. Figure 4.8(d) shows the training results of the original YOLOv8 neural network



(a) YOLOv4 Training Results

(b) YOLOv5 Training Results

(c) YOLOv7 Training Results

(d) YOLOv8 Training Results

Figure 4.8: YOLO Series of Neural Network Model Training Results Plot

model without improvements, from which it could be observed that the model exhibits smooth convergence and achieves a high accuracy, ultimately reaching 0.727.

As a result, the training results of the YOLO neural network model series were obtained using identical datasets, hardware and software environments, initial parameter settings, and training optimizer on the experimental platform. The loss function curved of these models tend to flatten as the number of training iterations (epochs) increases, indicating good convergence. This demonstrated that the YOLO model series is well-suited for addressing the detection task proposed in this study, and that the training-related configurations are appropriately selected. As shown in Figure 4.9, the training accuracy results of the YOLO model series and the improved YOLOv8 model proposed in this study are summarized. As shown in the image comparison, the improved YOLOv8 neural network model proposed in this study achieves the highest accuracy (81.4%). Specifically, it exhibits improvements of 53% over YOLOv4, 7.5% over YOLOv5, 36% over YOLOv7, and 12% over the unimproved native YOLOv8. These results indicate that the proposed model is more effective in performing the detection task addressed in this study, achieving superior accuracy compared to other YOLO variants. From the above results, the improved YOLOv8 network model proposed in this paper has higher accuracy, better performance and robustness than other YOLO series models.

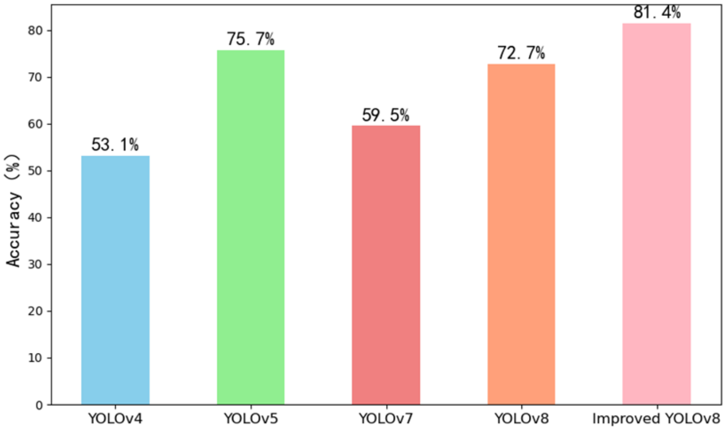


Figure 4.9: Comparison of YOLO Series Model Training Accuracy Results

4.3.2 Mask-RCNN Model Training

In the training of the Mask-RCNN detection model, the integrated dataset was randomly assigned to the training set and the test set in a ratio of 9:1 in the same way as above. Unlike YOLOv8, this model comprehensively considered the losses of target detection, instance segmentation, mask and region proposal network (RPN). In addition, the pre-trained weights were used as the initial training weights in this training task in order to speed up the convergence of the model training and adapt to the adaptability of the training dataset. In terms of model compilation and reading, the SGD (stochastic gradient descent) optimizer was used to set the model training learning rate decay mode to cosine, which was used to control the model learning rate at different stages of training. Such measures were used to avoid overfitting or convergence to local optimality. In the initialization parameters of model training, the initial learning rate was also set to 0.01 and the batch size is set to 16, but the image input size was different and set to 640×640 pixels. In addition, multi-threaded data reading technology was also used in this training task, which enabled the model to extract and analyze data sequences in parallel during the training process, improving the overall efficiency of model training. The Adam optimizer and cross entropy loss function were also selected in the model optimizer selection. This could help the model to dynamically adjust the learning rate during training to better adapt to different data distributions and training stages; the training data could be better fitted and improve the performance of target detection and instance segmentation.

In terms of model structure, the specific network structure of the Mask-RCNN model was not the same as that of the YOLOv8 model. The Mask-RCNN model requires a deeper convolutional layer or a larger perception domain to extract more refined data features because it needs to accurately crop the border of the detection target. At the same time, the losses of target detection of the model should be comprehensively considered, instance segmentation, mask and region proposal network (RPN). Therefore, this experiment adopts a series of specific strategies to enhance the performance and stability of the model.

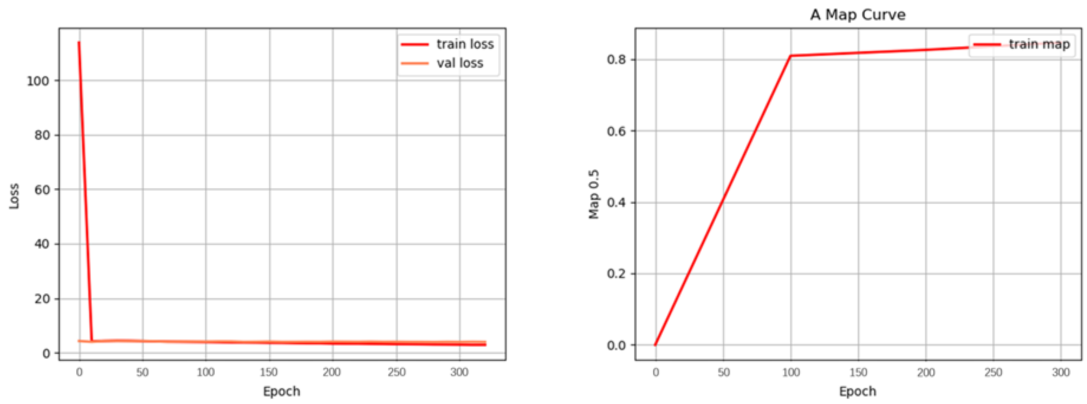
Firstly, appropriated weight data could be assigned to the loss items in each network structure to weight the loss value, so that each layer in the model structure could pay better attention to each classification task and balance the impact of different types of loss values on model training;

a multi-task and multi-threaded training and learning framework was used to integrate and optimize target detection, mask segmentation, and RPN loss. When balancing the target detection loss $L_{\text{detection}}$, mask loss L_{mask} and RPN loss L_{RPN} in the Mask-RCNN model, the total loss L_{total} could be expressed by the following equation:

$$L_{\text{total}} = \alpha \cdot L_{\text{detection}} + \beta \cdot L_{\text{mask}} + \gamma \cdot L_{\text{RPN}} \tag{4.6}$$

where α, β, and γ are the weights of the respective losses, which are used to balance their impact on model training. Specifically, these three weights determine the proportion of each loss data in the total loss data. Adjusting the numerical coefficients of these three weights can keep the total loss data optimal during the training process and ensure the quality of model training.

The training results are shown in Figure 4.10(a), which demonstrates the variation of loss values with epochs, and in Figure 4.10(b), the results show that the accuracy (mAP) increases as the number of iterations goes up, and finally the accuracy of the final example segmentation Mask-RCNN neural network model reaches 84.6% after an iterative process of 300 Epochs. These observations demonstrate the effectiveness of the proposed improved Mask-RCNN model in target detection and instance segmentation tasks.



(a) (b)

Figure 4.10: Example Segmentation Mask-RCNN Training Result Plot

4.3.3 Vehicle Data Algorithm Validation

In this study, brake and accelerator pedal signals are important detection objects in vehicle driving data detection. This study proposed a driving data detection method based on a sliding window method, focusing on the detection, analysis and evaluation of brake and accelerator signals. This method aimed to analyze and evaluate the driver's dangerous driving behaviors, such as sudden acceleration and deceleration, by analyzing these key signals in real time. By configuring the appropriate window size and step size, this study could effectively capture the instantaneous changes of brake and accelerator signals, thereby realizing real-time identification of rapid acceleration and deceleration behaviors.

In order to present the detection results more intuitively, Figure 4.11 shows the processing analysis and threshold comparison of the actual vehicle driving data. For example, as shown in Figure 4.11(a), the visualization results are divided into four parts: the first two parts show the signal values of the brake and accelerator opening, and the last two parts show the emergency braking and acceleration events based on the predetermined thresholds (the emergency braking threshold is 0.5; the emergency acceleration threshold is 0.8) (Zhihu, 2023). When the signal value of the brake or accelerator opening exceeds the corresponding threshold within the sampling time interval, the corresponding indicator in the visualization interface will become 1, which intuitively indicates that a dangerous driving behavior has occurred.

The detection method based on sliding window method proposed in this study shows high accuracy and real-time performance in experiments and tests of actual driving vehicle data sets. This method has stronger adaptability to the rapid changes of vehicle data transmitted by vehicle CAN line with short sampling time and can more accurately identify dangerous driving behaviors when driving at high speed or when the vehicle is stationary.

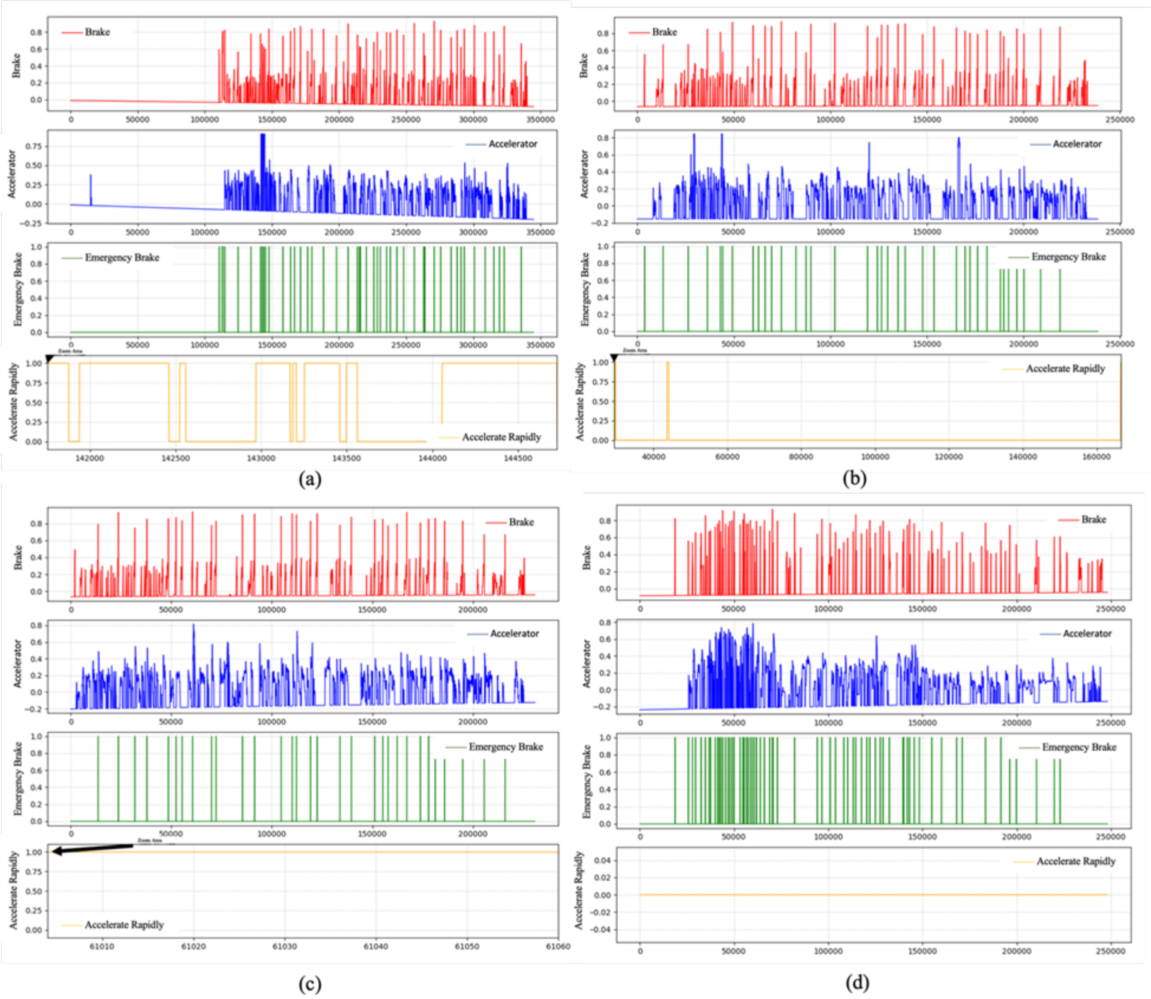


Figure 4.11: Driving Data Detection Effect Diagram

4.4 Dangerous Driving Behavior Detection System

This section introduces in detail the construction of a comprehensive detection platform for dangerous driving behaviors of drivers through Python and PyQt platforms. This experiment was designed and implemented by a front-end UI platform to simultaneously call YOLO target detection, Mask R-CNN instance segmentation, and detection algorithms based on driving data to detect multiple dangerous driving behaviors in real time and intuitively displays the output results of the comprehensive detection of dangerous driving behaviors of drivers. The proposed multi-data-driven method for identifying dangerous driving behaviors of drivers has been integrated into the front-end

UI interface. As shown in Figure 4.12, the upper part of the interface contains two QLabel components, which are used to display real-time video frame data of the driver's face, upper body movements, and foot movements on the brake and accelerator pedals.

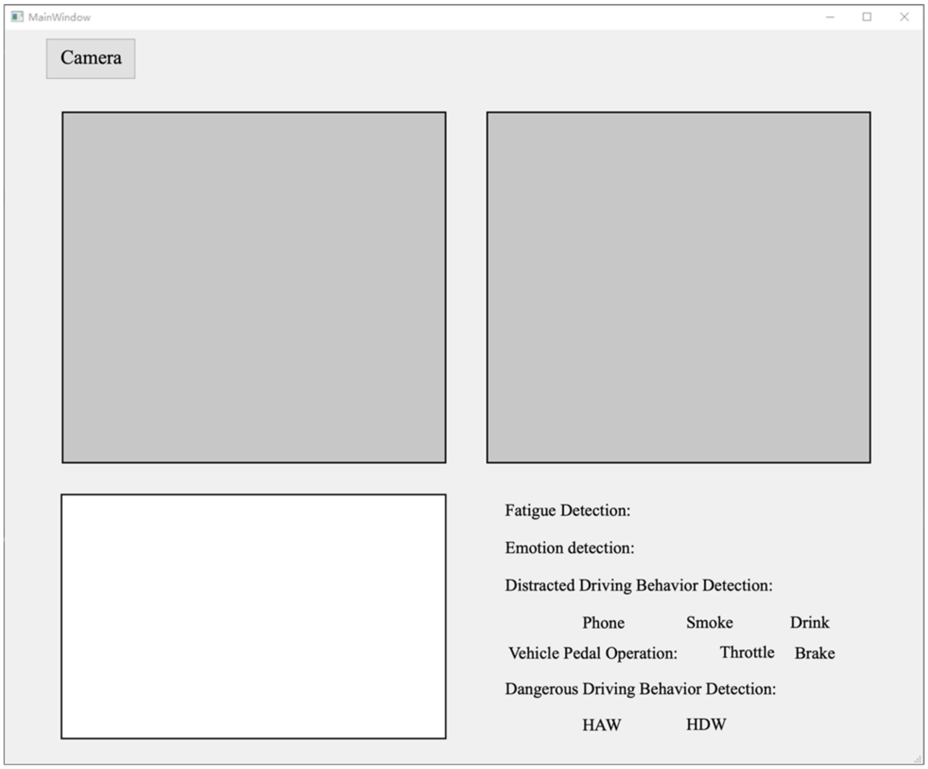


Figure 4.12: Front-end UI Interface Design Chart

In the implementation stage of the code, this experiment firstly called the OpenCV library to connect the transmission channel of the camera to read the video frame data intercepted from the camera video data. Specifically, the VideoCapture class of OpenCV was used to instantiate the camera object, which established communication with the camera output port and continuously obtains the video frame data in the loop structure. The image format of the video frame data obtained by OpenCV was BGR (blue-green-red), which was different from the RGB (red-green-blue) image format required by PqQt. Therefore, this experiment called the cv2.cvtColor() function to convert the color and spatial information of the image to ensure the normal processing and real-time display of the output results using the Qlabel component. In addition, a QTimer object was configured at the output end of the function cv2.cvtColor() to trigger screen updates regularly, realizing end-to-end real-time continuous video frame signal transmission and playback. In the real car experiment, just click the “Open Camera” button in the upper left corner of the interface to activate the system, start two camera streams, and realize real-time visual monitoring for testing and evaluation.

A QLabel interface was added to the left side of the lower half of the window page to display the real-time analysis process of the driver’s PERCLOS (percentage of closed eyes time) fatigue algorithm score during driving. During the real-time detection process, the detection system could display the real-time PERCLOS score output results through this interface. At the same time, on the right was the driver’s overall dangerous driving behavior detection interface, which contained multiple display labels, including fatigue detection results, distracted driving behavior detection results, vehicle pedal operation detection results, and dangerous driving behavior detection results. Similarly, during the real-time detection process, the detection system could display the real-time dangerous driving behavior output results through this interface.

If the system detects fatigue driving behavior, it will display a warning message in red font on the page: “ Fatigue! ! ! ”. Similarly, the results of the driver’s mood during driving will be displayed in the text box in the same way in real time. The results of distracted driving behavior, pedal handling and dangerous driving behavior are listed in the screen of distracted driving behavior, pedal handling and dangerous driving behavior. If the system detects one of these driving behaviors, the corresponding text font could turn red to remind the driver to regulate driving behavior and ensure driving safety.

In summary, in the front-end UI interface of distracted driving behavior detection results, vehicle pedal operation detection results, and dangerous driving behavior detection results, the system could list the relevant driving behaviors one by one. If the system detects one of these behaviors, the relevant dangerous driving behavior label could continue to turn red, reminding the driver to regulate driving behavior and ensure driving safety. The integration of the front-end UI interface enables real-time visualization within the driver risk behavior detection platform, thereby improving the end user’s interpretability of the detection process and results. Users can intuitively monitor driving behavior through the interface and receive early warnings and feedback in a timely manner, thereby improving driving safety and preventing potentially dangerous behaviors.

4.5 Real Vehicle Tests

4.5.1 Experimental Platform Introduction

In this study, the whole vehicle experimental platform was used in the real vehicle test. This experimental platform was based on the modification and debugging of the new energy pure electric vehicle with model code SCH5032XXY-BEV4 of Successful Motors and further modified and debugged by Tianjin Good Control Intelligent Technology Co., Ltd. to finally form an experimental platform with wire-control intelligent functions, as shown in Figure 4.13. The experimental platform was built with a real car chassis and electronic control and electrical systems, and could restore the driving conditions in the real driving environment and even the real road environment to the greatest extent.

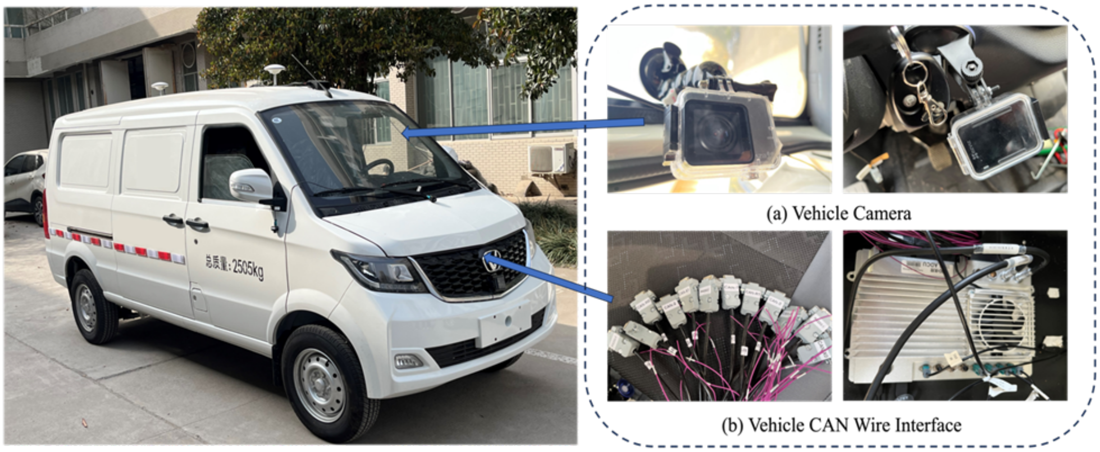


Figure 4.13: Vehicle Experiment Platform

On this experimental platform, this experiment processed the video frame signals of the driver's face, upper body movements, brake pedal and accelerator pedal in the vehicle, and processes the real-time CAN line data (including brake signal and throttle signal) transmitted by the vehicle's OBD interface to carry out the real-vehicle experiments. As shown in Figure 4.13(a), this experiment uses two sets of motion cameras (Fluorite S3 motion cameras: 6 million pixels, 2K resolution) mounted on the driver's side of the vehicle A-pillar and the top of the vehicle pedal, respectively, so as to be able to collect real-time video frames of the driver's risky driving behavior. On the other hand, as illustrated in Figure 4.13(b), the vehicle's OBD interface is employed to directly collect, analyze, and evaluate real-time CAN bus data for detection purposes.

4.5.2 Real Vehicle Tests

The real-vehicle experiment is an important step to verify the validity and reliability of the platform for detecting drivers' dangerous driving behaviors in real road environments, and Figure 4.14 shows the path planning diagram of this experiment. By conducting the experiment on campus and taking security measures at specific times, the road conditions and the safety of other traffic participants were ensured, which guaranteed the credibility of the experimental results.

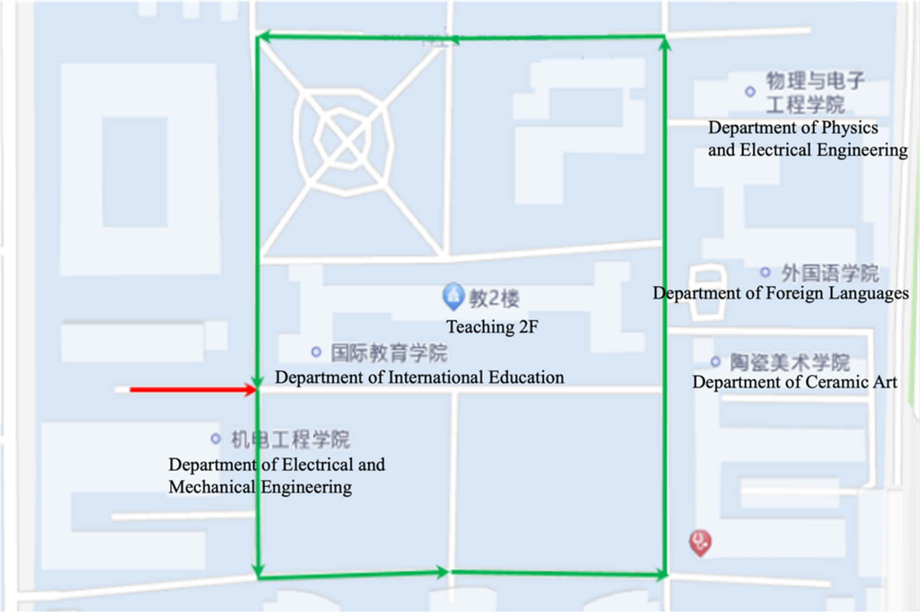

物理与电子
工程学院
Department of Physics
and Electrical Engineering
外国语学院
Department of Foreign Languages
教2楼
Teaching 2F
国际教育学院
Department of International Education
陶瓷美术学院
Department of Ceramic Art
机电工程学院
Department of Electrical and
Mechanical Engineering

Figure 4.14: Route Planner

In the real-vehicle experiment, two subjects participated in the related experiments of the Driver Risky Driving Behavior Detection Platform, and verified fatigue driving, distracted driving behavior, driving expression, and rapid acceleration and deceleration risky driving behaviors one by one. Examples of the results of this experiment are as follows:

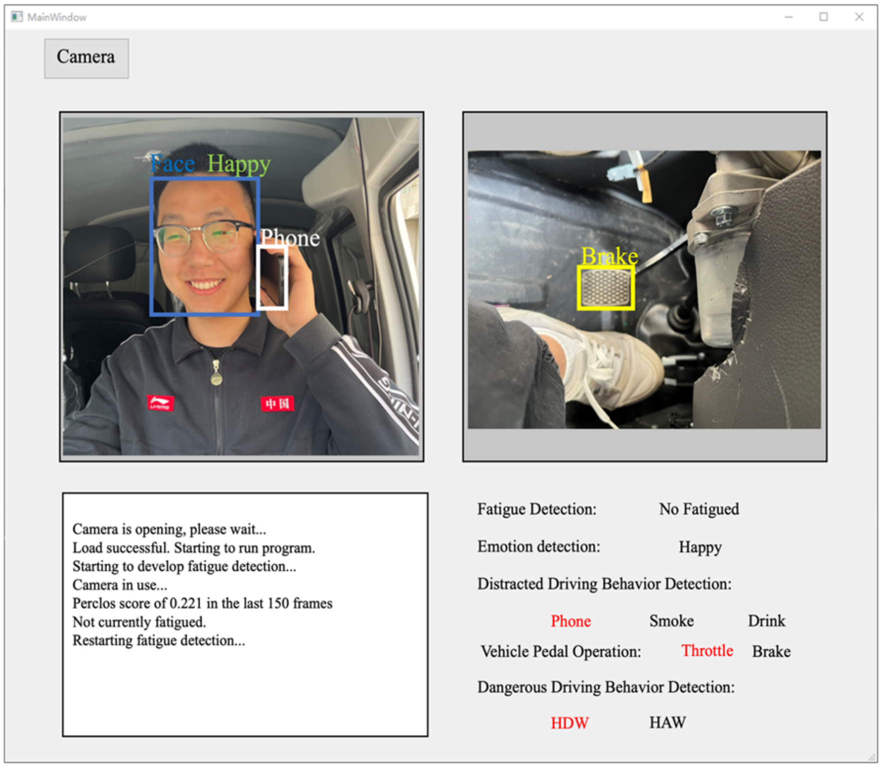
Experiment example 1 (Figure 4.15) shows that driver No. 1 is not fatigued during driving, but he is using a cell phone during driving, his driving mood is happy, and the vehicle is in a state of rapid acceleration.

Experiment example 2 (Figure 4.16) shows that driver No. 1 was not fatigued during the driving process, did not show distracted driving behavior, and his driving mood was frightened, and the vehicle was in a state of rapid deceleration.

Experiment example 3 (Figure 4.17) shows that driver No. 2 was fatigued during driving, did not show distracted driving behavior, his driving mood was normal, and the vehicle was in a normal state.

Example 4 (Figure 4.18) shows that Driver 2 was not fatigued, but smoked during driving, his driving mood was happy, and the vehicle was in a state of rapid acceleration.

In general, the above experimental results illustrated the effectiveness of the comprehensive driver dangerous driving behavior detection system driven by multi-source datasets proposed in this study. In the actual vehicle experiment, the detection system could detect, analyze and output the evaluation results of the driver's dangerous driving behavior in real time. However, the shortcomings of the detection system were also found during the actual vehicle experiment. The next step could be to continuously improve the underlying algorithm of the platform so that the detection system could adapt to more complex and more changeable actual driving environments, providing a basis for promoting road traffic safety.



```
Camera is opening, please wait...
Load successful. Starting to run program.
Starting to develop fatigue detection...
Camera in use...
Perclos score of 0.221 in the last 150 frames
Not currently fatigued.
Restarting fatigue detection...
```

Fatigue Detection: No Fatigued

Emotion detection: Happy

Distracted Driving Behavior Detection:

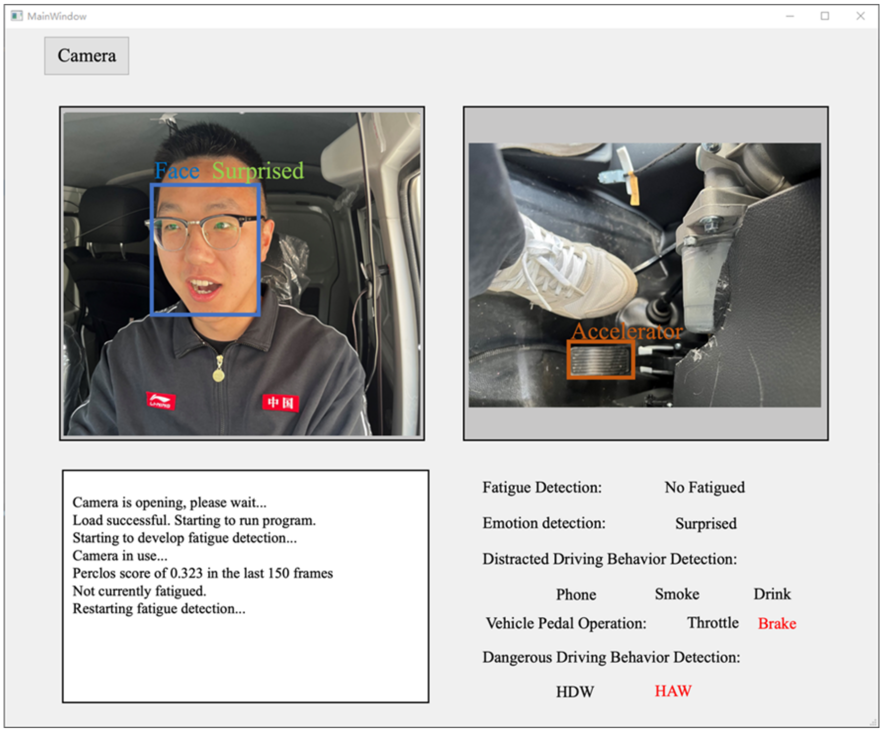
Phone Smoke Drink

Vehicle Pedal Operation: Throttle Brake

Dangerous Driving Behavior Detection:

HDW HAW

Figure 4.15: Diagram of Driver One's First Real-world Experiment



```
Camera is opening, please wait...
Load successful. Starting to run program.
Starting to develop fatigue detection...
Camera in use...
Perclos score of 0.323 in the last 150 frames
Not currently fatigued.
Restarting fatigue detection...
```

Fatigue Detection: No Fatigued

Emotion detection: Surprised

Distracted Driving Behavior Detection:

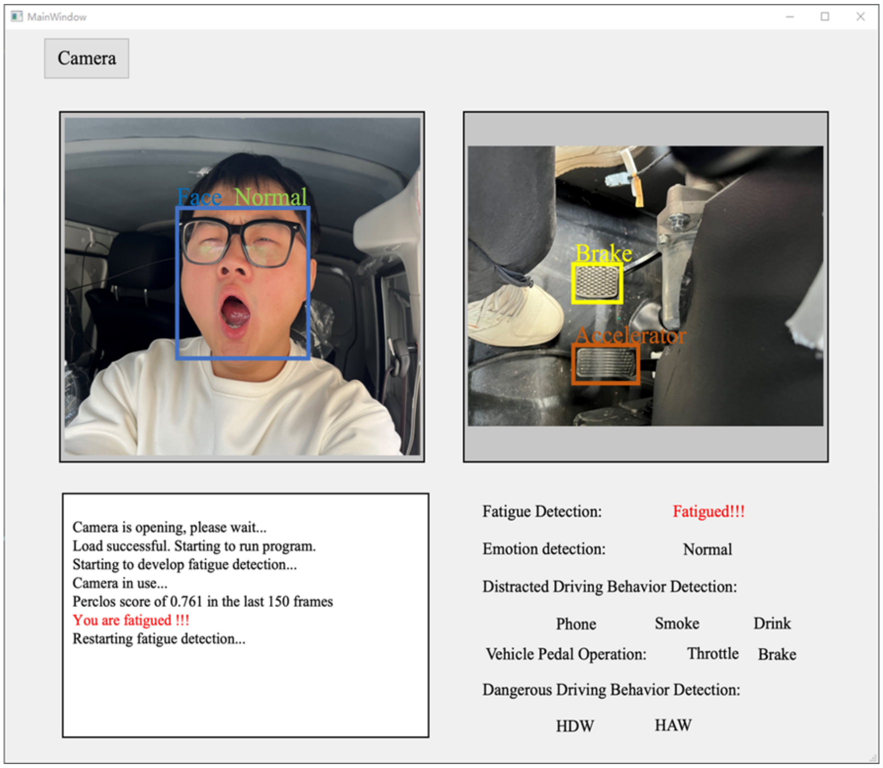
Phone Smoke Drink

Vehicle Pedal Operation: Throttle Brake

Dangerous Driving Behavior Detection:

HDW HAW

Figure 4.16: Diagram of Driver One's Second Real-world Experiment



```
Camera is opening, please wait...
Load successful. Starting to run program.
Starting to develop fatigue detection...
Camera in use...
Perclos score of 0.761 in the last 150 frames
You are fatigued !!!
Restarting fatigue detection...
```

Fatigue Detection: Fatigued!!!

Emotion detection: Normal

Distracted Driving Behavior Detection:

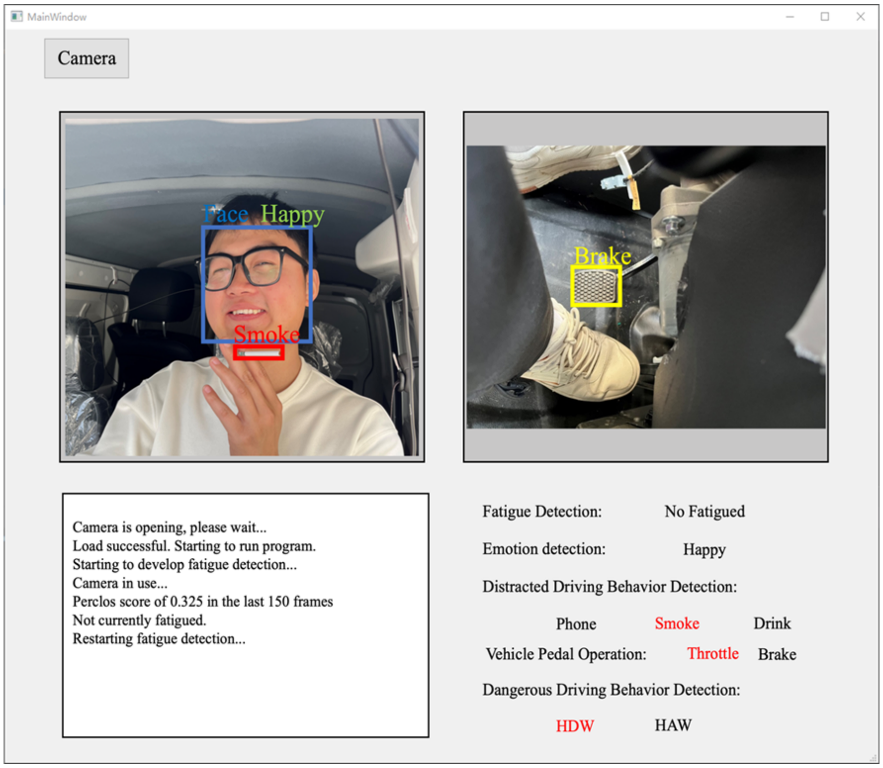
Phone Smoke Drink

Vehicle Pedal Operation: Throttle Brake

Dangerous Driving Behavior Detection:

HDW HAW

Figure 4.17: Diagram of Driver Two’s First Real-world Experiment



```
Camera is opening, please wait...
Load successful. Starting to run program.
Starting to develop fatigue detection...
Camera in use...
Perclos score of 0.325 in the last 150 frames
Not currently fatigued.
Restarting fatigue detection...
```

Fatigue Detection: No Fatigued

Emotion detection: Happy

Distracted Driving Behavior Detection:

Phone Smoke Drink

Vehicle Pedal Operation: Throttle Brake

Dangerous Driving Behavior Detection:

HDW HAW

Figure 4.18: Diagram of Driver Two’s Second Real-world Experiment

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis addresses the problem of detecting dangerous driving behaviors of drivers through the improved YOLOv8 target detection neural network algorithm, the example segmentation Mask-RCNN neural network algorithm, and the sliding window method based on driving data. The problem of fatigue driving, distracted driving, driving mood, and operation of brake pedal and gas pedal were detected in real time during the driving process. The neural network algorithms were integrated with a Python GUI page to build a multidata-driven platform for detecting dangerous driving behaviors. The main conclusions are as follows:

By improving the YOLOv8 network layer, the MHSA attention mechanism module and the driver emotion detection module were introduced into the neural network, which realized the detection of the driver's driving emotion during the driving process as well as the reduction of the model computation parameters and the improvement of the model detection speed and accuracy. The proposed improved YOLOv8 detection algorithm based on the MHSA attention mechanism module and the driver emotion detection module obtained an accuracy rate of 81.4%.

In order to realize the processing and evaluation of the video frame signal of the driver's operation of the brake pedal and accelerator pedal during the driving process, the instance segmentation Mask-RCNN detection algorithm was used, and the model achieved a better performance with an accuracy of 84.6% under the condition of poor detection environment.

In order to process and evaluate the brake opening signal and throttle opening signal in the CAN bus of the vehicle in real-time driving data, data filtering and window sliding algorithms were used to filter the signal data and compare the processed data with the set threshold, and the proposed algorithm makes real-time judgments of the dangerous driving behaviors of rapid acceleration and rapid deceleration and achieved the expected performance.

By integrating the neural network algorithm for video frame signal detection and the driving data detection algorithm proposed in this study, the front-end UI interface was created by using the GUI page in Python, thus realizing the real-time visualization page of the platform for detecting dangerous driving behaviors of drivers based on multidimensional data driving.

5.2 Future Work

The thesis investigated the dangerous driving behaviors related to drivers in existing studies, including fatigue driving behavior detection based on video frame signals, distracted driving behavior detection, driver emotion detection, and driver's operation of the brake pedal and accelerator pedal detection; rapid acceleration and deceleration detection was based on driving data, which used neural network modeling algorithms and data processing algorithms, and ultimately integrated them to realize the design and implementation of a multidata-driven detection platform. Neural network modeling algorithms and data processing algorithms were used for detection, and these algorithms were finally integrated to realize the design and implementation of a multidimensional data driven detection platform. This study achieved the expected experimental results and research theories based on the actual vehicle test, and the following aspects could be studied in the future.

Further expand the sample size. The distracted driving behavior dataset used in this thesis was a self-made dataset, and due to the protection of personal information and privacy, it was difficult to collect videos, which brings great challenges to the research. This led to the limitation of data collection and production. Meanwhile, the driving data and driver pedal operation video frame signal data used in this thesis came from the actual driving of Yutong Bus Co., Ltd. and due to the limited resources, it was not possible to collect a large amount of driving data with multiple driving conditions and scenarios. Accordingly, subsequent research will aim to expand the sample size and continue exploring driver detection through video frame signal processing to enhance the robustness and generalizability of the proposed approach.

In this study, two sets of cameras were used to detect dangerous driving behaviors, including the detection of driver fatigue, distracted driving behavior, driver emotion, and the detection of driver's operation of the brake pedal and accelerator pedal. This real-car experiment was used by two motion cameras at different settings to detect the driver's dangerous driving behavior, including the driver's fatigue driving, distracted driving behavior, driver's emotions, and the driver's operation of the brake pedal and accelerator pedal. This experiment only considered the detection of dangerous driving behaviors of drivers under good lighting conditions, without considering the impact of complex and changing driving conditions and environments on the detection method. In the future, using infrared cameras to analyze and detect video frame data will be considered in low-light environments, so as to accurately analyze and evaluate dangerous driving behaviors of drivers.

Owing to the limitations of the experimental driving path and the number of test drivers, this study only collected the real-time detection effect of two test drivers, and the driving path was only in the safe road in the campus, without carrying out in-depth research on various driver driving styles and various driving road conditions. Future research will include more experienced drivers and more diverse driving scenarios to facilitate the feasibility of datasets and complex environmental conditions for dangerous driving behavior detection.

Reference

Abe, T., Mollicone, D., Basner, M., and Dinges, D. F. (2014). Sleepiness and safety: Where biology needs technology. *Sleep and Biological Rhythms*, 12(2):74–84.

Ahmad, T., Ma, Y., Yahya, M., and Rehman, S. (2020). Object detection through modified yolo neural network. *Scientific Programming*, 2020:8403262.

Al-Sultan, S. J. (2013). *Context Aware Drivers' Behaviour Detection System for VANET*. Ph.d. dissertation, De Montfort University. Accessed: 2025-04-20.

Ali, P. J. M. (2022). Investigating the impact of min-max data normalization on the regression performance of k-nearest neighbor with different similarity measurements. *ARO – The Scientific Journal of Koya University*, 10(1):85–91.

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. Accessed: 2025-04-20.

Buccafusco, S. et al. (2021). Profiling industrial vehicle duties using can bus signal segmentation and clustering. In *EDBT/ICDT Workshops*.

Cai, A. W. T., Manousakis, J. E., Lo, T. Y. T., and Baran, T. M. (2021). I think i'm sleepy, therefore i am – awareness of sleepiness while driving: A systematic review. *Sleep Medicine Reviews*, 60:101533.

Cao, J. Y. (2022). *Fatigue driving monitoring technology based on GELM EOG signals and visual information fusion*. PhD thesis, Northeast Electric Power University, Jilin, China.

Carvalho, O. L. F. et al. (2020). Instance segmentation for large, multi-channel remote sensing imagery using mask-rcnn and a mosaicking approach. *Remote Sensing*, 13(1):39.

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6.

Cori, J. M., Anderson, C., Soleimanloo, S. S., and Galang, J. (2019). Narrative review: Do spontaneous eyeblink parameters provide a useful assessment of state drowsiness? *Sleep Medicine Reviews*, 45:95–104.

CSDN (2016). Deep learning [eb/ol]. Accessed: 2025-04-20.

Deligianni, S. P. et al. (2017). Analyzing and modeling drivers' deceleration behavior from normal driving. *Transportation Research Record*, 2663(1):134–141.

Dinges, D. F., Mallis, M. M., Maislin, G., and Powell, J. W. (1998). Evaluation of techniques for ocular measurement as an index of fatigue and as the basis for alertness management. Technical report DOT HS 808 762, U.S. Department of Transportation, National Highway Traffic Safety Administration.

Feng, S. P., Han, R. F., Zhao, Y. J., et al. (2024). Fatigue driving early warning system based on computer image processing. *Science and Innovation*, (02):77–80.

Ge, Z., Liu, S., Wang, F., and Li, J. (2021). Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430. Accessed: 2025-04-20.

Goodfellow, I. J. et al. (2013). Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Proceedings, Part III*, volume 8228 of *Lecture Notes in Computer Science*, pages 117–124. Springer Berlin Heidelberg.

Guo, Y. Z. (2011). *Research on fatigue driving detection technology based on EEG*. PhD thesis, Northeastern University, Shenyang, China.

Hasrouny, H., Samhat, A. E., Bassil, C., and Laouiti, A. (2017). Vanet security challenges and solutions: A survey. *Vehicular Communications*, 7:7–20.

Hayashi, Y., Foreman, A. M., Friedel, J. E., and Wirth, O. (2018). Executive function and dangerous driving behaviors in young drivers. *Transportation Research Part F: Traffic Psychology and Behaviour*, 52:51–61.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969.

Hnewa, M. and Radha, H. (2023). Integrated multiscale domain adaptive yolo. *IEEE Transactions on Image Processing*, 32:1857–1867.

Huang, X., Wang, X., Lv, W., et al. (2021). Pp-yolov2: A practical object detector. arXiv preprint arXiv:2104.10419. Accessed: 2025-04-20.

Huang, Y. K. (2020). *Driver fatigue state detection based on EEG and EOG signals*. PhD thesis, Harbin University of Science and Technology, Harbin, China.

Jia, L. J. (2019). *Research on fatigue driving detection methods based on steering wheel operation characteristics in real vehicles*. PhD thesis, Tsinghua University, Beijing, China.

Jiang, P., Ergu, D., Liu, F., and Cai, Y. (2022). A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073.

Jiao, Y. Y. (2019). *Driver drowsiness detection based on EEG and EOG signals*. PhD thesis, Shanghai Jiao Tong University, Shanghai, China.

Jin, L. (2017). *Fatigue driving research based on ECG signals*. PhD thesis, Chongqing University, Chongqing, China.

Jocher, G., Chaurasia, A., Stoken, A., et al. (2022). ultralytics/yolov5: v7.0 - yolov5 sota real-time instance segmentation. Zenodo.

Kampen, J. V., Knapen, L., de Mei, R. V., Pauwels, E., and Dugundji, E. R. (2022). Yearly development of car ownership in urban and rural environments. *Procedia Computer Science*, 201:101–108.

Kaplan, S., Guvensan, M. A., Yavuz, A. G., and Karalurt, Y. (2015). Driver behavior analysis for safe driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3017–3032.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*.

Ko, B. C. (2018). A brief review of facial emotion recognition based on visual information. *Sensors*, 18(2):401.

Lal, S. K. L. and Craig, A. (2000). Physiological indicators of driver fatigue. *Environments*, 19:20.

Lal, S. K. L. and Craig, A. (2001). A critical review of the psychophysiology of driver fatigue. *Biological Psychology*, 55(3):173–194.

Lan, Z. D. (2021). *Fatigue detection research based on EEG and vehicle motion information fusion*. PhD thesis, Dalian University of Technology, Dalian, China.

Li, C., Li, L., Jiang, H., et al. (2022). Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976. Accessed: 2025-04-20.

Li, D. H., Liu, Q., Yuan, W., and Liu, H. X. (2010). The relationship between fatigue driving and traffic accidents. *Journal of Transportation Engineering*, 10(2):104–109.

Liang, Y. et al. (2025). Study on the improvement of semi-hertzian wheel/rail contact algorithms. *Journal of Traffic and Transportation Engineering (English Edition)*.

Lu, K. X. (2016). Analysis of driving behavior based on onboard information fusion. Master's thesis, Harbin Institute of Technology, Harbin, China.

Mao, A., Mohri, M., and Zhong, Y. (2023). Cross-entropy loss functions: Theoretical analysis and applications. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR.

Martinez, C. M., Heucke, M., Wang, F. Y., Gao, B., and Cao, D. (2018). Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):666–676.

McManus, B., Heaton, K., Vance, D. E., and Stavrinos, D. (2016). The useful field of view assessment predicts simulated commercial motor vehicle driving safety. *Traffic Injury Prevention*, 17(7):763–769.

Miyajima, C. et al. (2011). Driver risk evaluation based on acceleration, deceleration, and steering behavior. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2120–2123. IEEE.

National Bureau of Statistics of China (2021). *China Statistical Data 2021*.

Petridou, E. and Moustaki, M. (2000). Human factors in the causation of road traffic crashes. *European Journal of Epidemiology*, 16(9):819–826.

Province, S. (2021). Implementation measures of the road traffic safety law of the people's republic of china. *Sichuan Daily*, page 007.

Purohit, S. and Govindarasu, M. (2022). Ml-based anomaly detection for intra-vehicular can-bus networks. In *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE.

Qin, L., Shi, Y., He, Y., Zhang, J., Zhang, X., Li, Y., Deng, T., and Yan, H. (2022). Id-yolo: Real-time salient object detection based on the driver's fixation region. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15898–15908.

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

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. Accessed: 2025-04-20.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1–3):157–173.

Sathyanarayana, A., Boyraz, P., and Hansen, J. H. L. (2008). Driver behavior analysis and route recognition by hidden markov models. In *2008 IEEE International Conference on Vehicular Electronics and Safety*, pages 276–281, Columbus, OH, USA.

Shi, X. Y. (2019). *Fatigue driving diagnosis based on ECG signals*. PhD thesis, North China University of Technology, Beijing, China.

Sikander, G. and Anwar, S. (2019). Driver fatigue detection systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2339–2352.

Song, W. (2023). Research on visual recognition method of illegal driving behavior based on deep learning. Master's thesis, Wuhan University of Science and Technology, Wuhan, China.

Sparrow, A. R., LaJambe, C. M., and Van Dongen, H. P. A. (2019). Drowsiness measures for commercial motor vehicle operations. *Accident Analysis & Prevention*, 126:146–159.

Uçar, S. and Oguchi, K. (2021). Distracted driving behavior detection to avoid rear-end collisions. In *2021 IEEE Vehicular Networking Conference (VNC)*, pages 115–116.

Wang, C. Y., Bochkovskiy, A., and Liao, H. Y. M. (2023a). 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 Workshops*, pages 7464–7475.

Wang, G., Chen, Y., An, P., Tian, B., Li, B., and Wang, Y. (2023b). Uav-yolov8: A small-object-detection model based on improved yolov8 for uav aerial photography scenarios. *Sensors*, 23(16):7190.

Wang, J. (2023). Research on fatigue driving detection methods based on eeg signals. Master's thesis, Zhejiang Normal University, Hangzhou, China.

Wang, J., Wu, Z. C., Li, F., and Zhang, J. (2021). A data augmentation approach to distracted driving detection. *Future Internet*, 13(1):1.

Wang, Z. R. (2024). Research on driver behavior detection methods based on video. Master's thesis, North China Electric Power University, Beijing, China.

Wei, T. Z., Lin, M., Li, C. X., et al. (2021). Research on perception characteristics and discrimination model of implicit dangerous drivers. *China Safety Science Journal*, 17(3):175–181.

Wierwille, W. W. and Ellsworth, L. A. (1994). Evaluation of driver drowsiness by trained raters. *Accident Analysis & Prevention*, 26(5):571–581.

Wierwille, W. W., Wreggit, S. S., Kirn, C. L., et al. (1994). Research on vehicle-based driver status/performance monitoring: Development, validation, and refinement of algorithms for detection of driver drowsiness. Technical report DOT HS 808 433, U.S. Department of Transportation, National Highway Traffic Safety Administration.

World Health Organization (2023). *Global status report on road safety 2023*. World Health Organization.

Xiang, H. et al. (2021). Prediction of dangerous driving behavior based on vehicle motion state and passenger feeling using cloud model and elman neural network. *Frontiers in Neurorobotics*, 15:641007.

Yang, C. (2023). Warning! there are dangers you can't see. *Jinan Daily*, page 005.

Yang, F. F. and Li, J. (2023). A review of yolo object detection algorithms for autonomous driving. *Automotive Engineers*, (11):1–11.

Ye, C. W. (2018). *Research on automotive driving fatigue based on ECG and EMG signals*. PhD thesis, Hefei University of Technology, Hefei, China.

Yin, Y. H. (2008). *Simulation experiment research on driver fatigue based on EEG and blink*. PhD thesis, Tongji University, Shanghai, China.

Zhang, C., Cheng, J., Li, L., Sun, F., and Zhao, C. (2017). Object categorization using class-specific representations. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4528–4534.

Zhang, P. et al. (2025). Research on transformer temperature early warning method based on adaptive sliding window and stacking. *Electronics*, 14(2).

Zhang, T. C. (2024a). Research on driver fatigue detection based on facial and human behavior characteristics. Master's thesis, Lanzhou Jiaotong University, Lanzhou, China.

Zhang, Y. J. (2009). Image segmentation in the last 40 years. In Khosrow-Pour, M., editor, *Encyclopedia of Information Science and Technology, Second Edition*, pages 1818–1823. IGI Global.

Zhang, Y. N. (2021). Research on dangerous driving behavior based on driving data. Master's thesis, Liaoning University of Technology, Jinzhou, China.

Zhang, Y. N., Tang, Y. S., Liu, H., et al. (2019). Determination of dangerous driving behaviors based on driving data. *Automotive Practical Technology*, (15):247–248.

Zhang, Z. (2018). Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2. IEEE.

Zhang, Z. H. (2024b). Research on driving behavior based on obd data analysis. Master's thesis, Chang'an University, Xi'an, China.

Zhang, Z. Y. and Ye, G. X. (2022). Review of driver fatigue monitoring technology. *Automotive Technology*, (01):8–14.

Zhao, L. (2021). Design and implementation of a fatigue detection system based on facial features. Master's thesis, Jinan University, Jinan, China.

Zheng, W., Zhang, Q. Q., Ni, Z. H., and Liu, Y. (2020). Distracted driving behavior detection and identification based on improved cornernet-saccade. In *2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1150–1155.

Zhihu (2023). Automotive design [eb/ol]. Accessed: 2025-04-20.

Zhou, W., Zhu, Y., Lei, J., Wang, Y., and Yang, F. (2023). Lsnet: Lightweight spatial boosting network for detecting salient objects in rgb-thermal images. *IEEE Transactions on Image Processing*, 32:1329–1340.

Zong, S. J., Dong, F., Cheng, Y. X., Yu, D. H., Yuan, K., Wang, J., Ma, Y. X., and Zhang, F. (2024). Applications and challenges of eeg signals in fatigue driving detection. *Progress in Biochemistry and Biophysics*, 51(7):1645–1669.

Zou, Z., Chen, K., Shi, Z., et al. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.