CRANFIELD UNIVERSITY

YANG XING

DRIVER LANE CHANGE INTENTION INFERNECE USING MACHINE LEARNING METHODS

SCHOOL OF AEROSPACE, TRANSPORT, AND MANUFACTURING
Transport Systems

PhD
Academic Year: 2015 - 2018

Supervisors: Dr. Dongpu Cao & Dr. Efstathios Velenis
April 2018

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT, AND MANUFACTURING
Transport Systems

PhD

Academic Year 2015 - 2018

Yang Xing

DRIVER LANE CHANGE INTENTION INFERENCE USING MACHINE LEARNING METHODS

Supervisors: Dr. Dongpu Cao & Dr. Efstathios Velenis
April 2018

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy
(NB. This section can be removed if the award of the degree is based solely on examination of the thesis)

“Intelligence is the ability to adapt to change.”

Stephen Hawking

“If a machine is expected to be infallible, it cannot also be intelligent.”

Alan Turing

ABSTRACT

Lane changing manoeuvre on highway is a highly interactive task for human drivers. The intelligent vehicles and the advanced driver assistance systems (ADAS) need to have proper awareness of the traffic context as well as the driver. The ADAS also need to understand the driver potential intent correctly since it shares the control authority with the human driver. This study provides a research on the driver intention inference, particular focus on the lane change manoeuvre on highways.

This report is organised in a paper basis, where each chapter corresponding to a publication, which is submitted or to be submitted. Part I introduce the motivation and general methodology framework for this thesis. Part II includes the literature survey and the state-of-art of driver intention inference. Part III contains the techniques for traffic context perception that focus on the lane detection. A literature review on lane detection techniques and its integration with parallel driving framework is proposed. Next, a novel integrated lane detection system is designed. Part IV contains two parts, which provides the driver behaviour monitoring system for normal driving and secondary tasks detection. The first part is based on the conventional feature selection methods while the second part introduces an end-to-end deep learning framework. The design and analysis of driver lane change intention inference system for the lane change manoeuvre is proposed in Part V. Finally, discussions and conclusions are made in Part VI.

A major contribution of this project is to propose novel algorithms which accurately model the driver intention inference process. Lane change intention will be recognised based on machine learning (ML) methods due to its good reasoning and generalizing characteristics. Sensors in the vehicle are used to capture context traffic information, vehicle dynamics, and driver behaviours information. Machine learning and image processing are the techniques to recognise human driver behaviour.

Keywords:

Driver intention inference, Machine learning, ADAS, Driver behaviours, Machine vision.

ACKNOWLEDGEMENTS

First of all, I would like to mention my supervisor Dr. Dongpu Cao, who has been always attentive and has encouraged me to always explore even further, broaden my frontiers and support me all the time. He leads me to the research road, which is full of fun and challenge. I would also like to thank my second supervisor Dr. Efstathios Velenis for sharing the valuable advice and the great support during this project.

Furthermore, I would like to thank Dr. Chen Lv who has been always encourage me, inspire me, and advise me. I would like to express my gratitude to Professor Saber Fallah and Dr. Stefano longo for their kind suggestions that review of this report. I would like to thank as well Dr. Huaji Wang, who give me great support for the experiment design in this project. I am also grateful for Dr. Yifan Zhao who offer the valuable advice and the chance to take the course of image processing, which is one of the most important weapons for my project. Special thanks are required by the colleagues and friends from Cranfield and all the people who helped me, especially thanks to Zhaozhong Zhang, and Chunwei Chang who share many hours with me in setting up the experiments and collecting data.

Finally, I would like to mention my parents for their unquestioning love and understanding, I could not finish the project without their great support. Last but not least, I so grateful for the love and support of my wife. Without her love and encouragement, none of this would be possible.

DECLARATION OF AUTHORSHIP

This thesis is submitted to Cranfield University in support of my application for the degree of Doctor of Philosophy. It contains my own original works during the three years study at Cranfield University and has not been submitted in any previous application for a degree. My contributions to the research community with regard to the work I completed can be found in the following list.

Published journal papers as 1st Author

Yang Xing, Chen Lv, Dongpu Cao, Huaji Wang, and Yifan Zhao. "Driver workload estimation using a novel hybrid method of error reduction ratio causality and support vector machine." Measurement 114 (2018): 390-397.

Xing, Yang, *et al*. "Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition." IEEE Transactions on Computational Social Systems 5.1 (2018): 95-108.

Y. Xing, C. Lv, L. Chen, H. J. Wang, H. Wang, D. P. Cao, E. Velenis, F. Y. Wang, "Advances in vision-based lane detection: algorithms, integration, assessment, and perspectives on ACP-based parallel vision," IEEE/CAA J. of Autom. Sinica, vol. 5, no. 3, pp. 645-661, May. 2018.

Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis. "Dynamic Integration and Online Evaluation of Vision-Based Lane Detection Algorithms". IET Intelligent Transport Systems. 2018.

Journal papers under review/to submit as 1st Author

Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang. Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach. IEEE Trans. on Vehicular Technology. (Under Review).

Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang. Driver Lane Change Intention Inference for Intelligent Vehicles: Framework, Survey, and Challenges. (To submit)

Yang Xing, Chen Lv, Huaji Wang, Hong Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang. Driver Lane Change Intention Inference: Framework, Algorithms, Testing Results, and Analysis. (To submit)

Conference papers

Yang, Xing, Chen Lv, Wang Huaji, Hong Wang, and Dongpu Cao. Recognizing Driver Braking Intention with Vehicle Data Using Unsupervised Learning Methods. No. 2017-01-0433. SAE Technical Paper, 2017.
Yang, Xing, Chen Lv, Dongpu Cao, Efstathios Velenis, and Fei-Yue Wang. End-to-End Driving Activities and Secondary Tasks Recognition Using Deep Convolutional Neural Network and Transfer Learning. 2018 Intelligent Vehicle Symposium.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACC	Adaptive Cruise Control
ACP	Artificial Society, Computational Experiments, and Parallel Execution
ACT-R	Adaptive Control of Thought-Rational
ADAS	Advanced Driver Assistance System
ANN	Artificial Neural Network
AVM	Around View Monitoring
BCI	Brain Computer Interface
BF	Bayesian Filter
BN	Bayesian Network
CDI	Comprehensive Decision Index
CHMM	Continuous Hidden Markov Model
CLNF	Conditional Local Neural Fields
CNN	Convolutional Neural Network
DBSCAN	Density based Spatial Clustering of Application with Noise
ED	Edge Distribution
EEG	Electroencephalograph
ECG	Electrocardiogram
EMG	Electromyography
EOG	Electrooculography
EPRC	Error Reduction Ratio Causality
FFNN	Feedforward Neural Network
FPR	False Positive Rate
GA	Genetic Algorithm
GMM	Gaussian Mixture Model
GNSS	Global Navigation Satellite System
GOLD	Generic Obstacle and Lane Detection
GPS	Global Positioning System
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HRI	Human Robot Interaction
HT	Hough Transform
IMM	Interactive Multiple Model
IMU	Inertial Measurement Unit
IOHMM	Input-Output Hidden Markov Model
IPM	Inverse Perspective Mapping
LADAR	Laser Detection and Ranging
LANA	Lane Finding in Another Domain

LCII	Lane Change Intention Inference
LDA	Lane Departure Avoidance
LDW	Lane Departure Warning
Lidar	Light Detection and Ranging
LKA	Lane Keeping Assistance
LOO	Leave-One-Out
LSV	Lane Sampling and Voting
LSTM	Long Short-Term Memory
MIC	Maximal Information Coefficient
NFIR	Nonlinear Finite Impulse Response
OLS	Orthogonal Least Squares
OOB	Out-of-Bag
PCA	Principal Component Analysis
RANSAC	Random Sample Consensus
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROI	Region of Interest
RVM	Relevance Vector Machine
SBL	Sparse Bayesian Learning
SF	Steerable Filter
SCR	Skin Conductance Response
SWA	Side Warning Assistance
SVM	Support Vector Machine
SVR	Support Vector Regression
TDV	Traffic-Driver-Vehicle
TS	Time Sliced
TPR	True Positive Rate
WHO	World Health Organization

PART I:

INTRODUCTION AND MOTIVATION

1

Introduction and Motivation

1.1 Introduction and Motivation

Worldwide traffic departments have reported that more than 1.2 million traffic-related injuries happen each year. Among these traffic accidents, more than 80% were caused by human errors [1]. The world Health Organization (WHO) reported that traffic accidents each year cost around 518 billion euros worldwide, and on average 1 to 2 % of the world Gross Domestic Product [2] [3]. In the past, in-vehicle passive safety systems like airbags and seat-belts have played a significant role in the protection of driver and passengers. These technologies have saved millions of lives. However, they are not designed to prevent the accidents from happening, but just trying to minimise the injuries after the accidents happen [4]. Therefore, recent efforts have been devoted on the development of safer and intelligent systems towards the prevention of accidents. These systems are known as the Advanced Driver Assistance Systems (ADAS).

ADAS are a series of fast developing techniques which are designed for improving the driver safety and increasing the driving experience [5]. ADAS products rely on the multi-modal sensor fusion technique to integrate multiple sensors such as the light detection and ranging (Lidar), radar, camera, and GPS into a holistic system. Most of the current ADAS techniques such as Lane Departure Avoidance (LDA), Lane Keeping Assistance (LKA) and Side Warning Assistance (SWA) can help the driver to make the right decision and reduce their workload. However, the inputs of these systems are vehicle dynamic states and traffic context information only. Most of the systems ignore the most important factor, the driver. Vehicles are working in a three-dimensional environment with continuous driver-vehicle-road interactions. Drivers are the most important part in this system, who control the vehicle based on the surrounding traffic context perception. Therefore, allowing ADAS to understand driver behaviours and follow their intention is of importance to the driver safety, vehicle drivability and traffic efficient.

Driver intention inference is an ideal way to allow ADAS to obtain the ability of reasoning. The reasons for developing driver intention inference technique are multi-folds: the most significant motivation is to improve the driver safety. There are two different driving scenarios which require inferring the driver intention: one is to better assess the risk in the future, the second one is to avoid making misleading decisions which are opposite to the driver intent. For the first case, there is an evidence that a large amount of accidents are caused by human errors or misbehaviours such as the cognitive (47%),

judgment (40%), and operational (13%) errors, respectively [6]. Therefore, understanding driver intention and behaviours is crucial to the effectiveness of future ADAS. Meanwhile, it is sure that the increasingly usage of the in-vehicle devices and entertainment systems would heavily distract the driver from the driving task. For the design of future ADAS, it is beneficial to integrate the driver intention system from the initial design stages [7]. In terms of the second scenario, ADAS systems automatically intervene vehicle dynamics and share the control authority with drivers. To ensure the effective cooperation of driver and the vehicle, ADAS should be aware of driver intention and not operate against the driver's willingness. [8].

Furthermore, the intention information will contribute to the development of automated vehicles. Driver intention inference will benefit the construction of driver model, which can act as the guidance to design automatic decision-making system. Moreover, in terms of the level three automated vehicle (according to the SAE international standard on the classification of automated vehicles), accurate driver intention prediction enables a smoother and safer transition between the driver and the autonomous controller [9] [10]. When the level three automated vehicles operating at autonomous condition, all the driving manoeuvres are handled by the vehicle. However, once the vehicle cannot deal with an emergent situation, it has to give the driving authority to the driver. This process is known as disengagement [11]. In such case, the vehicle can assess the takeover ability of the driver according to the continuously detected intention. If the driver is focusing on the driving task at that moment and has an explicit intention, the vehicle can warn the driver and pass the driving authority to him. This will make sure the transition between driver and controller as smooth as possible. However, if the driver is believed to be not able to handle the driving scenario, the autonomous driving units should help the driver gain the situation awareness as soon as possible or take emergency action immediately.

1.2 Research Aims and Objectives

As discussed above, teaching ADAS to understand driver intention is an important as well as challenging task to enhance the safety of the driver-vehicle-traffic close-loop system. To be more focus, this project will be targeting at one of the most popular driving scenarios, namely, the lane change manoeuvre. During a normal lane change manoeuvre, driver is expected to perform a series of behaviours (e.g. mirror checking and turn steering wheel). This project aims at inferring driver lane change intention as early as possible by

recognising driver behaviours and traffic context situation. A driver lane change intention system (LCII) which facing to next generation ADAS is developed in this project. Based on this, four main objectives are determined:

1. Driver intention process analysis. To predict driver lane change intention, it is vital to understand the human intention mechanism, such as how the intention is generated and what is the stimuli of the intention. The nature behind driver intention is the first question that need to be answered.
2. Traffic context perception. Driver is the in the middle of the traffic-driver-vehicle loop. Traffic context is the inputs to the driver perception system, which makes it act as the stimuli of the driver intention. Therefore, understanding the current traffic situation will benefit the intention inference system.
3. Driver behaviours understanding. Driver behaviours such as mirror checking is the most important clues before the driver making a lane change. Driver has to perform a series of checking action to have a fully understanding about the surrounding context before they decide to change the lane. Therefore, driver behaviour analysis is of important to infer driver intention.
4. Driver lane change intention inference algorithms. Based on the specific traffic context and driver behaviours, the next task is to properly infer driver intention. The algorithms for intention inference should have the ability to capture the long-term dependency between the temporal sequences. Moreover, the intention inference algorithms should predict the intention as early as possible.

The driver lane change intention platform requires the integration of software and hardware system. Driver intention inference has to take either the traffic context, driver behaviours, and vehicle dynamic information into consideration, which will fuse multi-modal signals and mining the long-term dependency between different signals based on machine learning methods. In terms of the hardware system, the sensors included in this project contains RGB and RGB-D cameras and vehicle navigation system. Besides, all the sensors will be mounted on a real vehicle in case to collect the naturalistic data. Specifically, the traffic context like lane positions and front vehicle position will be processed with image-processing methods. One web camera is mounted inside of the cabinet. The driver behaviour dynamics will be evaluated within a steady and dynamic

vehicle. The RGB-D camera (Microsoft Kinect V2.0) will be used for steady vehicle while another web camera will be used to record the driver behaviour during the highway driving task. These signals are recorded with one laptop for further processing and analysing. The algorithms used in this project are mainly focus on the machine learning methods, which include supervised learning, unsupervised learning, and deep learning models. All the algorithms are written in MATLAB and C++.

This PhD project mainly rely on the machine learning algorithms to infer the real-time driver intention. The reasons of using machine learning can be multi-fold. Firstly, the real-time traffic context and driver behaviour data can be high dimensional and large volume, very few mathematical models can deal with such data. However, machine learning algorithms are useful for high-dimensional multi-modal data processing. Secondly, the utilization of machine learning algorithm enables learning the long-term dependency between driver behaviours and traffic context, which significantly increase the inference accuracy for the lane change intention. Finally, it is hard to find the intention generation and inference patter based on observation and modelling. The machine learning algorithms provide an efficient way to learn knowledge from the naturalistic data. With some advanced deep learning technique, it is even possible to achieve an end-to-end learning process. Although machine learning algorithms are very powerful in dealing with the tasks in this project, the limitation are also obvious. The major limitation of using machine learning algorithms is the data collection. Data is the heart for the machine learning algorithms. To obtain an accurate intention inference results, several experiments need to be designed and data need to be collected. Insufficient data volume will lead to overfitting and bad inference results. Besides, most of the data used in the project need manual labelling, which is time-consuming. Finally, the training and testing of machine learning algorithms give rise to a higher computational burden both for the financial and temporal cost.

1.3 Thesis Outline

The main content of this thesis is divided in chapters per paper according to the current research outcome. The thesis starts with an overview to the current issues related to the ADAS, intelligent and automated vehicles. This part serves as motivation to the following research and states the application area of the proposed technology. Part II provides the analysis of human intention as well the driver intention. Literature surveys about driver

lane change focus on the state-of-art of the intention inference and the challenges. Part III covers literature review of lane detection techniques, and the integration methodologies for lane detection system. Meanwhile, a parallel driving framework for lane detection namely the parallel lane detection system is introduced. Part IV illustrates the driver behaviour recognition studies with respect to the lane change manoeuvre prediction. Part V covers the lane change intention inference framework and the prediction results using naturalistic data. Finally, Part VI concludes the whole thesis and discuss the challenges as well as the future works. The organisation of this thesis is summarised as follows.

Chapter 1 – Introduction and motivation for the research detailed throughout the thesis.

Chapter 2 – **Paper I:** provides the studies about human intention as well as the driver intention. Driver intention is classified into different categories according to different criteria. The lane change intention is also surveyed from the sensory level, algorithm level, and the evaluation level.

Chapter 3 – **Paper II and paper III:** as traffic context is the major reason for lane change manoeuvres, one of the most important feature namely lane detection is studied in this part. Paper II covers a sufficient survey of the lane detection techniques and the integration with other on-board systems. A novel parallel lane detection framework is also introduced. Paper III introduces a novel integrated lane detection and evaluation system to improve the robust of the lane detection system.

Chapter 4 - **Paper IV and paper V:** driver behaviours are the major clues for driver intention inference since driver behaviours carries rich information that can reflect the mental states of the driver. Paper IV provides a driver behaviour and secondary tasks recognition algorithm based on driver head and body feature selection. Paper V moves a step further, which use an end-to-end deep learning approach to avoid complex feature extraction and selection procedure.

Chapter 5 – **Paper VI:** driver intention inference algorithm should be able to learn the dependency roles among the temporal sequence data. In this module, a driver lane change intention inference algorithm that based on the deep learning method is proposed. Multi-modal naturalist data that are collected on highways are fed into the DII system. The algorithm can predict driver lane change intention few seconds before the lane change manoeuvre initiate.

Chapter 6 – concluded the final remarks, discussions, and the future work required for driver lane change intention inference.

Figure 1-1 illustrates the connection between the thesis parts and chapters. Table 1.1 summarises the list of paper submitted during the research that are fully related with the concept hereby presented. An extension to the driver workload based on the proposed theory and algorithms is included in the Appendix with Paper VII.

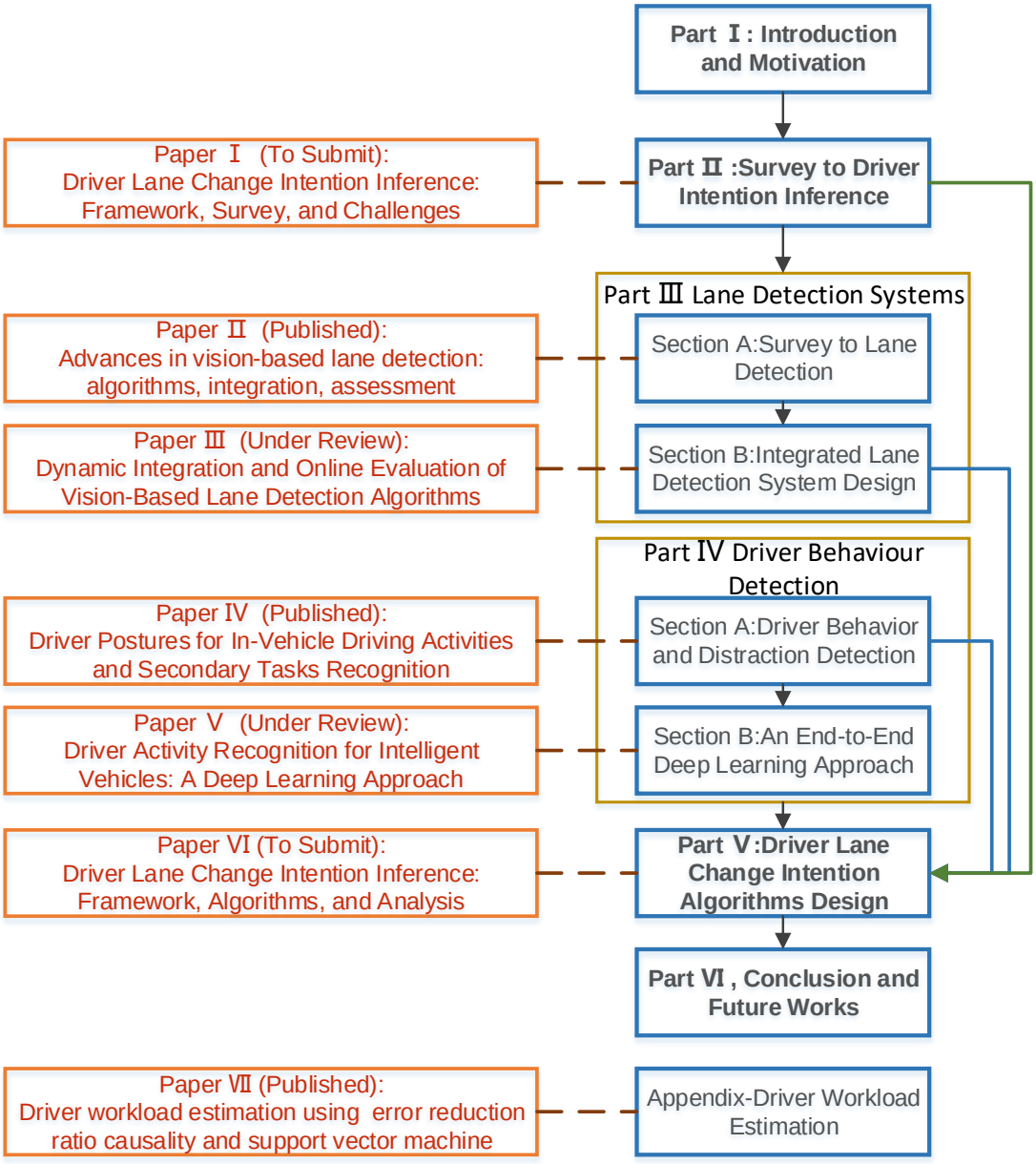

Figure 1-1. Thesis outline and chapters connections.

TABLE 1-1 LIST OF PAPERS PUBLISHED AND UNDER REVIEW COMPLETED DURING THE PHD RESEARCH FULLY RELATED WITH THE PROJECT

Journal Paper I	To Submit	Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang. Driver Lane Change Intention Inference for Intelligent Vehicles: Framework, Survey, and Challenges.
Journal Paper II	Published in IEEE/CAA JAS	Y. Xing, C. Lv, L. Chen, H. J. Wang, H. Wang, D. P. Cao, E. Velenis, F. Y. Wang, "Advances in vision-based lane detection: algorithms, integration, assessment, and perspectives on ACP-based parallel vision," IEEE/CAA J. of Autom. Sinica, vol. 5, no. 3, pp. 645-661, May. 2018.
Journal Paper III	Under Review in IET ITS	Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis. "Dynamic Integration and Online Evaluation of Vision-Based Lane Detection Algorithms". IET Intelligent Transport Systems
Journal Paper IV	Published in IEEE TCSS	Xing, Yang, *et al*. "Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition." IEEE Transactions on Computational Social Systems 5.1 (2018): 95-108.
Journal Paper V	Under Review in IEEE TVT	Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang. Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach. IEEE Trans. on Vehicular Technology
Journal Paper VI	To Submit	Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang. Driver Lane Change Intention Inference: Framework, Algorithms, Testing Results, and Analysis.
Journal Paper VII	Published in Measurement	Yang Xing, Chen Lv, Dongpu Cao, Huaji Wang, and Yifan Zhao. "Driver workload estimation using a novel hybrid method of error reduction ratio causality and support vector machine." Measurement 114 (2018): 390-397.
Conference Paper	Accepted	Yang, Xing, Chen Lv, Dongpu Cao, Efstathios Velenis1, and Fei-Yue Wang. End-to-End Driving Activities and Secondary Tasks Recognition Using Deep Convolutional Neural Network and Transfer Learning. 2018 Intelligent Vehicle Symposium.

1.4 Reference

[1] Angell, Linda S., J. Auflick, P. A. Austria, Dev S. Kochhar, Louis Tijerina, W. Biever, T. Diptiman, J. Hogsett, and S. Kiger. *Driver workload metrics task 2 final report*. No. HS-810 635. 2006.

[2] Peden, Margie. "World report on road traffic injury prevention." Geneva, World Health Organization. 2004.

[3] Fundación Instituto Tecnológico para la Seguridad Del Automóvil (FITSA), 2008. El valor de la seguridad vial. Conocer los costes de los accidentes de tráfico para invertir más en su prevención (in Spanish). Report funded by the Spanish General Directorate of Traffic, Universidad Politécnica de Madrid, Madrid.

[4] Kowsari, Taha. Vehicular Instrumentation and Data Processing for the Study of Driver Intent. Diss. The University of Western Ontario, 2013.

[5] Geronimo, David, *et al*. "Survey of pedestrian detection for advanced driver assistance systems." *IEEE transactions on pattern analysis and machine intelligence* 32.7 (2010): 1239-1258.

[6] Ortiz, Michael Garcia. "Prediction of Driver Behaviour." PhD diss., Universitätsbibliothek Bielefeld, 2013.

[7] McCall, Joel Curtis. "Human attention and intent analysis using robust visual cues in a Bayesian framework." Diss. The University of California, San Diego.(2006)

[8] H. Berndt and K. Dietmayer, "Driver intention inference with vehicle onboard sensors," 2009 IEEE Int. Conf. Veh. Electron. Saf., pp. 102–107, 2009.

[9] Eriksson, Alexander, and Neville A. Stanton. "Takeover time in highly automated vehicles: noncritical transitions to and from manual control." *Human factors* 59.4 (2017): 689-705.

[10] Nilsson, Josef, Paolo Falcone, and Jonny Vinter. "Safe transitions from automated to manual driving using driver controllability estimation." *IEEE Transactions on Intelligent Transportation Systems* 16.4 (2015): 1806-1816.

[11] Lv, Chen, et al. "Analysis of autopilot disengagements occurring during autonomous vehicle testing." IEEE/CAA Journal of Automatica Sinica 5.1 (2018): 58-68.

PART II:

LITERATURE REVIEW. State-of-art of driver lane change inference

2

PAPER I – Survey to Driver Lane Change Intention Inference

Driver Lane Change Intention Inference for Intelligent Vehicles: Framework, Survey, and Challenges

Authors:
Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis

Abstract

Lane change manoeuvre on highway is an interactive task for human drivers. The intelligent vehicles and the advanced driver assistance systems (ADAS) need to have proper awareness of the traffic context as well as the driver to assist the driving tasks. Besides, it would be better if the ADAS understand the driver potential intent correctly since it shares the control authority with the human driver. This study provides an overview on the driver intention inference, especially focus on the lane change manoeuvre on highways. The lane change manoeuvre is one of the most common and complex task during driving as it requires both longitudinal and lateral control actions. In this study, to have a general understanding about the driver intention, a human intention mechanism is discussed in the beginning. Next, the driver intention was classified into different categories according to different criteria. The driver intention inference system is divided into different modules, which consists of traffic context awareness, driver states monitoring, and vehicle status measurement module. The relationship between these modules and the corresponding impacts on the driver intention inference are analysed. The lane change intention inference system is reviewed from the input signals, algorithms, and evaluation aspects, respectively. Finally, the challenges and future works for driver intention inference are discussed.

Index Terms - Intelligent vehicle, ADAS, lane change, driver intention, driver states monitoring.

2.1 Introduction

Traffic accidents statistics have shown that more than 80% traffic accidents were caused by driver errors [1]-[3]. Various passive safety systems like airbags and seat-belts have played a significant role in the protection of the driver and passengers. Although these technologies have saved a lot of lives, they are not designed to prevent accidents from happening but only protect the passengers after the accident happens [4] [5]. Therefore, many efforts have been devoted on the development of safer and intelligent systems towards the prevention of accidents instead of only minimizing the impact of the accidents. The most successful active safety system is the ADAS.

Most of the ADAS techniques such as LDA, LKA, and SWA can assist the drivers to make the right decisions and reduce their workloads [6]-[8]. However, the inputs of these systems are vehicles dynamic states like the steering wheel angle, and traffic information, while most of these systems ignore the most important factor, the driver. Vehicles operate in a three-dimensional environment with continuous driver-vehicle interactions. Drivers are the major component in this system, whom control the vehicle according to the surrounding traffic context. Therefore, allowing ADAS to understand drivers' intention and behaviours is of important to the driver safety, vehicle drivability, and traffic efficient.

Driver intention inference is an ideal way of allowing the ADAS to understand the driver. The reasons of recognizing driver intention are multi-folds: firstly, driver intention inference improve the driving safety. Specifically, there are two different driving scenarios which require inferring the driver intention: to better assess the risk in the future and avoid making decisions that are opposite to driver's intent [9]. For the first case, there is an evidence that a large amount of accidents are caused by human error or misbehaviour, cognitive overload, misjudgement, and operational errors[10]. Monitoring and correcting driver intention in time are crucial to the effectiveness of ADAS. Meanwhile, the increasing usage of the in-vehicle devices and other entertainment devices can distract the driver from the normal driving. For the design of intelligent ADAS, it is beneficial to understand the driver intentions and execute correct assistance actions [11] [12].

In terms of making the right decisions, ADAS systems intervene the vehicle dynamics and share the control authority with the driver. To ensure the efficient cooperation, it is important for the ADAS to aware the driver intention and not operate against the driver's willingness. For example, in the complex traffic conditions such as at the intersection and

roundabout, it is crucial not to interrupt the driver, especially not to interrupt the driver with misleading instructions. This makes it reasonable for the ADAS systems to accurately understand the driver intention in real time.

Furthermore, driver intention inference system benefits the development of future automated vehicles. Driver intention inference can be used to construct driver model, which can act as the guidance for the design of the automated driver. Moreover, in terms of the level three automated vehicles according to the SAE international standard, accurate driver intention prediction enables a smoother and safer transition between driver and the autonomous vehicle controller [13]-[15]. When level three automated vehicles working at automated condition, all the driving manoeuvres are handled by the vehicle, however, once an emergent situation occurs, it has to disengage and give the driving authority to the driver. In such case, the vehicle can determine whether the driver is ready to takeover or not by assessing their intention in advance. If the driver has reasonable driving intention, the vehicle should follow the driver's manoeuvre and the assistance system should give less instructions. The application of driver intention inference can make the transition between driver and controller as smooth as possible.

In this section, literatures about the driver intention inference, particularly the lane change intention will be reviewed. The contribution of this study can be summarized as follows. Firstly, a state-of-art literature review about driver lane change intention is proposed. Then, the lane change intention inference system is categorized. Secondly, the critical time flow of driver intention inference will be introduced. This leads to a comprehensive understanding about the architecture of the intention inference system. Finally, future works of driver intention inference will be proposed and discussed to benefit the development of future intelligent vehicles.

2.2 Human Intention Mechanism

Human intention has been theoretically studied in the past. From cognitive psychology perspective of view, intention refers to the thoughts that one has before the actions [16]. Similarly, in this study, intention is the attitude of performing a series of vehicle control manoeuvre. The intention is determined by three aspects: the attitude towards the behaviour, subjective norm and the perceived behaviour control [17]. In [17], human behaviours are found to be the response of the intention. The attitude towards the behaviour describes how willing and how much effort the human want to take the

behaviour, a strong level of attitude can give a strong willingness of taking actions in the certain task. Secondly, subjective norm reflects the pressure from the surrounding social life of the human. Finally, the perceived behaviour control was developed from the self-efficacy theory. It describes the confidence of an individual to perform the behaviour.

Bratman pointed out that intention is the main attitude that directly influence the future plans [18]. In addition, Heinze described a triple level description of the intentional behaviour which contained intentional level, activity level, and state levels [19]. In the human-machine-interface scope, according to [20], intention recognition is the process of understanding the intention of another agents. More technically, it is the process of inferring an agent's intention based on their actions. Elisheva proposed a cognitive model with two core components, which were intention detection and intention prediction [21]. Intention detection refers to detect whether a sequence of actions has any underlying intention. Intention prediction, on the other hand, refers to the prediction of the intentional goal based on a set of incomplete sequence of actions. The intention inference and reasoning process makes people clever and enable them to take part in the social community. Human can recognize other's intention based on their observation and the social skill knowledge. However, it is difficulty to make the intelligent machine such as an intelligent vehicle to learn how to inference human intention accurately. To some extent, only when a robot can detect human intention based on their own observation can they be viewed as intelligent agent.

Human intention inference has been widely studied in the past decades. One of the most significant applications of human intention inference is human-robot-interface design [22]-[25]. Thousands of the service robots were designed to assist the human to complete their works either in daily life or in dangerous workspace. The traditional robots were designed from the robot perspective of view rather than from human point of view, which reduces the interaction level between human and robot. To improve the efficiency of human-robot-interaction (HRI) as well as the intelligence of robot, robot should have the ability to learn and infer human's intention and obtain basic reasoning intelligence.

A widely accepted classification way for human intention in HRI scope is to classify the human intention into explicit and implicit intention. Implicit human intention can be further separated into informational and navigational according to [27]. Explicit intention is clearer than the implicit intention and hence, easier to be recognized. Explicit intention

means the human directly transmit their intention to the robot by language or directly command through the computer interface. While implicit intention reflects human mental state without any communication with the robot [26]. The robot has to observe and understand the human behaviour first, then they make an estimation of the human intention at the right movement based on the knowledge base. Human intention inference problem contains a large amount of uncertainty, and noise exists in the measurement device. Therefore, probabilistic-based machine learning methods are powerful tools in solving this kind of problem and it has been successfully applied in many cases. In terms of the human intention inference task, which usually equals to infer human mental hidden states. The Hidden Markov model (HMM) as well as the dynamic Bayesian theory are two popular methods for the inference of human mental state [28]-[30].

2.3 Driver Intention Classification

Driver intention can be classified into different categories from different perspective of views. For example, it can be classified according to the motivation, time-scale, and the direction of driving. Among these, two most straightforward classifications ways are based on the time scales of the intention and the driving direction.

2.3.1 Time Scale based Driver Intention Classification

In terms of the time-scale based classification method, Michon pointed out that the cognitive structure of human behaviour in the traffic environment is a four levels hierarchical structure, which contains road user, transportation consumer, social agent, and psycho-biological organism [31]. Among these, the road user level is directly connected with the drivers and can be further divided into three sublevels: strategy, tactical, and operational level (also known as control level) respectively, as shown in Figure 2-1. The three cognitive levels can be viewed as three driver intention level based on the time-scale characteristic. Strategy level defines the general plan of a trip such as the trip route, destination, and risk assessment, etc. The time constant will be at least in minutes or even longer. At this moment, driver will decide the transport mobility and comfortable issues, which is a long time-scale problem. In terms of the tactical level, which the time constants are in seconds, the driver will make a short-term decision and control the vehicle to negotiate the prevailing circumstance. Tactically planned

intentional manoeuvre consist of a sequence of operational manoeuvre to fulfil the short-term goal such as the turning, lane changing, and braking manoeuvre [32].

All the control commands must meet the criteria from the general goal that are set at the strategically level. Lastly, the operational intention is the shortest one among the three levels and stands for the willing of the driver to remain safe and comfortable in the traffic situation. The driver directly gives control signals to the vehicle and the time constant are in normally milliseconds. As mentioned earlier, real-time lane change intent inference plays a critical role in the improvement of driving safety. In addition, continuous lane change intention inference is a relative complex and difficult task than some other driving intentions. Salvucci and Liu concluded that lane change was not simply a control procedure but also incorporated a set of critical aspects of driving such as lower level controls [33]. Normally, lane change manoeuvre will contain a series of short term driving behaviors like the acceleration and deceleration in the longitudinal direction and the steering wheel control in the lateral direction.

There are also some other classification methods for driver intention. Salvucci developed a driver model, namely, Adaptive Control of Thought-Rational cognitive architecture [34]. Similar to the three-level architecture of road user model given by Michon, Salvucci developed the integrated driver model into three main components which are control, monitoring and decision-making modules. The control component is like the operational level given by Michon, which is responsible for the perception of external world and transfer the perceptual signals directly to the vehicle. The monitoring component keeps aware of the surrounding situation and environment by periodically perceiving data and inferring. The decision component, which has same function with part of Michon's tactical level, makes tactical decisions for each manoeuvre according to the awareness of current situation and the information gathered from control and monitoring module. One significant advantage of the cognitive driver model is the incorporation of the built-in features enables to mimic human abilities.

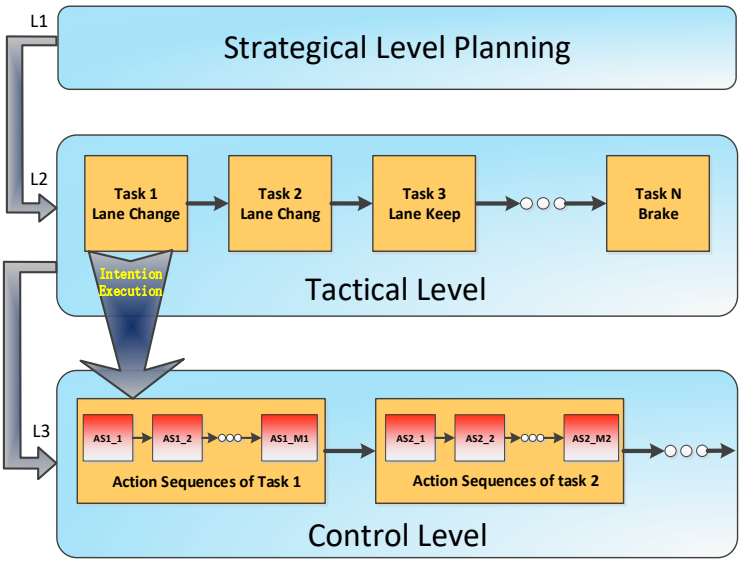

Figure 2-1. Driver intention classification based on time constant.

2.3.2 Directional based Driver Intention Classification

The direction-based driver intention classification, on the other hand, is quite straightforward. There are two basic directions for the underground vehicle, which are the longitudinal and lateral intention, respectively. Driver's longitudinal behaviour contains braking, acceleration, starting, and lane keeping, etc. Lateral behaviours usually contain turning, lane changing, and merging. In previous studies, most researchers pay attention to the lateral intention prediction such as the lane change, turn, and overtaking manoeuvres. The lateral intentions are more complicate than the longitudinal intention dues to the frequent interaction with surrounding vehicles.

In terms of the longitudinal intention, most of the previous studies focus on the braking intention recognition. Haufe, *et al.* proposed a driver braking intention prediction method using EEG (electroencephalograph) and EMG (electromyography) signals [35] [36]. Khaliliardali proposed a driver intention prediction model to determine whether the driver want to go ahead or stop [37]. The method was to classify the go and stop intention based on the brain-machine-interface. The, EEG, EMG, and EOG (electrooculography) signals from six subjects in the simulation environment were collected and two classification methods (linear and quadratic discriminant analysis) were used separately to evaluate the classification performance. McCall and Trivedi integrated driver intention into an intelligent braking assistance system [38]. A sparse Bayesian learning algorithm was used to infer the driver's intention of braking. Trivedi *et al.* predicted the driver's braking intention by directly monitoring the foot gesture through cameras [39] [40]. They showed

that the driver foot gesture plays an important role in the vehicle control. Therefore, the usage of vision-based foot tracking is more directly and accurately. Mabuchi and Yamada estimated driver's stop and go intention at intersections when the yellow light occurs [41]. Takahashi *et al*. predicted the driver deceleration intent during downhill road [42]. Kumagai *et al*. proposed a method to predict the driver's braking intention at intersections (particular right turns) by using dynamic Bayesian network [43].

As aforementioned, direction-based intention classification is less precise compared with the time-scale based methods as the driving manoeuvres can be very complex and contain multiple short-stage actions. For example, the lane change manoeuvres can consist of short period acceleration and turn, it is less accurate to describe the intention as merely longitudinal or lateral.

2.3.3 Task based Driver Intention Classification

Driver tactical manoeuvres consists of a series of operational manoeuvres. Some of the existing studies focus on the analysis of multiple tactics rather than single tactical task. Multi-task-based model usually contains both longitudinal and lateral manoeuvre compared with the single-task oriented model. By using machine learning theory, single intention inference task can be modelled with discriminative and generative model. However, multiple-tasks inference model prefers to use the generative models such as Bayesian networks and HMM [32]. Oliver and Pentland proposed a driver behaviour recognition and prediction method based on the dynamic graphical models [44]. Seven driver manoeuvres which were passing, changing right and left, turning right and left, starting, and stopping were analysed. Liebner inferred the driver's intent based on an explicit model for the vehicle's velocity, and an intelligent driver model was used to represent car-following and turning behaviour [45]. Liu and Pentland aimed to analyse the patterns within a driving action sequence [46]. The main approach was to model the human behaviours in a Markov process. Imamura developed a driver intention identification and intention labelling method based on the assumption of compliance with traffic rules [47].

As can be seen, multi-tasks driving intention recognition is more complex than the signal task. It must define and clarify several driving manoeuvres and need more experiments. Moreover, it relies on the design of generative inference algorithms.

2.4 Driver Intention Inference Methodologies

2.4.1 Architecture of Driver Intention Inference System

Driver intention inference system requires multiple techniques such as vision-based perception system, data fusion and synchronization, and model training based on machine learning methods. According to the previous literatures, Lane change intention inference system mainly contains the following modules: road and traffic perception module, vehicle dynamic measurement module, driver behaviour recognition module, and driver intention inference module, as shown in bounding box in Figure 2-2 below.

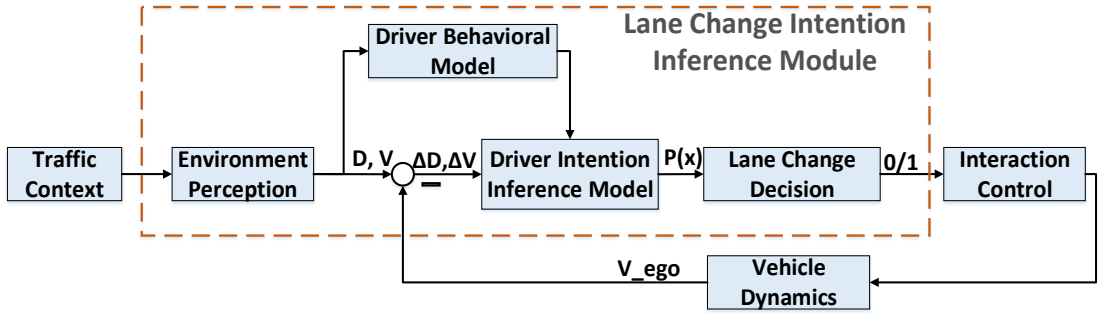

Figure 2-2. Driver lane change intention inference framework.

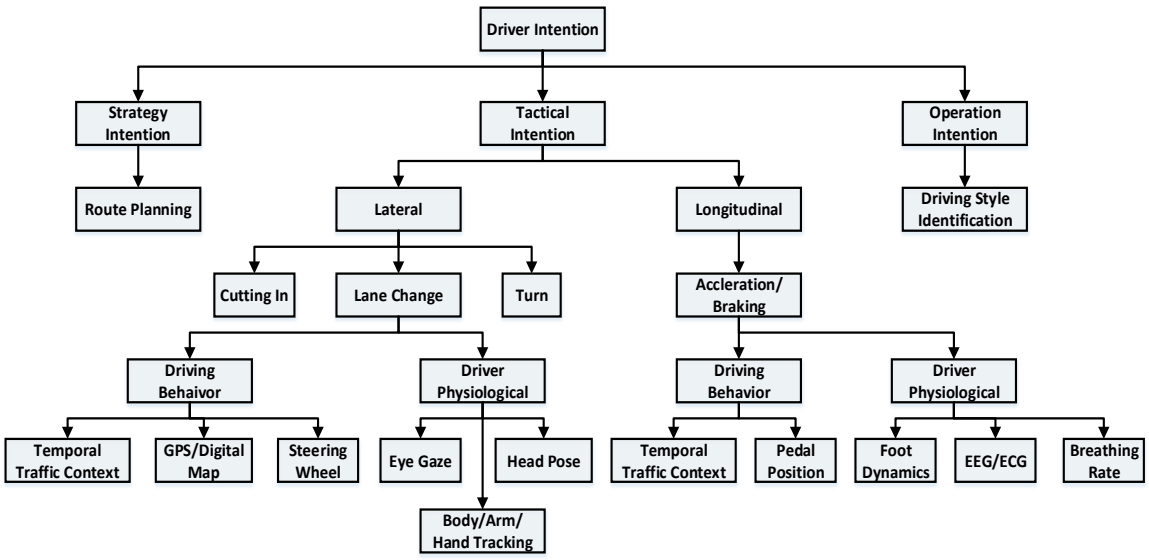

Figure 2-3. Taxonomy of driver intention systems.

As shown in Figure 2-2, the traffic information is firstly captured by the environment perception block. Similar to previous literatures which use cameras, light detection and ranging (Lidar), radar, and GPS signals to detect the surrounding traffic situation, this block will process the current road and traffic information with machine vision techniques and output the position of ego-vehicle and the velocity. After the traffic context being

detected, the relative distance and velocity between the ego-vehicle and the front vehicles can be obtained with the collection of vehicle status information captured through the CAN bus.

The traffic and vehicle data will be fed into the intention inference model along with the driver behaviour signals. The driver behavioural signals contain the driver head rotation, eye gaze, as well as the body movement, etc. Next, the intention inference model will calculate the probability of a lane change intention based on the fused information. Once the decision is made, the lane change decision module outputs a binary signal to indicate a lane change manoeuvre. After the lane change decision is activated, the interaction module models the driver hand and foot dynamics and the interaction with vehicle control interface. A taxonomy of driver intention modules is depicted in Figure 2-3

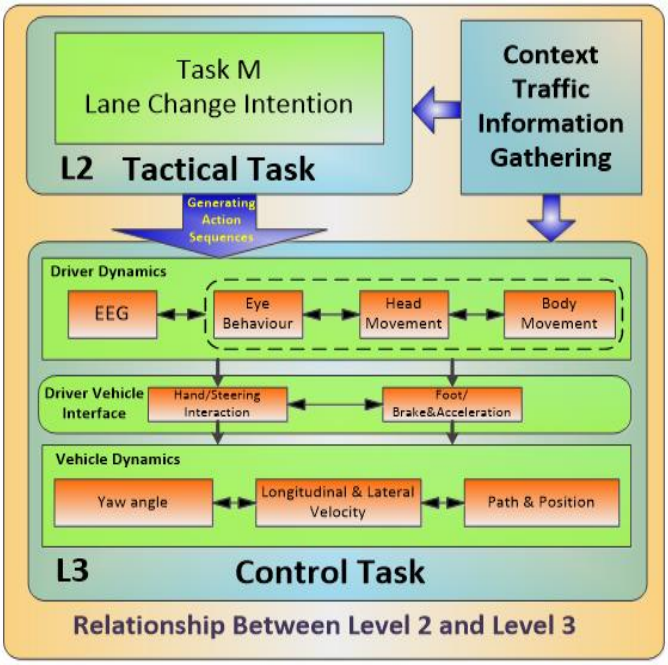

Figure 2-4. Relationship between tactical intention and operational intention with respect to multi-modal inputs.

The relationship between the tactical intention and the operational intention is illustrated in Figure 2-4. Figure 2-4 contains three parts namely the traffic context perception unit, level two tactical intention unit, and level three control units. Specifically, in the third level, three layers are defined. The upper layer is driver dynamics, which represent the checking and monitoring behaviour of the driver. The second interface layer will be activated once the lane change decision is made. Finally, the control signals are

fed into the lowest vehicle control layer. In the driver dynamics module, the most common dynamics are brain dynamics which can be measured by EEG, eye gaze behaviour, head movement, and body movement (contains hand, body, and foot dynamics, etc.).

From Figure 2-4 we can clearly define the time flow of the driver intention procedure. Driver first captures the traffic context and then generate corresponding intention according to the traffic. Next, driver will check the surrounding traffic by performing a series of checking behaviours to make sure a safety control. Once the driver is confidence with the lane change decision, he will control the vehicle through steering wheel and the pedal. Finally, the vehicle response to the relative control behaviours and the vehicle dynamic changes.

2.4.2 Inputs for Driver Intention Inference System

Driver is in the centre of the Traffic-Driver-Vehicle (TDV) loop. The signals from the three parts of the TDV loop can be used to infer human driver intentions. The major signals that can be used as the inputs of DLII system is summarized in Table 2-1.The perception of traffic context enables the ADAS understand the specific driving environment of the human driver so that a more reasonable intention inference can be detected. The driver behaviours information will help to determine how long the driver has generate the intention and how the driver check the surrounding traffic with respect to the corresponding intention. Finally, the vehicle dynamic signals indicate what kind of driving action the driver has taken to realize the intent.

TABLE 2-1 COMMON INPUT SIGNALS AND SENSORS USED FOR DRIVER INTENTION INFERENCE.

Sensor Sources	Sensor Categories
Traffic	Current ego-vehicle position (collected with GPS and digital map), Relative distance, velocity and acceleration with respect to front and surrounding vehicles (collected with Cameras, radar or Lidar).
Vehicle	CAN bus signals (including steering wheel angle, steering wheel velocity, brake/gas pedal position, velocity, heading angle, etc.)
Driver	Cameras (Head rotation, gaze direction, foot dynamics). EEG, EMG, Heart rate, etc.

2.4.2.1 Traffic Context

Traffic context is the major stimuli for the driver intention. A better understanding of the surrounding traffic information will improve the intention inference accuracy. For example, lane changing can occur when there is a low speed vehicle front of the host vehicle. There are many kinds of sensors that can be used to capture the surrounding traffic context, such as the camera, radar, and Lidar systems. The most popular vision-based ADAS systems are LDW and LKA. Vision-based LDW is able to compute the

distance between host vehicle and the lane boundary, the vehicle lateral velocity and acceleration, yaw angle, and road curvature, etc. [48]. Adaptive Cruise Control (ACC) is a cruise system, which can be divided into laser-based and radar-based. The relevant distance for host vehicle with front vehicles can be detected through radar of the ACC system [49]. SWA system uses at least two radars that are mounted under the side mirrors to monitor the rear and side vehicles. The scan area can be up to 50 meters behind the vehicle [50] [51]. In [51], the authors evaluated the impact of different sensors on the prediction of driver intention. Bernt designed a HMM-based intention classifier which extracted the distance to the next turn, the street curvature, and street type from a digital map [52]. Rafael introduced an interactive multiple-model (IMM) based approach to predict lane change manoeuvres on highway. The system used the GPS/IMU sensors to collect the vehicle position data [53]. The advantage of GPS system is that it is able to give the location and time information in rough weather conditions, at which moment the camera and radar system cannot work.

In [54], McCall and Trivedi proposed a preliminary work for the study of driver-centred assistance system, which focus on the lane change events. A modular scalable architecture was provided to capture the driver behaviours and the surrounding environment. Radar and video devices are used to obtain the forward, rear, and side information. Meanwhile, inner cameras were also used to monitor the driver foot gesture and head movement. The work in [55] concentrated on the lane change intention prediction according to the sensory data. The data contains the lane information given by a lane tracker, the vehicle velocity, lateral position and its derivation, and the steering wheel angle.

2.4.2.2 Vehicle Dynamics

Vehicle status information such as steering wheel angle, brake pedal position, and velocity can be viewed as the direct response to the control actions. This information reflects the driver's control actions that are taken on the vehicle. Hence, these signals have been widely used in many works for driver intention identification. Vehicle data can be collected from the vehicle CAN bus such as the velocity, acceleration, heading angle, etc., which can reflect the driver control to the vehicle, which can process a large amount of data with a high transfer speed. In [56], vehicle speed, acceleration pedal, and brake pedal position are collected to predict the driver braking intention. In terms of the lane change intent, throttle pedal position, brake pressure, cross acceleration, steering wheel

angel, steering wheel angel velocity, yaw rate, and velocity are collected from CAN bus in [50][52]. These signals are particular useful to understand the driving intent after the driver has determine the finish the intention. Schmidt and Beggiato proposed a lane change intention recognition method based on the construction of an explicit mathematical model of the steering wheel [57]. In [58], the authors proposed a driver lane change/keep intention inference method on a driving simulator. The input signals contained the steering wheel angle, acceleration, and the relative speed of front vehicle. In [59], the speed, transmission position, steering angle velocity, steering angle, acceleration pedal position, and lateral acceleration were captured on a simulator for lane change intention inference. The authors in [60] proposed a driver intention recognition method based on artificial neural networks. CAN bus data and driver gaze information were collected and fed into the intention model. The experiment was designed in a six degrees-of-freedom dynamic driving simulator and a total 284 lane change and lane keeping instances are recorded.

Although vehicle dynamic data reflect the response to the driving actions, it gives a delayed information compared with the driver behaviour and traffic context information with respect to the intention inference. Vehicle data can reflect the driver intention only after the manoeuvre has been initiated. Meanwhile, the vehicle dynamic status-based mathematical model has limited ability in the prediction of driver intention. Although the vehicle data gives limited contribution to the prediction of driver intention, it is still an important data source that can increase the accuracy of the intention identification and help to predict the intent at an early stage after the intended behaviour being initiated.

2.4.2.3 Driver Behaviours

Unlike the CAN bus data, driver behavioural signals such as the head and eye movement can give an early clue about the driver intention. Many studies have evaluated the impact of head/eye movement on the intention prediction. In [61], the authors applied the pupil information as the cognitive signals for the lane change intent prediction. Normally, driver eye movement can be classified into intention guided or non-intention guided. Intention guided eye movement means the eye fixation or saccades is in purpose, while non-intention-based movement may due to the distraction issues. Driver visual fixation will no longer follow driver' attention when the driver is distracted. At this moment, eye movement can neither reflect the driver's mental purpose nor the intention prediction result being trust [63]. In terms of the intention-oriented eye tracking, it can be

view as a cognitive progress of information gathering, which can reflect the driver's mental state earlier than the vehicle parameters. In addition, the driver intention at information gathering step is less likely to change when compared with that in the action execution step [64]. Although head/eye movement can be caused by distraction, at most of the time, driver shifts the eye gaze in purpose, which makes eye movement a useful signal for the intention decoding and inference [65]. Doshi and Trivedi evaluated the relationship between gaze pattern, surrounding traffic information, and the driver intention [66].

or irrelevant stimuli lead to different eye movement. Many studies have paid attention to the eye tracking techniques to predict driver intention [67]-[69]. It has been proved that the eye movement information do improve the intention prediction accuracy and help to decrease the false alarm rate.

A significant challenge to eye movement detection is the eye tracking. The eye movement is normally detected with cameras mounted on the dashboard or the wearable eye tracking system. Dues to the physical characteristics of the eye (small scale and occlusion, etc.), it is not easy to detect the eye and track the pupil robustly. Moreover, the glass, lightness, and even hairs near the eye can influence the eye tracking performance. According to these challenges, some robust algorithms for eye movement detection have been proposed [70]-[72]. Lethaus *et al*. [73] evaluated how early and how much gaze data can be used to predict the driver intention. In their experiment, vehicle data was collected in a driving simulator and the eye data was captured by the SMI eye tracking system. They finally concluded that a 10 seconds window for the eye gaze data is enough for intention prediction, while a 5 seconds window gives a better performance since the 10 seconds window carries more noise.

Similar to the gaze direction, head motion is another cognitive process for information gathering. In [62], the authors proposed a driver head tracking system for driver lane change intention inference. The head tracking system consists of six cameras. Head movement was regarded as a more important factor than the eye gaze for driver intention prediction [68]. Head movement was widely adopted for driver intention inference [68] [74] [75]. In [68], the authors claimed that both eye and head movement are useful data for the detection of driver distraction, attention, and mental state inference. However, there is a difference between the eye and the head movement for the classification of

driver mental state. Specifically, Head moves earlier than the eye when the driver is executing a mental goal-oriented task. On the other hand, when the outer stimuli occur, driver's eye will shift first, and the head moves later. This is an interesting conclusion since it offers a way to determine whether the ongoing driver behaviour is goal-oriented or stimuli-based.

Murphy and Trivedi [76] concluded that several head pose estimation algorithms can be used to the track head movement. However, in-vehicle head tracking system facing its own problems. One significant challenge is the online computing ability of the on-board processor. In [77], a processing method for in-vehicle driver eye/head tracking system was proposed aiming at handling the naturalistic eye/head tracking system for driver distraction detection. The data processing method was able to improve the sensitivity and specificity of the eye tracking for 10%. Another challenge is the noise issue. Vibration from the road and lightness variation issues exist in the vehicle cabinet, which bring a large amount of noise to the captured images. Sometimes head tracking data will get lost dues to the algorithm problems and head tracking usually need re-initializing [78]. Besides, according to the current head tracking algorithms, those developed with a monocular camera show a worse performance than the multi-cameras-based algorithms. However, using multiple cameras in the vehicle will increase the system cost.

In addition to the eye and head tracking system, some other driver behavioural signals like EEG, foot, hand, and body gesture were also studied in some literatures [79]-[82]. A carefully selected driver behavioural feature will improve the performance of driver intent prediction.bEEG measures the flow of brain electric currents with the non-invasive electrodes on the scalp. It has been widely used in cognitive neuroscience for the study of brain activities. Meanwhile, EEG is also an important sensor for brain-computer-interface (BCI) design. EEG is sensitive to the small changes in the electrical activities, which is suitable to detect human mental state. Therefore, many studies have evaluated the impact of EEG on the detection of driver mental activity. A direct application of EEG to the in-vehicle BCI design is driver workload detection. EEG has been widely used for driver workload monitoring and other status like drowsiness, happiness, sadness, mental fatigue, and abnormal condition detection [80].

However, a drawback of the EEG signals is it usually contains various noise, hard to acquire the signal, and getting weak if sampled with poor quality [83]. This is because

the brain electric current is detected with a non-invasive method that the signals must cross brain layers, skull, and scalp. EEG signals are a set of high dimensional data with large noise. Hence, machine learning methods are main solution to process the EEG signals. Machine learning algorithms can be used for the feature extraction and classification [84]. Since EEG measures the brain activities, it can reflect the intention faster than the human muscle reaction. It was found that by using EEG, the braking intention can be detected130 milliseconds faster than only consider the brake pedal position [85]. EEG has also been used in driver steering intention prediction [86]. By using the machine learning based classification algorithm, the prediction accuracy for steering intention can achieve 65% to 80%.

2.4.3 Algorithms for Driver Intention Inference

Conventional intention inference algorithms can be roughly divided into the following groups: the mathematical-based model, driver cognitive model, and the widely used machine learning models. Dues to the ability of dealing with high-dimensional feature vector, machine learning methods are widely accepted by the LCII system. As mentioned in [50], the intelligent vehicle adopted 200 sensors signals. At this moment, machine learning methods are the most suitable tools to fuse the signals and construct the LCII system. The machine learning algorithms can be divided into generative models and discriminative models. The generative model provides a joint probability distribution over the observed and target values, which can generate both the inputs and outputs according to some learned hidden states. However, generative model is less easy to be trained compared with the discriminative model since it may require training multiple models and provide a model to each class based on the different probability distribution.

On the contrary, the discriminative model only provides the dependence of the target on the observed data. Discriminative models usually can be generated from the specific generative models through the Bayes rule. The most popular generative models contain Bayesian networks, Hidden Markov models, Gaussian mixture models, etc. whereas the widely used discriminative models are support vector machine (SVM), neural networks (NN), and linear regression, etc. As mentioned in [32], discriminative models lead to a better result on the single target problem than the generative models, while the generative models are more suitable for the multi-target problems. Instead of these two typical methods, driver intention also can be modelled based on the cognitive models and the

deep learning models. A taxonomy of the algorithms for intention inference is shown in Figure 2-5 and Table 2-2 the lane change intention inference results based on some articles are illustrated.

2.4.3.1 Generative Model

Generative models like hidden Markov model are widely used in previous LCII studies [57][58][61][68][87][88]. Pentland and Liu [46] used the HMM to recognize seven kinds of driver intention. Berndt used the HMM to investigate early lane change intention [52]. The aim of study is to identify the left and right lane change manoeuvre at an early stage after the lane change was started. Final results showed that a 71% and 74% recognition accuracy for the left and right lane changes were recognized. In [89], a new feature named comprehensive decision index (CDI) was introduced.

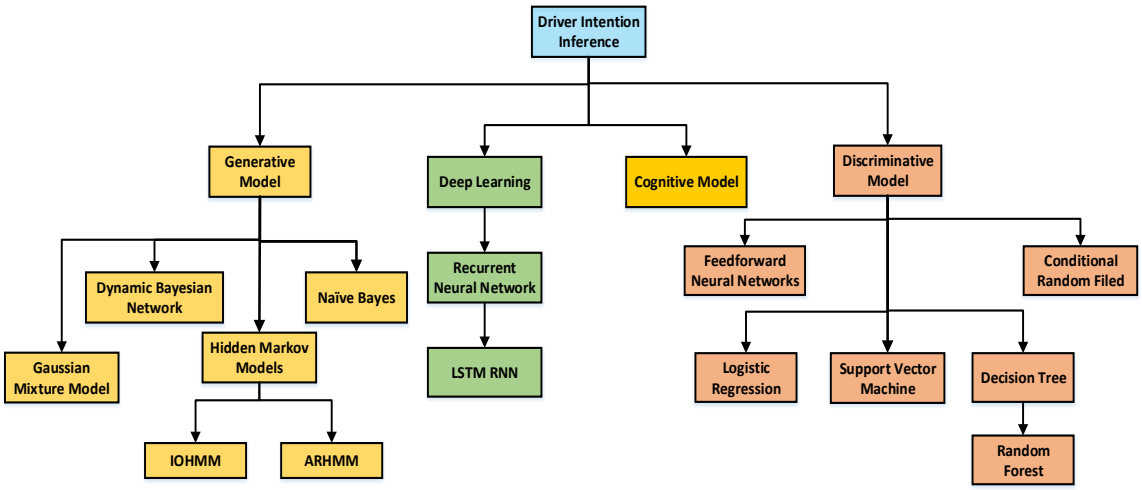

Figure 2-5. Taxonomy of Algorithms for driver intention inference system.

Fuzzy logic was applied to represents the surrounding environment and the lane change willingness of the driver. The overall performances of the algorithm with different input parameters (situation, situation and vehicle, Situation, vehicle and CDI) were analysed. Li. *et al*. proposed an integrated intention inference algorithm based on HMM and Bayesian Filtering (BF) technique [90]. A preliminary output from the HMM was further filtered using the BF method to make the final decision. The HMM-BF framework achieved a recognition accuracy of 93.5% and 90.3% for the right and left lane change, respectively. In [58], the authors proposed a driver lane change/keep intention inference method based on a dynamic Bayesian network. A four-step framework for the driver intention inference was developed and the auto-regression (AR) was combined with hidden Markov model (HMM) to take the past driver behaviours into consideration.

In [59], the authors constructed a lane change intention recognition method based on Continuous Hidden Markov Model (CHMM). The authors evaluated the CHMM performance with different model structures (3, 6, and 9 hidden states, respectively). According to the results, CHMM model with six hidden states and 1.5s window size (data collected between 0s and 1.5s prior the vehicle crossing the lane) gave the best classification result (95.48%). The authors also concluded that the most important factors for lane change intention recognition were data representation, size of the sliding window, and the initial sets of the model parameters. In [91], the authors proposed a context-based highway lane change intention system based on HMM. Four different inference systems were defined, which were vehicle state model only, front vehicle distance-based model, rear vehicle distance based model, and front and rear distance-based model. From the comparison of the four scenarios, the author pointed out that the classification performance did not show significant increase with the additional context information. However, they showed that the additional context information lead to a high false positive and the system performance was worse than the system with vehicle state information only. One possible explanation is that the HMM has limited ability to capture the context information during the lane change process. Therefore, a more powerful algorithm such as double layered HMM and input-output HMM (IOHMM) [98] should be used.

Besides the HMM, other generative models such as Bayesian network and Naive Bayesian classifier are also used for intention detection. For example, in [69], the authors used three different algorithms, which were artificial neural network, Bayesian network, and Naive Bayesian. With a direct comparison, the result demonstrates that the performance of ANN was the best among the three. In [50], a driver lane change behaviour classifier based on a hybrid model that combine Bayesian network and SVM was proposed. In [62], the authors proposed a driver head tracking system for driver lane change intention inference. A Naive Bayesian classifier was trained and used to classify the glance area of the driver. In [92], the authors proposed a lane change detection method based on the object-oriented Bayesian network. The system was designed according to the modularity and reusability of the Bayesian network, which makes it easy to extend the system according to different requirements. The whole system was constructed by various sub-Bayesian networks with different function. Seven main driving manoeuvres

were studied, which were object follow, lane follow, ego-vehicle cut out, object cut out, ego-vehicle cut in, object cut in, and other.

2.4.3.2 Discriminative Model

Discriminative model such as the SVM and ANN are also widely used in LCII dues to the rich background theories and the successful application experience. In [50][51], a Bayesian extension to the support vector machine algorithm, namely, the relevance vector machine (RVM) was used to classify the driver lane change (right and left) and lane keeping intention. To decrease the false alarm rate of the system, a multiple detection suppression technique was used, which under the assumption that consecutive detection arises from the same intention. The classifier can achieve 80% accuracy with a relative low false alarm rate. The authors also examined the classification result with a certain time prior to the manoeuvre happens using different sources of sensors and testing situation. Conclusion was given that the online classification results was worse in real-time environment than that in the experiment environment.

Campbell identified three kinds of driver intention, which were lane keeping, preparing for lane changing, and lane changing [88]. Three classification algorithms were used in the study (support vector machine, random forest and logistic regression). The SVM model achieved the best classification performance compared with the other two algorithms. The authors in [60] proposed a driver intention recognition method based on artificial neural networks. The accuracy for lane change left detection is better than that for the lane change right. The results showed that the head rotation had consistent gains between 1.5s to 2.5s prior to the lane change manoeuvre.

The work in [55] constructed a multiclass classifier by combining the SVM and Bayesian filter (BF). Results showed that the proposed algorithm realized an average of 1.3s prediction of the intention in advance and can achieve a maximum prediction horizon of 3.29s. It was concluded that one of the important tasks in the future is to improve the performance of the lane tracker system by reducing the false alarm rate. The authors in [93] proposed a driver lane change and lane keeping intention classification method based on SVM. In [94], the authors introduced and compared three machine learning methods, which were the Feedforward neural network (FFNN), recurrent neural network (RNN), and SVM. To evaluate the classification performance, four evaluation criteria were introduced, which were mean value of prediction horizon, the number of correctly recognized lane change, the number of not recognizing the lane change, and the number

of false alarms. The results showed that SVM gave the best result followed by RNN. The classifiers were able to predict the lane change 1s to 1.5s prior to the vehicle crossing the lane.

TABLE 2-2 SUMMARIZE OF VARIOUS PREVIOUS LANE CHANGE INTENTION INFERENCE SYSTEMS

Ref.	Signals	Algorithm	No. Subjects	Environment	Performance	Predict Horizon
[59]	Steering angle, steering force, velocity	CHMM	10	Simulator	100%	0.5-0.7s after steering
[68]	Lane Position, CAN bus, Eye and Head	RVM	8	On-road	88.51%	3s prior to the lane change
[103]	Lane Position, CAN, and Head	Sparse Bayesian Learning	3	On-road	90%	3s prior to the lane change
[67]	Eye movement	Finish Questionnaire	17	Simulator	77%	–
[104]	Eye movement (pupil size variation)	SVM	24 Samples	On-road	73.13%±1.25%	–
[52]	CAN, Digital map	HMM	50 LCL, 50 LCR	On-road	71%L, 74%R	–
[91]	CAN, Distance between vehicles	HMM	20	On-road	80%-90%	–
[51]	CAN, LDW, ACC, Head, SWA	RVM	15	On-road	91%	1s prior to the lane change
[44]	CAN, Head, Eye	HMM	70	On-road	12.5%LR,17.6LL	1s prior the manoeuvre
[41]	CAN, Lane style and position, Head, Eye	Relevance vector machine	108 Lane changes	On-road	79.20%	–
[69]	CAN, Eye	ANNs, Bayesian Networks, Naïve Bayes Classifiers	10	Simulator	95.8%(ANN) 93.2%(BNs), 90.6%(NBCs)	–
[87]	Steering angle	Queuing network model	14	Simulator	LCN 98.61% LCE 91.67%,	–
[89]	Steering angle, rate, Comprehensive Decision Index	Fuzzy logic & HMM	4 (69 samples)	Simulator	94%	1.67s after lane change manoeuvre start
[105]	CAN, Eye movement	State Transition Diagram	20 (8576 lane changes)	Simulator	80%	–
[88]	Relative Position and Velocity, Heading, Time-to-Collision, Time Headway	SVM Random Forest, Logistic Regression	5	Simulator	89.5%(SVM), 88.9%(RF), 87.2%(LR)	–
[46]	Steering angle and velocity, vehicle velocity and acceleration	HMM	8	Simulator	88.3%±4.4%	–
[50]	CAN, ACC, SWA, LDW, Head	RVM	15 (500 samples)	On-road	80%	3s prior to the lane change
[106]	CAN, GPS, Eye	Finish Questionnaire	22	On-road	–	–
[55]	Steering angle and relative lane position	SVM and Bayesian Filtering	2 (139 Samples)	On-road	80%	1.3s prior to the lane change
[60]	CAN, Eye	ANN	10	Simulator	95%(L), 85%(R)	–
[107]	CAN, Lidar, Radar, Hand, Head, Foot	Latent Dynamic Conditional Random Field (LDCRF)	1000 samples	On-road	90%	2s prior the lane change
[108]	CAN	SVM and Bayesian Network	4	Simulator	95%(LK), 80%(LC)	–

[34]	CAN, eye movement	Computational model based on ACT-R	11	Simulator	90%	1s after steering
[90]	CAN bus	CHMM and Bayesian Filtering	188LCL, 212LCR, 242 LK	On-road	93.5%(L), 90.3%(R)	0.5-0.7s after steering
[98]	GPS, digital map, head, CAN bus	LSTM-RNN	10 drivers, 1180 miles	On-road	90.5%	3.5s prior the lane change

2.4.3.3 Cognitive Model

Despite of the machine learning algorithms, human cognitive models are also used in some studies. For example, Salvucci introduced a real-time system for detecting driver lane change intention based on a mind tracking architecture [33-34]. The mind tracking computational model continually infers the driver's unobserved intention from the observed actions. The computation model was based on the cognitive model implemented in the Adaptive Control of Thought-Rational (ACT-R) study. The system contains four steps which were data collection, model simulation, action tracking and thought inference. During simulation, the system ran several models simultaneous. Mind tracking detected the driver intentions by examining the "thoughts" of the best matching model. The mind-tracking system achieved 85% accuracy with 4% false alarm rate for the lane change intention detection.

In [87], the authors constructed a queuing network cognitive architecture to model the driver behaviour during normal and emergency lane change manoeuvres. The differences between the outputs of the driver model and the measured data are compared. Driver lane change behaviour model was built by the queuing network model, which contains three main modules: preview module, prediction module and control module. Intention was detected based on a threshold of root-mean-square error value (RMSE). The intention with the smallest RMSE value is determined as the final output. The proposed method achieved a high accuracy (above 90%) and low false alarm rate (0.294%). Comparing with those intelligent inference methods that based on the eye gaze and head moment, this method can be easily extended into real world application. However, since the algorithm was based on the steering wheel angle signal only, it cannot infer the driver manoeuvre at a very early stage or before the manoeuvre happens.

2.4.3.4 Deep Learning Methods

Recently, tremendous achievement has made on the deep learning area, dues to the development of deep learning theories, parallel computation hardware, and large-scale

annotated dataset. The deep networks have achieved the state-of-art performance on many computer vision tasks, such as the image classification, segmentation, and object detection domains [95][96]. The deep convolutional neural network (CNN) has been widely used in many intelligent and automated vehicles [97]. Meanwhile, the RNN also achieved significant results on natural language processing and image captioning area [75] [98]. RNN can be used to process the temporal dependence between the dataset as it allows the connection between the previous layers and current layers. To increase the long-term dependency property and overcome the gradient descent, a long short-term memory (LSTM) scheme was proposed [99]. The LSTM largely increases the long-term memory ability of the RNN model and has been successfully applied to many tasks.

As aforementioned, driver intention inference usually require taking the previous driver behaviours and traffic context into consideration. The conventional HMM method has limited ability to capture the long-term dependency. However, at this moment, the RNN can provides a better prediction of the driver intention. In [100], a LSTM-RNN model was designed to infer the driver intention when the vehicle enters an interaction. The RNN outperforms the quadratic discriminate analysis model. Similarly, a series studies have been proposed in [98]. The authors compared the lane-change intention inference performance of the LSTM-based RNN with multiple HMMs. Driver head rotations along with the traffic context from the GPS and digital map were collected. The lane change intent can be detected 3.5 seconds before the vehicle come into another lane with a precision and recall of 90.5% and 87.4% respectively. Based on the review of previous lane change intention inference algorithms, the deep learning algorithms show the significant advantage in the prediction accuracy and larger prediction horizon. Although deep learning algorithms lead to a higher computational burden, these algorithms are much powerful than conventional methods. With the development of hardware platform and software, the deep learning methods can be more distinct in the future.

2.4.4 Evaluation of Driver Intention Inference System

It is important to evaluate the performance of the driver intention classification system. To have a clear perspective about how the classifier work, driver intent classification can be evaluated with the true positive rate (TRP), false positive rate (FRP), and the predicting horizon.

2.4.4.1 Detection Accuracy

True positive rate and false positive rate are two important factors that describe the performance of the classifier and have been used in a large amount of related studies. True positive rate, also called the hit rate or recall, measures how many times the classifier detects the intent successfully, while the false alarm rate describes how many times the classifier miss-classify the intent into wrong category. Sometimes, false positive rate can be more critical than the true positive rate since driver normally do not want to be disrupted by the classifier. If a classifier pursues a high true positive rate at the price of high false positive rate, this system will hardly be accepted. However, if a system has a slightly lower true positive rate and a lower false alarm rate, it is still helpful in some situations. Therefore, the main objective of the classifier is to maximum the true positive rate while minimize the low false positive rate. A continent way to visualize the performance of the classifier is using receiver operating characteristics (ROC). ROC is a graph technique which is used to visualizing, organizing, and selecting classifier based on their performance [101].

2.4.4.2 Prediction Horizon

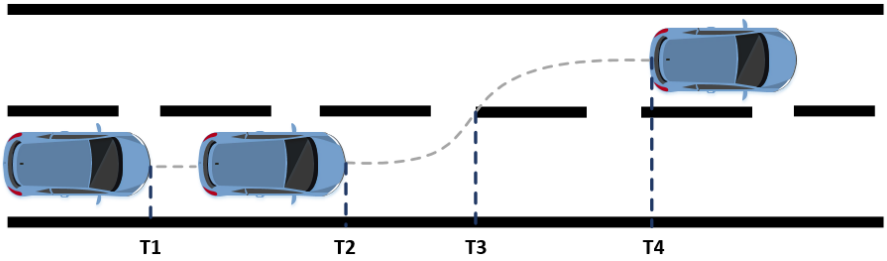

Figure 2-6. Illustration lane change progress.

Despite of the TRP and FRP index, predicting horizon of the intent is another important factor to evaluate the performance of the classifier. Some of the previous studies report a TRP and FRP without giving a clear predicting horizon, which are unfair. As shown in Figure 2-6, there are four critical moments for a lane change process. T1 is the moment when the driver generates the lane change intention. T2 is the moment when the driver finishes traffic context checking and begin to make lane change. T3 represents the moment that vehicle start to cross the lane. At T4, the driver finishes the lane change task. Normally, since there is no precise driver intention model that can explain when the driver generates an intention, T1 is very difficult to be determined. Therefore, most of the studies use T2 and T3 as the time criteria to evaluate the prediction horizon. The earlier the

prediction is made, the difficulty the task will be. After the driver has taken some actions such as steering the wheel and slightly brake the vehicle based on their intent, it will be easier to recognize their intent. However, if the intelligent inference unit tries to recognize the driver intent before some actions are taken, the task will be much difficult since only limited and uncertain information can be obtained. Moreover, the earlier the prediction is made, the higher FRP will be. Therefore, a trade-off between the FRP and the prediction horizon exists, which need to be carefully considered. It is mentioned in [51] that the TPR and FPR can vary with the prediction horizon. These performance index will be less powerful to describe the performance of the classifier without a clear prediction horizon.

Some researchers have paid attention to the prediction horizon issue. Doshi *et al.* found that the data collected 3s prior the manoeuvre was significant enough to present the lane change intention [51]. Salvucci tested their lane change intent inference system on a driving simulator and detected the lane change at the very start with 65% accuracy [34]. As time goes on, the accuracy increased to 80% after half second and 90% after one second. Bernt identified a lane change manoeuvre at a very early stage after it was initialized and achieved 71% prediction accuracy for lane change left and 74% for lane change right in a conditional simulation environment [52]. Bi. *et al.* detected a driver's normal and emergency lane change intent within 0.325s and 0.268s respectively [87]. Kumar realized an average of 1.3s prediction of the intent in advance and can achieve a maximum of 3.29s predicting horizon with a maximum of 82% accuracy [55]. By evaluating the performance of SVM and ANN, Dogan proposed a classifier to predict the lane change 1s to 1.5s prior to the vehicle crossing the lane on a driving simulator [94]. The predicting horizon in the simulation environment is always better than that in the real-world testing. This is mainly due to the large noise and distraction in the real-world environment. However, the performance of the classifier in real-world gives a real indication and is beneficial to the analysis of driver response in real-world.

2.5 Challenges and Future Works

In this section, some future works and challenges that tend to face in the next years are highlighted.

Currently, the study of driver behaviours can be summarized into the following aspects, driver attention, driver intention, driver workload, and driver distraction, etc. For each research area, a vast amount of studies has been proposed. However, it is still unfamiliar

with the relationship between these systems. It is believed that driver behaviours under the distracted condition and the non-distracted condition are different [109] [110]. Meanwhile, if the driver is overloaded after a long drive, the behaviours of surrounding context perception is also different [111]. In terms of the driver intention inference system, how to correctly infer driver intention according to the different mental status need to be clarified. Therefore, the construction of a robust driver intention inference system, which can adapt to different driver status is expected to be studied.

In the real-world, the DII system should not be working as an isolated function. The system need to cooperate with other driver assistance system to make an accurate prediction result. By considering driver distraction and intention has a whole, the control conflicts between the driver and the vehicle can be minimized. Meanwhile, for those partially automated vehicles, the estimation of driver intention will help the decision model to decide whether the drivers are capable to take over the vehicle control authority for some emergency tasks [14][15].

A more challenge work is to obtain a comprehensive understanding about the intention generation process according to the traffic context and human behaviors. Currently, driver attention and workload have been properly modeled. The mathematic models enable the better explanation of driver attention and workload [113][114]. However, there are still limited studies on the explicit modeling of driver intention. A more comprehensive understanding of driver intention will benefit the design of human-like decision making system for the highly automated vehicles.

2.6 Conclusions

Driver intention inference is an important function for ADAS and intelligent vehicles. It is an efficient method to avoid the conflicts between human driver and the intelligent units. Meanwhile, the understanding of human intention enables a better design of the decision-making algorithms for the automated vehicles. The human-like decision making is a major task for current intelligent and automated vehicles. Driver intention can be classified into three levels based on the time constant property. The relationship between level two and level three are clarified in this study. Based on the framework, the traffic context is viewed as the stimuli for the intention, while the driver behavioural information and vehicle dynamics are the response to the stimuli. With respect to the lane change intention inference, multi-modal sensors are used for intention inference. The signal

sources can be classified into three parts as well, which are the traffic context, driver behavioural dynamics, and the vehicle dynamic information. Existing algorithms for intention inference can be summarized into four major groups, which are generative model, discriminative model, cognitive model, and the deep learning models. A comprehensive evaluation method for the intention inference should consider both the accuracy and the prediction horizon. A smart intention inference system should try to predict the driver intention as early as possible. Future works for driver intention inference should concentrate on the precise modelling of intention generation process and its cooperation with other driver status monitoring system.

2.7 Reference

[1] Koesdwiady, Arief, *et al*. "Recent Trends in Driver Safety Monitoring Systems: State of the Art and Challenges." *IEEE Transactions on Vehicular Technology* 66.6 (2017): 4550-4563.

[2] Bellis, Elizabeth, and Jim Page. National motor vehicle crash causation survey (NMVCCS) SAS analytical users manual. No. HS-811 053. 2008.

[3] Martinez, Clara Marina, *et al*. "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey." *IEEE Transactions on Intelligent Transportation Systems* (2017).

[4] Michałek, Maciej Marcin, and Marcin Kiełczewski. "The concept of passive control assistance for docking manoeuvres with n-trailer vehicles." *IEEE/ASME Transactions on Mechatronics* 20.5 (2015): 2075-2084.

[5] Wang, Fei-Yue, and Shu-ming Tang. "Concepts and frameworks of artificial transportation systems." *Complex Systems and Complexity Science* 1.2 (2004): 52-59.

[6] Gaikwad, Vijay, and Shashikant Lokhande. "Lane departure identification for advanced driver assistance." *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015): 910-918.

[7] Son, Young Seop, *et al*. "Robust multirate control scheme with predictive virtual lanes for lane-keeping system of autonomous highway driving." *IEEE Transactions on Vehicular Technology* 64.8 (2015): 3378-3391.

[8] He, Jibo, Jason S. McCarley, and Arthur F. Kramer. "Lane keeping under cognitive load: performance changes and mechanisms." *Human factors* 56.2 (2014): 414-426.

[9] Liebner, Martin, and Felix Klanner. "Driver intent inference and risk assessment." Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort (2014): 1-20.

[10] Bellis, Elizabeth, and Jim Page. National motor vehicle crash causation survey (NMVCCS) SAS analytical users manual. No. HS-811 053. 2008.

[11] Qu, Fengzhong, *et al*. "A security and privacy review of VANETs." *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015): 2985-2996.

[12] Liu, Wei, *et al*. "Parking like a human: a direct trajectory planning solution." *IEEE Transactions on Intelligent Transportation Systems* 18.12 (2017): 3388-3397.

[13] Lv, Chen, *et al*. "Characterization of Driver Neuromuscular Dynamics for Human-Automation Collaboration Design of Automated Vehicles." *IEEE/ASME Transactions on Mechatronics* (2018).

[14] Kim, Hyung Jun, and Ji Hyun Yang. "Takeover requests in simulated partially autonomous vehicles considering human factors." *IEEE Transactions on Human-Machine Systems* 47.5 (2017): 735-740.

[15] Eriksson, Alexander, and Neville A. Stanton. "Takeover time in highly automated vehicles: noncritical transitions to and from manual control." Human factors 59.4 (2017): 689-705.

[16] Carruthers, Peter. "The illusion of conscious will." *Synthese* 159.2 (2007): 197-213.

[17] Ajzen, Icek. "The theory of planned behavior." *Organizational behaviourand human decision processes* 50.2 (1991): 179-211.

[18] Bratman, Michael. "Intention, plans, and practical reason." (1987).

[19] Heinze, Clint. *Modelling intention recognition for intelligent agent systems*. No. DSTO-RR-0286. DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS SCIENCES LAB, 2004.

[20] Tahboub, Karim A. "Intelligent human-machine interaction based on dynamic Bayesian networks probabilistic intention recognition." *Journal of Intelligent and Robotic Systems* 45.1 (2006): 31-52.

[21] Bonchek-Dokow, Elisheva, *Cognitive Modeling of Human Intention Recognition*. Diss. Bar Ilan University, Gonda Multidisciplinary Brain Research Center, 2011.

[22] Bösch, Holger, Fiona Steinkamp, and Emil Boller. "Examining psychokinesis: The interaction of human intention with random number generators--A meta-analysis." *Psychological bulletin* 132.4 (2006): 497.

[23] Chadalavada, Ravi Teja, *et al*. "That's on my mind! robot to human intention communication through on-board projection on shared floor space." *Mobile Robots (ECMR), 2015 European Conference on*. IEEE, 2015.

[24] Huang, Jian, *et al*. "Control of upper-limb power-assist exoskeleton using a human-robot interface based on motion intention recognition." *IEEE transactions on automation science and engineering* 12.4 (2015): 1257-1270.

[25] Han, Ji-Hyeong, Seung-Jae Lee, and Jong-Hwan Kim. "BehaviourHierarchy-Based Affordance Map for Recognition of Human Intention and Its Application to Human–Robot Interaction." *IEEE Transactions on Human-Machine Systems* 46.5 (2016): 708-722.

[26] Jang, Young-Min, *et al*. "Human intention recognition based on eyeball movement pattern and pupil size variation." *Neurocomputing* 128 (2014): 421-432.
[27] Li, Songpo, and Xiaoli Zhang. "Implicit Intention Communication in Human–Robot Interaction Through Visual BehaviourStudies." *IEEE Transactions on Human-Machine Systems* 47.4 (2017): 437-448.
[28] Takeda, Takahiro, Yasuhisa Hirata, and Kazuhiro Kosuge. "Dance step estimation method based on HMM for dance partner robot." *IEEE Transactions on Industrial Electronics* 54.2 (2007): 699-706.
[29] Zhu, Chun, Qi Cheng, and Weihua Sheng. "Human intention recognition in smart assisted living systems using a hierarchical hidden markov model." *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*. IEEE, 2008.
[30] Oliver, Nuria M., Barbara Rosario, and Alex P. Pentland. "A Bayesian computer vision system for modeling human interactions." *IEEE transactions on pattern analysis and machine intelligence* 22.8 (2000): 831-843.
[31] Michon, John A. "A critical view of driver behaviourmodels: what do we know, what should we do?." *Human behaviourand traffic safety*. Springer, Boston, MA, 1985. 485-524.
[32] Doshi, Anup, and Mohan M. Trivedi. "Tactical driver behaviourprediction and intent inference: A review." *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011.
[33] Salvucci, Dario D., and Andrew Liu. "The time course of a lane change: Driver control and eye-movement behavior." *Transportation research part F: traffic psychology and behaviour*5.2 (2002): 123-132.
[34] Salvucci, Dario D. "Modeling driver behaviourin a cognitive architecture." *Human factors* 48.2 (2006): 362-380.
[35] Haufe, Stefan, *et al*. "Electrophysiology-based detection of emergency braking intention in real-world driving." *Journal of neural engineering* 11.5 (2014): 056011.
[36] Kim, Il-Hwa, *et al*. "Detection of braking intention in diverse situations during simulated driving based on EEG feature combination." *Journal of neural engineering* 12.1 (2014): 016001.
[37] Khaliliardali, Zahra, *et al*. "Detection of anticipatory brain potentials during car driving." *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. Ieee, 2012.
[38] McCall, Joel C., and Mohan M. Trivedi. "Driver behaviourand situation aware brake assistance for intelligent vehicles." *Proceedings of the IEEE* 95.2 (2007): 374-387.
[39] Tran, Cuong, Anup Doshi, and Mohan Manubhai Trivedi. "Modeling and prediction of driver behaviourby foot gesture analysis." *Computer Vision and Image Understanding* 116.3 (2012): 435-445.
[40] Ohn-Bar, Eshed, *et al*. "On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems." *Computer Vision and Image Understanding* 134 (2015): 130-140.
[41] Mabuchi, Ryuki, and Keiichi Yamada. "Study on driver-intent estimation at yellow traffic signal by using driving simulator." *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011.
[42] Takahashi, Hiroshi, and Kouichi Kuroda. "A study on mental model for inferring driver's intention." *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*. Vol. 2. IEEE, 1996.
[43] Kumagai, Toru, *et al*. "Prediction of driving behaviourthrough probabilistic inference." *Proc. 8th Intl. Conf. Engineering Applications of Neural Networks*. 2003.
[44] Oliver, Nuria, and Alex P. Pentland. "Graphical models for driver behaviourrecognition in a smartcar." *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*. IEEE, 2000.
[45] Liebner, Martin, *et al*. "Driver intent inference at urban intersections using the intelligent driver model." *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012.
[46] Liu, Andrew, and Alex Pentland. "Towards real-time recognition of driver intentions." *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*. IEEE, 1997.
[47] Imamura, Takashi, *et al*. "Estimation for driver's intentions in straight road environment using hidden markov models." *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*. IEEE, 2010.
[48] Beauchemin, Steven S., *et al*. "Portable and scalable vision-based vehicular instrumentation for the analysis of driver intentionality." *IEEE Transactions on Instrumentation and Measurement* 61.2 (2012): 391-401.
[49] Vahidi, Ardalan, and Azim Eskandarian. "Research advances in intelligent collision avoidance and adaptive cruise control." *IEEE transactions on intelligent transportation systems* 4.3 (2003): 143-153.
[50] Morris, Brendan, Anup Doshi, and Mohan Trivedi. "Lane change intent prediction for driver assistance: On-road design and evaluation." *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011.
[51] Doshi, Anup, Brendan Morris, and Mohan Trivedi. "On-road prediction of driver's intent with multimodal sensory cues." *IEEE Pervasive Computing* 10.3 (2011): 22-34.
[52] Berndt, Holger, Jorg Emmert, and Klaus Dietmayer. "Continuous driver intention recognition with hidden markov models." *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008.
[53] Toledo-Moreo, Rafael, and Miguel A. Zamora-Izquierdo. "IMM-based lane-change prediction in highways with low-cost GPS/INS." *IEEE Transactions on Intelligent Transportation Systems* 10.1 (2009): 180-185.
[54] McCall, Joel C., *et al*. "A collaborative approach for human-centered driver assistance systems." *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004.
[55] Kumar, Puneet, *et al*. "Learning-based approach for online lane change intention prediction." *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013.
[56] Kumagai, Toru, *et al*. "Prediction of driving behaviourthrough probabilistic inference." *Proc. 8th Intl. Conf. Engineering Applications of Neural Networks*. 2003.
[57] Schmidt, Kim, *et al*. "A mathematical model for predicting lane changes using the steering wheel angle." *Journal of safety research* 49 (2014): 85-e1.
[58] Li, Fang, *et al*. "Driving intention inference based on dynamic Bayesian networks." *Practical Applications of Intelligent Systems*. Springer, Berlin, Heidelberg, 2014. 1109-1119.
[59] Hou, Haijing, *et al*. "Driver intention recognition method using continuous hidden markov model." *International Journal of Computational Intelligence Systems* 4.3 (2011): 386-393.
[60] Lethaus, Firas, *et al*. "Using pattern recognition to predict driver intent." *International Conference on Adaptive and Natural Computing Algorithms*. Springer, Berlin, Heidelberg, 2011.
[61] Jang, Young-Min, Rammohan Mallipeddi, and Minho Lee. "Identification of human implicit visual search intention based on eye movement and pupillary analysis." *User Modeling and User-Adapted Interaction* 24.4 (2014): 315-344.

[62] Pech, Timo, Philipp Lindner, and Gerd Wanielik. "Head tracking based glance area estimation for driver behaviourmodelling during lane change execution." *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014.
[63] Shinar, David. "Looks are (almost) everything: where drivers look to get information." *Human Factors* 50.3 (2008): 380-384.
[64] Caceres, N., J. P. Wideberg, and F. G. Benitez. "Deriving origin–destination data from a mobile phone network." *IET Intelligent Transport Systems* 1.1 (2007): 15-26.
[65] Borji, Ali, Andreas Lennartz, and Marc Pomplun. "What do eyes reveal about the mind?: Algorithmic inference of search targets from fixations." *Neurocomputing* 149 (2015): 788-799.
[66] Doshi, Anup, and Mohan Trivedi. "Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions." *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE, 2009.
[67] Zhou, Huiping, Makoto Itoh, and Toshiyuki Inagaki. "Toward inference of driver's lane-change intent under cognitive distraction." *SICE Annual Conference 2010, Proceedings of*. IEEE, 2010.
[68] Doshi, Anup, and Mohan Manubhai Trivedi. "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes." *IEEE Transactions on Intelligent Transportation Systems* 10.3 (2009): 453-462.
[69] Lethaus, Firas, *et al*. "A comparison of selected simple supervised learning algorithms to predict driver intent based on gaze data." *Neurocomputing* 121 (2013): 108-130.
[70] Timm, Fabian, and Erhardt Barth. "Accurate Eye Centre Localisation by Means of Gradients." *Visapp* 11 (2011): 125-130.
[71] Wang, Shuo, *et al*. "Atypical visual saliency in autism spectrum disorder quantified through model-based eye tracking." *Neuron* 88.3 (2015): 604-616.
[72] Martin, Sujitha, *et al*. "Dynamics of Driver's Gaze: Explorations in BehaviourModeling and Manoeuvre Prediction." *IEEE Transactions on Intelligent Vehicles* (2018).
[73] Lethaus, Firas, *et al*. "Windows of driver gaze data: how early and how much for robust predictions of driver intent?." *International Conference on Adaptive and Natural Computing Algorithms*. Springer, Berlin, Heidelberg, 2013.
[74] Ohn-Bar, Eshed, and Mohan Manubhai Trivedi. "Looking at humans in the age of self-driving and highly automated vehicles." *IEEE Transactions on Intelligent Vehicles* 1.1 (2016): 90-104.
[75] Jain, Ashesh, *et al*. "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture." *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016.
[76] Murphy-Chutorian, Erik, and Mohan Manubhai Trivedi. "Head pose estimation in computer vision: A survey." *IEEE transactions on pattern analysis and machine intelligence* 31.4 (2009): 607-626.
[77] Ahlstrom, Christer, *et al*. "Processing of eye/head-tracking data in large-scale naturalistic driving data sets." *IEEE transactions on intelligent transportation systems* 13.2 (2012): 553-564.
[78] Tawari, Ashish, Sujitha Martin, and Mohan Manubhai Trivedi. "Continuous head movement estimator for driver assistance: Issues, algorithms, and on-road evaluations." *IEEE Transactions on Intelligent Transportation Systems* 15.2 (2014): 818-830.
[79] Tran, Cuong, Anup Doshi, and Mohan Manubhai Trivedi. "Modeling and prediction of driver behaviourby foot gesture analysis." *Computer Vision and Image Understanding* 116.3 (2012): 435-445.
[80] Tiwari, Ravikumar K., and S. D. Giripunje. "Design approach for EEG-based human computer interaction driver monitoring system." *Int J Latest Trends Eng Technol IJLTET* 3.4 (2014): 250-255.
[81] Das, Nikhil, Eshed Ohn-Bar, and Mohan M. Trivedi. "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics." *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015.
[82] Xing, Yang, *et al*. "Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition." *IEEE Transactions on Computational Social Systems* 5.1 (2018): 95-108.
[83] Nicolas-Alonso, Luis Fernando, and Jaime Gomez-Gil. "Brain computer interfaces, a review." *Sensors* 12.2 (2012): 1211-1279.
[84] Wang, Xiao-Wei, Dan Nie, and Bao-Liang Lu. "Emotional state classification from EEG data using machine learning approach." *Neurocomputing* 129 (2014): 94-106.
[85] Haufe, Stefan, *et al*. "EEG potentials predict upcoming emergency brakings during simulated driving." *Journal of neural engineering* 8.5 (2011): 056001.
[86] Ikenishi, Toshihito, Takayoshi Kamada, and Masao Nagai. "Classification of Driver Steering Intentions Using an Electroencephalogram." *Journal of System Design and Dynamics* 2.6 (2008): 1274-1283.
[87] Bi, Luzheng, *et al*. "Detecting driver normal and emergency lane-changing intentions with queuing network-based driver models." *International Journal of Human-Computer Interaction* 31.2 (2015): 139-145.
[88] Driggs-Campbell, Katherine, and Ruzena Bajcsy. "Identifying modes of intent from driver behaviors in dynamic environments." *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015.
[89] Ding, Jieyun, *et al*. "Driver intention recognition method based on comprehensive lane-change environment assessment." *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014.
[90] Li, Keqiang, *et al*. "Lane changing intention recognition based on speech recognition models." *Transportation research part C: emerging technologies* 69 (2016): 497-514.
[91] Polling, D., *et al*. "Inferring the driver's lane change intention using context-based dynamic Bayesian networks." *Systems, Man and Cybernetics, 2005 IEEE International Conference on*. Vol. 1. IEEE, 2005.
[92] Kasper, Dietmar, *et al*. "Object-oriented Bayesian networks for detection of lane change manoeuvres." *IEEE Intelligent Transportation Systems Magazine* 4.3 (2012): 19-31.
[93] Mandalia, Hiren M., and Mandalia Dario D. Salvucci. "Using support vector machines for lane-change detection." *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 49. No. 22. Sage CA: Los Angeles, CA: SAGE Publications, 2005.
[94] Dogan, Ueruen, Hannes Edelbrunner, and Ioannis Iossifidis. "Towards a driver model: Preliminary study of lane change behavior." *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008.
[95] Girshick, Ross. "Fast r-cnn." *arXiv preprint arXiv:1504.08083* (2015).
[96] Lv, Yisheng, *et al*. "Traffic flow prediction with big data: a deep learning approach." *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015): 865-873.
[97] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.
[98] Jain, Ashesh, *et al*. "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture." *arXiv preprint arXiv:1601.00740* (2016).

[99] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
[100] Zyner, Alex, Stewart Worrall, and Eduardo Nebot. "A Recurrent Neural Network Solution for Predicting Driver Intention at Unsignalized Intersections." *IEEE Robotics and Automation Letters* (2018).
[101] Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27.8 (2006): 861-874.
[102] Imamura, Takashi, *et al*. "Real-time implementation of estimation method for driver's intention on a driving simulator." *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014.
[103] McCall, Joel C., *et al*. "Lane change intent analysis using robust operators and sparse bayesian learning." *IEEE Transactions on Intelligent Transportation Systems* 8.3 (2007): 431-440.
[104] Jang, Young-Min, Rammohan Mallipeddi, and Minho Lee. "Driver's lane-change intent identification based on pupillary variation." *Consumer Electronics (ICCE), 2014 IEEE International Conference on*. IEEE, 2014.
[105] Zhou, Huiping, Makoto Itoh, and Toshiyuki Inagaki. "Eye movement-based inference of truck driver's intent of changing lanes." *SICE Journal of Control, Measurement, and System Integration* 2.5 (2009): 291-298.
[106] Henning, Matthias J., *et al*. "Modelling driver behaviourin order to infer the intention to change lanes." *Proceedings of European Conference on Human Centred Design for Intelligent Transport Systems*. Vol. 113. 2008.
[107] Ohn-Bar, Eshed, *et al*. "Predicting driver manoeuvres by learning holistic features." *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014.
[108] Xu, Guoqing, Li Liu, and Zhangjun Song. "Driver behaviouranalysis based on bayesian network and multiple classifiers." *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*. Vol. 3. IEEE, 2010.
[109] Young, Kristie, Michael Regan, and M. Hammer. "Driver distraction: A review of the literature." *Distracted driving* (2007): 379-405.
[110] Liao, Yuan, *et al*. "Detection of driver cognitive distraction: A comparison study of stop-controlled intersection and speed-limited highway." *IEEE Transactions on Intelligent Transportation Systems* 17.6 (2016): 1628-1637.
[111] Healey, Jennifer A., and Rosalind W. Picard. "Detecting stress during real-world driving tasks using physiological sensors." *IEEE Transactions on intelligent transportation systems* 6.2 (2005): 156-166.
[112] Klauer, Sheila G., *et al*. "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data." (2006).
[113] Regan, Michael A., Charlene Hallett, and Craig P. Gordon. "Driver distraction and driver inattention: Definition, relationship and taxonomy." *Accident Analysis & Prevention* 43.5 (2011): 1771-1781.
[114] Ma, Yu-Fei, *et al*. "A generic framework of user attention model and its application in video summarization." *IEEE transactions on multimedia* 7.5 (2005): 907-919.

PART III:

TRAFFIC CONTEXT PERCEPTION.

Integrated lane detection systems

3

PAPER II – Survey to Lane Detection Systems Integration and Evaluation

Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision

Authors:
Yang Xing, Chen Lv, Long Chen, Huaji Wang, Hong Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang

This paper has been published by
IEEE/ACC Journal of Automatic Sinica (2018)

Abstract

Lane detection is a fundamental aspect of most current ADAS. A large volume of existing studies focuses on the study of vision-based lane detection methods due to the extensive knowledge background and the low-cost of camera devices. In this study, previous vision-based lane detection studies are reviewed in terms of three aspects, which are lane detection algorithms, integration, and evaluation methods. Next, considering the inevitable limitations that exist in the camera-based lane detection system, the system integration methodologies for constructing more robust detection system are reviewed and analysed. The integration methods are further divided into three levels namely algorithm, system, and sensor level. Algorithm level combines different lane detection algorithms while system level integrates other object detection systems to comprehensively detect lane positions. Sensor level integration uses multi-modal sensors to build a robust lane recognition system. Regarding the complexity of evaluating the detection system, and the lack of common evaluation procedure and uniform metrics in past studies, existing evaluation methods and metrics are analysed and classified to propose a better evaluation of the lane detection system. Next, a comparison of representative studies is performed. Finally, a discussion on the limitations of current lane detection systems and the future developing trends toward an Artificial Society, Computational experiment-based parallel (ACP) lane detection framework is proposed.

Index Terms - Advanced driver assistance Systems (ADAS), ACP theory, benchmark, Lane detection, parallel vision, performance evaluation

3.1 Introduction

3.1.1 Background

Traffic accidents are mainly caused by human mistakes such as inattention, misbehaviour, and distraction [1]. Many companies and institutes have proposed methods and techniques for the improvement of driving safety and reduction of traffic accidents. Among these techniques, road perception and lane marking detection play a vital role in helping drivers avoid mistakes. The lane detection is the foundation of many ADAS such as LDW and LKA [2] [3]. Some successful ADAS or automotive enterprises, such as Mobileye, BMW, and Tesla, etc. have developed their own lane detection and lane keeping products and have obtained significant achievement both in research and real-world application. Either of the automotive enterprises or the personal customers have accepted the Mobileye Series ADAS products and Tesla Autopilot for self-driving. Almost all the current mature lane assistance products use vision-based techniques since the lane markings are painted on the road for human visual perception. The utilization of vision-based techniques detects lanes from the camera devices and prevent the driver from making unintended lane changes. Therefore, the accuracy and robustness are the two most important properties for lane detection systems. Lane detection systems should have the capability to be aware of unreasonable detections and adjust the detection and tracking algorithm accordingly [4] [5]. When a false alarm occurs, the ADAS should alert the driver to concentrate on the driving task. On the other hand, vehicles with high levels of automation continuously monitor their environment and should be able to deal with low-accuracy detection problems by themselves. Hence, evaluation of lane detection systems becomes even more critical with increasing automation of vehicles.

Most vision-based lane detection systems are commonly designed based on image processing techniques within similar frameworks. With the development of high-speed computing devices and advanced machine learning theories such as deep learning, lane detection problems can be solved in a more efficient fashion using an end-to-end detection procedure. However, the critical challenge faced by lane detection systems is the demand for high reliability and the diverse working conditions. One efficient way to construct robust and accurate advanced lane detection systems is to fuse multi-modal sensors and integrate lane detection systems with other object detection systems, such as detection by surrounding vehicles and road area recognition. It has been proved that lane detection

performance can be improved with these multi-level integration techniques [4]. However, the highly accurate sensors such as light/laser detection and ranging (LIDAR/LADAR) are expensive and not available in public transport.

3.1.2 Contribution

In this study, the literature reviews on the lane detection algorithms, the integration methods, and evaluation methods are provided. The contribution of this paper can be summarized as follows.

A considerable number of existing studies does not provide enough information on the integration methodologies of lane detection systems and other systems or sensors. Therefore, in this study, the integration methodologies are analysed in detail and the ways of integration are categorized into three levels: sensor level, system level, and algorithm level.

Dues to the lack of ground truth data and uniform metrics, the evaluation of the lane detection system remains a challenge. Since various lane detection systems differ with respect to the hardware and software they use, it is difficult to undertake a comprehensive comparison and evaluation of these systems. In this study, previous evaluation methods are reviewed and classified into offline methods, which use still images and videos, and online methods, which are based on real time confidence calculation.

Finally, a novel lane detection system design framework based on ACP parallel theory is introduced towards a more efficient way to deal with the training and evaluation of lane detection models. ACP is short for Artificial Society, Computational experiments, and Parallel execution, which are the three major components of parallel system. The ACP-based lane detection parallel system aims to construct virtual parallel scenarios for model training and benefit the corresponding real-world system. The construction method for the lane detection parallel vision system will be analysed.

3.2 Vision-based Lane Detection Algorithm Review

Literature reviews of lane detection algorithms and their corresponding general frameworks have been proposed in [4]-[6]. Hillel *et al*. [4] concluded that road colour, texture, boundaries, and lane markings are the main perception aspects for human drivers. In [5], McCall and Trivedi classified the lane detection objectives into three categories, which are lane departure warning, driver attention awareness, and automated vehicle

control system design. However, these papers paid more attention to the design of lane detection algorithms and incompletely reviewed the integration and evaluation methods. This study tries to comprehensively review the lane detection system from the perspective of algorithms, integration method, and evaluation method. Firstly, in this section, lane detection algorithms and techniques are reviewed from the scope of conventional image processing and novel machine learning methods. In the first part of this section, basic lane detection procedure and general framework will be analysed. The second part will concentrate on the review of commonly used conventional image processing methods. In the last part, lane detection algorithms based on machine learning and deep learning methods, especially the utilization of convolutional neural network (CNN) will be discussed.

3.2.1 General Lane Detection Procedure

Vision-based lane detection systems described in studies usually consist of three main procedures, which are image pre-processing, lane detection and lane tracking. Among these, the lane detection process, which comprises feature extraction and model fitting, is the most important aspect of the lane detection system, as shown in Figure 3-1. The most common procedures in the pre-processing step includes region of interest (ROI) selection, vanishing point detection, transferring colour image into greyscale image or a different colour format, noise removal and blur, inverse perspective mapping (IPM), also known as birds-eye view, segmentation, and edge statistics, etc. Among these tasks, determining the ROI is usually the first step performed in most of the previous researches. The main reason for focusing on ROI is to increase computation efficiency and reduce false lane detection. ROI can be roughly selected as the lower portion of an input image or dynamically determined according to the detected lanes. It can also be more efficiently determined with prior road area detection [7] [8]. Details of these methods are described in the next section. Generally speaking, a carefully-designed ROI will significantly improve lane detection accuracy as well as computation efficiency.

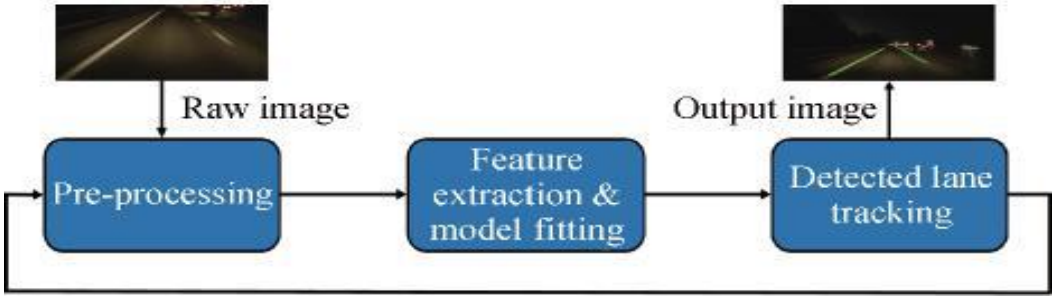

Figure 3-1. General architecture of lane detection system.

Once the input images have been pre-processed, lane features such as the colours and edge features can be extracted, hence can be detected based on these features. The Hough Transform algorithm, which uses the edge pixel images, is one of the most widely used algorithms for lane detection in previous studies. However, this method is designed to detect straight lines in the beginning and is not efficient in curve lane detection. Curve lanes can often be detected based on model fitting techniques such as random sample consensus (RANSAC). RANSAC fits lane models by recursively testing the model fitting score to find the optimal model parameters. Therefore, it has a strong ability to cope with outlier features. Finally, after lanes have been successfully detected, lane positions can be tracked with tracking algorithms such as Kalman filter or particle filters to refine the detection results and predict lane positions in a more efficient way.

3.2.2 Conventional Image-Processing-Based Lane Detection Algorithms

Vision-based lane detection can be roughly classified into two categories: feature-based [9]−[19] and model-based [20]−[29]. Feature-based methods rely on the detection of lane marking features such as lane colours, textures, and edges. For example, in [9], noisy lane edge features were detected using the Sobel operator and the road images were divided into multiple sub regions along the vertical direction. Suddamalla *et al* detected the curves and straight lanes using pixel intensity and edge information with lane markings being extracted with adaptive threshold techniques [10]. To remove camera perspective distortions from the digital images and extract real lane features, lane markings can be efficiently detected with a perspective transform. Collado *et al*. [11] created a bird-view of the road image and proposed an adaptive lane detection and classification method based on spatial lane features and the Hough Transform (HT) algorithm. A combination of IPM and clustered particle filters method based on lane features was used to estimate multiple lanes in [12]. The authors claimed that it is less

robust if a strong lane model is used in the context and they only used a weak model for particle filter tracking. Instead of using colour images, lanes can also be detected using other colour format images. The general idea behind the colour format transform is that the yellow and white lane markings can be more distinct in other colour domain, so the contrast ratio is increased. In [13], lane edges were detected with an extended edge linking algorithm in the lane hypothesis stage. Lane pixels in the YUV format, edge orientation, and width of lane markings were used to select the candidate edge-link pairs in the lane verification step. In [14], lanes were recognized using an unsupervised and adaptive classifier. Colour images were first converted to HSV format to increase the contrast. Then, the binary feature image was processed using the threshold method based on the brightness values. Although in some normal cases the colour transform can benefit the lane detection, it is not robust and has limited ability to deal with shadows and illumination variation [4].

Borkar *et al*. [15] proposed a layered approach to detect lanes at night. A temporal blur technique was used to reduce video noise and binary images were generated based on an adaptive local threshold method. The lane finding in another domain algorithm (LANA) represented lane features in the frequency domain [16]. The algorithm captured the lane strength and orientation in the frequency domain and a deformable template was used to detect the lane markings. Results showed that LANA was robust under varying conditions. In [17], a spatiotemporal lane detection algorithm was introduced. A series of spatiotemporal images were generated by accumulating certain row pixels from the past frames and the lanes were detected using Hough Transform applied on the synthesized images. In [18], a real-time lane detection system based on FPGA and DSP was designed based on lane gradient amplitude features and an improved Hough Transform. Ozgunalp and Dahnoun proposed an improved feature map for lane detection [19]. The lane orientation histogram was first determined with edge orientations and then the feature map was improved and shifted based on the estimated lane orientation.

In general, feature-based methods have better computational efficiency and are able to accurately detect lanes when the lane markings are clear. However, due to too many constraints are assumed, such as the lane colours and shapes, the drawbacks of these methods include less robustness to deal with shadows and poor visibility condition compared to model-based methods.

Model-based methods usually assume that lanes can be described with a specific model such as a linear model, a parabolic model, or various kinds of spline models. Besides, some assumptions about the road and lanes, such as a flat ground plane, are required. Among these models, spline models were popular in previous studies since these models are flexible enough to recover any shapes of the curve lanes. Wang *et al*. fitted lanes with different spline models [20], [21]. In [20], a Catmull-Rom spline was used to model the lanes in the image. In [21], the lane model was improved to generate a B-snake model, which can model any arbitrary shape by changing the control points. In [22], a novel parallel-snake model was introduced. In [23], lane boundaries were detected based on a combination of Hough Transform in near-field areas and a river-flow method in farfield areas. Finally, lanes were modelled with a B-spline model and tracked with a Kalman filter. Jung and Kelber described the lanes with a linear-parabolic model and classified the lane types based on the estimated lane geometries [24]. Aly proposed a multiple lane fitting method based on the integration of Hough Transform, RANSAC, and B-spline model [25]. Initial lane positions were first roughly detected with Hough Transform and then improved with RANSAC and B-spline model. Moreover, a manually labelled lane dataset called the Caltech Lane dataset was introduced.

The RANSAC algorithm is the most popular way to iteratively estimate the lane model parameters. In [26], linear lane model and RANSAC were used to detect lanes, and a Kalman filter was used to refine the noisy output. Ridge features and adapted RANSAC for both straight and curve lane fitting were proposed in [27], [28]. The ridge features of lane pixels, which depend on the local structures rather than contrast, were defined as the centre lines of a bright structure of a region in a greyscale image. In [29], [30], hyperbolic model and RANSAC were used for lane fitting. In [30], input images were divided into two parts known as far-field area and near-field area. In near-field area, lanes were regarded as straight lines detected using the Hough Transform algorithm. In far-field area, lanes were assumed to be curved lines and fitted using hyperbolic model and RANSAC.

In [31], a conditional random field method was proposed to detect lane marks in urban areas. Bounini *et al*. introduced a lane boundary detection method for autonomous vehicle working in a simulation environment [32]. A least-square method was used to fit the line model and the computation cost was reduced by determining a dynamic ROI. In [33], an automated multi-segment lane-switch scheme and a RANSAC lane fitting method were

proposed. RANSAC algorithm was applied to fit the lines based on the edge image. A lane-switch scheme was used to determine lane curvatures and choose the correct lane models from straight and curve models to fit the lanes. In [34], a Gabor wavelet filter was applied to estimate the orientation of each pixel and match a second-order geometric lane model. Niu *et al*. [35] proposed a novel curve fitting algorithm for lane detection with a two-stage feature extraction algorithm (LDTFE). A density based spatial clustering of application with noise (DBSCAN) algorithm was applied to determine whether the candidate lane line segments belong to ego lanes or not. The identified small lane line segments can be fitted with curve model and this method is particularly efficient for small lane segments detection tasks.

Model-based methods are more robust than feature-based methods because of the use of model-fitting techniques. The noisy measurement and the outlier pixels of lane markings usually can be ignored with the model. However, model-based methods usually entail more computational cost since RANSAC has no upper limits on the number of iterations. Moreover, model-based methods are less easy to be implemented compared to the feature-based systems.

3.2.3 Machine Learning-Based Lane Detection Algorithms

Despite using conventional image processing-based methods to detect lane markings, some researchers focus on detecting lane marking using novel machine learning and deep learning methods. Deep learning techniques have been one of the hottest research areas in the past decade due to the development of deep network theories, parallel computing techniques, and large-scale data. Many deep learning algorithms show great advantages in computer vision tasks and the detection and recognition performance increases dramatically compared to conventional approaches. The CNN is one of the most popular approaches used for object recognition research. CNN provides some impressive properties such as high detection accuracy, automatic feature learning, and end-to-end recognition. Recently, some researchers have successfully applied CNN and other deep learning techniques for lane detection. It is reported that by using CNN model, the lane detection accuracy increased dramatically from 80% to 90% compared with traditional image processing methods [36].

Li *et al.* [37] proposed a lane detection system based on deep CNN and RNN. A CNN was fed with a small ROI image that was used for multiple tasks. There are two types of

CNN outputs. The first is a discrete classification result indicating if the visual cues are lane markers or not. If a lane was detected, then the other output would be the continuous estimation of lane orientation and location. To recognize the global lane structures in a video sequence instead of local lane positions in a single image, RNN was used to recognize the lane structures in sequence data with its internal memory scheme. Training was based on a merged scene with three cameras facing front, left side and rear area, respectively. Accurate detection results showed that the integrated lane detection method using CNN and RNN can work in practice. Besides, RNN can help recognize and connect lanes that are covered by vehicles or obstacles.

Gurghian [38] proposed another deep CNN method for lane marking detection using two side-facing cameras. The proposed CNN recognized the side lane positions with an end-to-end detection process. The CNN was trained with both real-world images and synthesized images and achieved a 99% high detection accuracy. To solve the low accuracy and high computational cost problem, authors in [36] proposed a novel lane marking detection method based on a point cloud map generated by a laser scanner. To improve the robustness and accuracy of the CNN result, a gradual up-sampling method was introduced. The output image was in the same format as the input images to get an accurate classification result. The reported computation cost of each algorithm is 28.8 s on average, which can be used for offline high-precision road map construction.

In [39], a spiking neural network was used to extract edge images and lanes were detected based on Hough Transform. This was inspired by the idea that a human neuron system produces a dense pulse response to edges while generating a sparse pulse signal to flat inputs. A similar approach can be found in [40]. The study proposed a lane detection method based on RANSAC and CNN. One eight-layer CNN including three convolution layers was used to remove the noise in edge pixels if the input images were too complex. Otherwise, RANSAC was applied to the edge image directly to fit the lane model. He *et al*. proposed a dual-view CNN for lane detection [41]. Two different views, which were the front view and top view of the road obtained from the same camera, were fed into the pre-trained CNN simultaneously The CNN contained two sub-CNN networks to process the two kinds of input images separately and concatenate the results eventually. Finally, an optimal global strategy taking into account lane length, width, and orientations was used to threshold the final lane markings.

Instead of using general image processing and machine learning methods, some other researchers also used evolution algorithms or heuristic algorithms to automatically search lane boundaries. For example, Revilloud proposed a novel lane detection method using a confidence map and a multi-agent model inspired by human driver behaviour [42]. Similarly, an ant colonies evolution algorithm for optimal lane marking search was proposed in [43]. A novel multiple-lanes detection method using directional random walking was introduced in [44]. In that study, a morphology-based approach was used to extract lane mark features at the beginning. Then, the directional random walk based on a Markov probability matrix was applied to link candidate lane features. The proposed algorithm required no assumption about the road curvatures or lane shapes.

In summary, it can be stated that machine learning algorithms or intelligent algorithms increase the lane detection accuracy significantly and provide many efficient detection architectures and techniques. Although these systems usually require more computational cost and need large amount of training data, these systems are more powerful than conventional methods. Therefore, many novel efficient and robust lane detection methods with lower training and computation requirements are expected to be developed soon.

3.3 Integration Methodologies for Vision-Based Lane Detection Systems

3.3.1 Integration Methods Introduction

Although many studies have been done to enable accurate vision-based lane detection, the robustness of the detection systems still cannot meet the real-world requirements, especially in urban areas, due to the highly random properties of the traffic and the state of roads. Therefore, a reasonable way to enhance the lane detection system is to introduce redundancy algorithms, integrate with other object detection systems or use sensor fusion methods. It is a common agreement among automotive industries that a single sensor is not enough for vehicle perception tasks. Some companies such as Tesla, Mobileye, and Delphi developed their own intelligent on-vehicle perception system using multiple sensors like cameras, and radar (especially the millimetre-wave radar). In this section, the integration methods will be classified into three levels, which are algorithm level, system level, and sensor level, as shown in Figure 3-2.

Specifically, algorithm level integration combines different lane detection algorithms together to comprehensively determine reasonable lane positions and improve the

robustness of the system. In system level integration, different object detection systems work simultaneously with real-time communication with each other. Finally, in sensor level integration, multi-modal sensors are integrated. The proposed sensor fusion methods in this level are believed to improve the robustness of the lane detection system most significantly. In the following sub-sections, the multi-level integration techniques will be described in detail and the studies conducted within each scope will be discussed.

3.3.2 Algorithm Level Integration

Integration of vision-based lane detection algorithms has been widely used in the past. Past studies have focused on two main integration architectures, which can be summarized as parallel and serial combination methods. Moreover, feature-based and model-based algorithms can also be combined. Serial combination methods are commonly seen in previous studies. Studies described in [20], [21], [25] demonstrate examples of methods that serially combine the Hough Transform, RANSAC, and spline model fitting methods. Another method followed in multiple studies involves applying a lane tracking system after the lane detection procedure to refine and improve the stability of the detected lanes [5], [21], [22], [45] − [47]. For lane tracking, Kalman filter and particle filter are the two most widely used tracking algorithms [4]. Shin proposed a super-particle filter combining two separate particle filters for ego lane boundary tracking [48]. In [49], a learning-based lane detection method is proposed and tracked with a particle filter. The learning-based algorithm requires no prior road model and vehicle velocity knowledge.

Parallel combination methods can be found in [50], [51]. In [50], a monocular vision-based lane detection system is combined with two independent algorithms in parallel to make a comprehensive judgement. The first algorithm used a lane marking extractor and road shape estimation to find potential lanes. Meanwhile, a simple feature-based detection algorithm was applied to check the candidate lanes chosen by the first algorithm. If the results from the two algorithms are comparable with each other, the detection result is accepted. Douret *et al*. proposed three parallel integrated algorithms to pursue a robust lane detection with higher confidence [51]. Two lower level lane detection algorithms, namely, lateral and longitudinal consistent detection methods, were processed simultaneously. Then, the sampling points of the detected lanes given by these two lower level detection algorithms were tested. If the results were close to each other, the detection

was viewed as a success and the average position from the two algorithms was selected as the lane position.

Some studies also combined different lane features to construct a more accurate feature vector for lane detection. In [52], the lane detection system was based on the fusion of colour and edge features. Colour features were used to separate road foreground and background regions using Otsus method, while edges were detected with a canny detector. Finally, curve lanes in the image were fitted using Lagrange interpolating polynomial. In [53], a three-feature based automatic lane detection algorithm (TFALDA) was proposed. Lane boundary was represented as a three-feature vector, which includes intensity, position, and orientation value of the lane pixels. The continuity of lanes was used as the selection criteria to choose the best current lane vector that is at the minimum distance with the previous one.

Although parallel integration methods improve the robustness of the system by introducing redundancy algorithms, the computation burden will increase correspondingly. Therefore, a more efficient way is to combine algorithms in a dynamic manner and only initiate a redundancy system when it is necessary.

3.3.3 System Level Integration

Lane detection in the real world can be affected by surrounding vehicles and other obstacles, which may have similar colour or texture features to the lane markings in the digital images. For instance, the guardrail usually shows strong lane-like characteristics in colour images and can easily cause false lane detection [54]−[56]. Therefore, integrating the lane detection system with other on-board detection systems will enhance the accuracy of the lane detection system. Obstacle detection and road painting detection are the two basic categories of vision-based detection techniques, as shown in Figure 3-2. By introducing an obstacle, noise measurement or outlier pixels can be filtered. Similarly, road recognition can narrow down the searching area for lane detections and lead to a reasonable result.

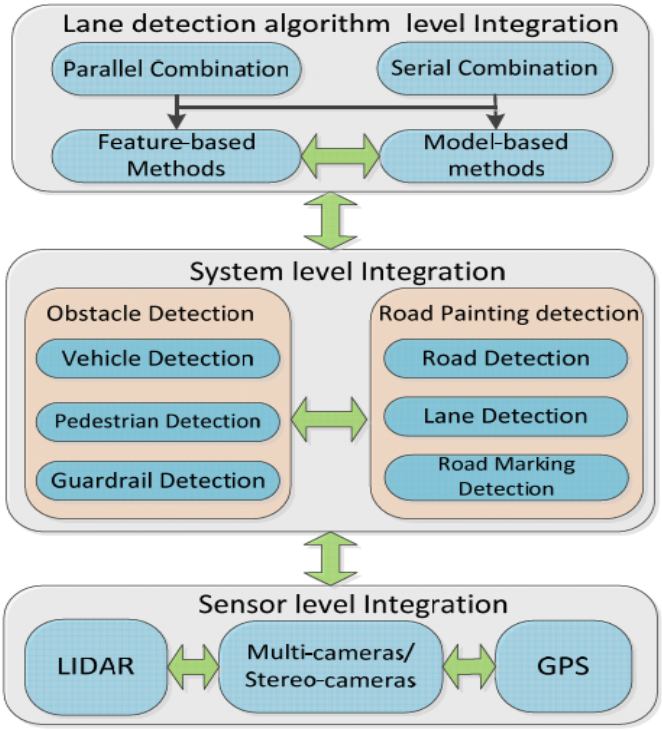

Figure 3-2. Diagram for Lane Detection Integration Level.

Lane detection algorithms usually require lane features for model fitting tasks. Nearby vehicles, especially passing vehicles are likely to cause a false detection result due to occlusion and similar factors. With the detection of surrounding vehicles, the colour, shadow, appearance, and the noise generated by the vehicles ahead can be removed and a higher accuracy of lane boundaries can be achieved [30]. In [30], [57] − [60], the lane detection result was reported to be more accurate with a front-vehicle detection system. This reduces the quantities of false-lane features and improves the model fitting accuracy. Cheng *et al*. proposed an integrated lane and vehicle detection system. Lane markings were detected by analysing road and lane colour features, and the system was designed so as not to be influenced by variations in illumination [57]. Those vehicles that have similar colours with the lanes were distinguished on the basis of the size, shape, and motion information.

Sivaraman and Trivedi [58] proposed a driver assistance system based on an integration of lane and vehicle tracking systems. With the tracking of nearby vehicles, the position of surrounding vehicles within the detected lanes and their lane changes behaviours can be recognized. Final evaluation results showed an impressive improvement compared to the results delivered by the single lane detection algorithm. In [61], a novel lane and

vehicle detection integration method called efficient lane and vehicle detection with integrated synergies (ELVIS) was proposed. The integration of vehicles and lane detection reduces the computation cost of finding the true lane positions by at least 35%. Similar results can be found in [62]. An integrated lane detection and front vehicle recognition algorithm for a forward collision warning system was also introduced. Front vehicles were recognized with a Hough Forest method. The vehicle tracking system enhanced the accuracy of the lane detection result in high-density traffic scenarios.

In terms of road painting recognition, Qin *et al*. proposed a general framework of road marking detection and classification [63]. Four common road markings (lanes, arrows, zebra-crossing, and words) were detected and classified separately using a support vector machine. However, this system only identifies the different kinds of road marking without further context explanation of each road marking. It is believed that road marking recognition results contribute to a better understanding of ego-lanes and help decide current lane types such as a right/left turning lanes [64], [65], etc. Finally, many researches were dedicated to the integration of road detection and lane detection [4], [7], [66] − [68]. The Tesla and Mobileye are all reported to use a road segmentation to refine the lane detection algorithms [69], [70]. Road area is usually detected before lanes since an accurate recognition of road area increases the lane marking searching speed and provides an accurate ROI for lane detection. Besides, since the road boundaries and lanes are correlated and normally have the same direction, a road boundary orientation detection enhances the subsequent lane detection accuracy. Ma *et al*. proposed a Bayesian framework to integrate road boundary and lane edge detection [71]. Lane and road boundaries were modelled with a second-order model and detected using a deformable template method.

Fritsch *et al*. [7] proposed a road and ego-lane detection system particularly focusing on inner-city and rural roads. The proposed road and ego-lane detection algorithm was tested in three different road conditions. Another integrated road and ego-lane detection algorithm for urban areas was proposed in [72]. Road segmentation based on an illumination invariant transform was the prior step for lane detection to reduce the lane detection time and increase the detection accuracy. The outputs of the system consisted of road region, ego-lane region and markings, local lane width, and the relative position and orientation of the vehicle.

3.3.4 Sensor Level Integration

Sensor fusion dramatically improves the lane detection performance since more sensors are used and perception ability is boosted. Using multiple cameras including monocular, stereo cameras, or combining multiple cameras with different field of view are the most common ways to enhance the lane detection system [46], [55], [73]. In [73], a dense vanishing point detection method for lane detection using stereo camera was proposed. The combination of global dense vanishing point detection and stereo camera makes the system very robust to various road conditions and multiple lanes scenarios. Bertozzi and Broggi proposed a generic obstacle and lane detection (GOLD) system to detect obstacles and lanes based on stereo camera and IPM image [55]. The system was tested on the road for more than 3000 km and it showed robustness under exposure to shadow, illumination, and road variation. In three wide-field cameras and one tele-lens camera were combined and sampled at the frequency of 14 Hz. Raw images were converted to HSV format and IPM was performed. In an around view monitoring (AVM) system with four fish eye cameras and one monocular front-looking camera are used for lane detection and vehicle localization. The benefit of using AVM system is that a whole picture of the top-view of the vehicle can be generated, which contains the front, surrounding, and rear view of the vehicle in one single image.

Instead of using only camera devices, lane detection system also can be realised by combining cameras with global positioning system (GPS) and RADAR [76]−[82]. An integration system based on vision and RADAR was proposed in [71]. RADAR was particularly used for road boundary detection in ill-illuminated conditions. Jung *et al*. proposed an adaptive ROI-based lane detection method aimed at designing an integrated ACC and LKA system [76]. Range data from ACC was used to determine a dynamic ROI and improve the accuracy of monocular vision-based lane detection system. Lane detection system was designed using a conventional method, which includes edge distribution function (ED), steerable filter, model fitting and tracking. If nearby vehicles were detected with the range sensor, all the edge pixels were eliminated to enhance the lane detection. Final results show that recognition of nearby vehicles based on the range data improves lane detection accuracy and simplifies the detection algorithm.

Cui *et al*. [77] proposed an autonomous vehicle positioning system based on GPS and vision system. Prior information like road shape was first extracted from GPS and then

used to refine the lane detection system. The proposed method was extensively evaluated and found to be robust in varying road conditions. Jiang *et al*. proposed an integrated lane detection system in a structured highway scenario [78]. Road curvatures were determined using GPS and digital maps in the beginning. Then, two-lane detection modules designed for straight lanes and curved lanes were selected accordingly. Schreiber *et al*. [83] introduced a lane marking-based localisation system. Lane markings and curbs were detected with a stereo camera and vehicle localisation was performed with the integration of a global navigation satellite system (GNSS), high accuracy map and stereo vision system. The integrated localisation system achieved accuracy up to a few centimetres in rural areas.

An integrated lane departure warning system using GPS, inertial sensor, high-accuracy map, and vision system was introduced in [84]. Vision-based LDW was easily affected by various road conditions and weather. A sensor fusion scheme increases the stability of the lane detection system and makes the system more reliable. Moreover, a vision-based lane detection system and an accurate digital map help reduce the position errors from GPS, which lead to a more accurate vehicle localization and lane keeping.

Lidar was another widely used sensor and was the primary sensor used in most autonomous vehicles in the DARPA challenge [85], [86], due to its high accuracy and robust sensing ability. Lane markings are on-road paintings that have higher reflective properties than the road surface in the 3D points cloud map given by Lidar. Therefore, Lidar can detect lane markings according to those high reflectance points on the road. Lidar uses multiple channel laser lights to scan surrounding surfaces and build 3D images. Therefore, Lidar and vision integrated lane detection systems can be more accurate and robust to shadows and illumination change than vision-based systems [87]. Shin *et al*. proposed a lane detection system using camera and Lidar [88]. The algorithm consists of ground road extraction, lane detection with multi-modal data, and lane information combination. The proposed method shows a high detection accuracy performance (up to 90% accuracy) in real world experiments. Although camera and Lidar-based methods can cope with curved lanes, shadow, and illumination issues, it requires a complex co-calibration of the multi-modal sensors. Amaradi *et al*. proposed a lane-following and obstacle detection system using camera and Lidar [89]. Lanes are first detected with Hough Transform. Lidar was used to detect obstacles and measure the distance between

the ego-vehicle and front obstacles to plan an obstacle free driving area. In [56], a fusion system of multiple cameras and Lidar was proposed to detect lane markings in urban areas. The test vehicle was reported as the only vehicle that used vision-based lane detection algorithm in the final stage of the DARPA urban challenge. The system detects multiple lanes followed by the estimation and tracking of the centre lines. Lidar and cameras were first calibrated to detect road paint and curbs. Lidar was used to reduce the false positive detection rate by detecting obstacles and drivable road area.

According to the implementation angle and surveying distances, the laser scanner device can efficiently identify the lane marking. Lane detection using this laser reflection method have also been widely applied [80], [90]−[94]. Li *et al*. [80] proposed a drivable region and lane detection system based on Lidar and vision fusion at the feature level. The test bed vehicle uses two cameras mounted at different angles and three laser scanners. The algorithm detects the optimal drivable region using multi-modal sensors. The system was able to work under both structured and unstructured roads without any prior terrain knowledge. A laser-camera system for lane detection was introduced in [91]. The two-dimensional laser reflectivity map was generated on the roof of the vehicle. Instead of using constrained rule-based methods to detect lanes on the reflectivity map, a DBSCAN algorithm was applied to automatically determine the lane positions and the number of lanes in the field according to the 2D map. In [93], an integration system with laser scanner and stereo cameras was proposed. The system achieved an accurate driving area detection result even in the desert area. However, in some unstructured road or dirty road, the signals from laser scanner may carry more noise than the frame signals from the camera. Therefore, a signal filtering for the laser scanner and a sensor fusion are usually needed for the integrated systems.

In this section, some sensors that are relevant to lane detection task are reviewed. Other sensors such as vehicle dynamic signals such as the velocity, longitudinal/lateral acceleration, yaw angle, etc, inertial measurement unit (IMU) are also commonly used in the construction of a complete vehicle perception system. Although Lidar-based lane detection system can be more precise than other systems, the cost is still too high for public transport. Therefore, recent studies like [79] tend to fuse sensors such as GPS, digital map, and cameras, which are already available in commercial vehicles, to design a robust lane detection and driver assisting system.

3.4 Evaluation Methodologies for Vision-Based Lane Detection Systems

Most of the previous lane detection studies used visual verification to evaluate the system performance due to the lack of ground data, and only few researches proposed quantitative performance analysis and evaluation. In addition, lane detection evaluation is a complex task since the detection methods can vary across hardware and algorithms. There are still no common metrics that can be used to comprehensively evaluate each aspect of lane detection algorithms. An accurate lane detection system in one place is not guaranteed to be accurate in another place since the road and lane situation in different countries or areas differ significantly. Some detection algorithms may even show significantly different detection results in days and nights. It is also not fair to say that a monocular vision-based system is not as good as a system with vision and Lidar fusion and use a complex synergistic algorithm since the system cost is higher.

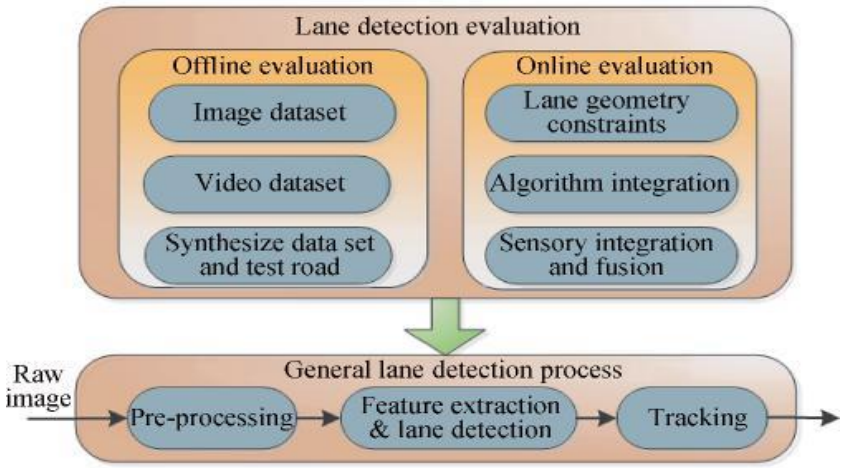

Figure 3-3. Lane detection evaluation architecture with two different evaluation methodologies.

Therefore, the performance evaluation of lane detection systems is necessary, and it should be noted that the best index for the lane detection performance is the driving safety issues and how robust the system is to the environment change. In this section, the evaluation methodologies used in studies are divided into offline evaluation and online evaluation categories where the online evaluation can be viewed as a process of calculating the detection confidence in real time. The main evaluation architecture is shown in Figure 3-3. As mentioned earlier, common vision-based lane detection system can be roughly separated into three parts, which are the pre-processing, lane detection,

and tracking. Accordingly, evaluation can be applied to all these three parts and the performance of these modules can be assessed separately. In the following section, influencing factors that affect the performance of a lane detection system will be summarized first. Then, the offline and online evaluation methods used in past studies and other literature are described. Finally, the evaluation metrics will be discussed.

3.4.1 Influential Factors for Lane Detection Systems

Vision-based lane detection systems studied in previous studies differed in terms of hardware, algorithms, and application situations. Some focus on the highway implementation while some systems were tested in urban areas. An accurate highway-oriented lane detection system is not guaranteed to be accurate in urban road areas since more disturbance and dense traffic will be observed in such areas. Therefore, it is impossible to use one single evaluation method or metric to assess all the existing systems. Some important factors that can affect the performance of lane detection system are listed in Table 3-1. A fair evaluation and comparison of lane detection systems should take these factors and the system working environment into consideration. Since different lane detection algorithms are designed and tested for different places, different road and lane factors in different places will affect the detection performance. Moreover, the data recording device, the camera or other vision hardware are other aspects that can significantly influence lane detection systems. For example, the lane detection systems may have different resolution and field of view with different cameras, which will influence the detection accuracy. Lastly, some traffic and weather factors can also lead to a different lane detection performance.

TABLE 3-1. FACTORS THAT INFLUENCE LANE DETECTION SYSTEMS

Lane and road factors	Crosswalk stop lane, lane colour, lane style, road curvature, poor quality lane markings, complex road texture
Hardware factors	Camera types, camera calibration, camera mounting position, other sensors
Traffic factors	Road curbs, guardrail, surrounding vehicles, shadow, illumination issues, vibration
Weather factors	Cloudy, snowy, rainy, foggy

As shown in Table 3-1, many factors can cause a less accurate detection result and make the performance vary with other systems. For example, some lane detection systems were tested under a complex traffic context, which had more disturbances like crosswalks or poor-quality lane markings, while some other systems were tested in standard highway

environments with few influencing factors. Therefore, an ideal way is to use a common platform for algorithm evaluation, which is barely possible in real life. Therefore, a mature evaluation system should take as many influential factors as possible into account and comprehensively assess the performance of the system. One potential solution for these problems is using parallel vision architecture, which will be discussed in the next section.

In the following part, the methodologies and metrics that can be used to propose a reasonable performance evaluation system are described.

3.4.2 Offline Evaluation

Offline evaluation is commonly used in previous literatures. After the framework of a lane detection system has been determined, system performance is first evaluated offline using still images or video sequences. There are some public datasets such as KITTI Road and Caltech Road [7], [25] that are available on the internet. KITTI Road dataset consists of 289 training images and 290 testing images separated into three categories. The road and ego-lane area were labelled in the dataset. The evaluation is usually done using ROC curves to illustrate the pixel-level true and false detection rate. Caltech Road dataset contains 1224 labelled individual frames captured in four different road situations. Both these datasets focus on evaluating road and lane detection performance in urban areas. The main drawbacks of image-based evaluation methods are that they are less reflective of real traffic environments and the datasets contain limited annotated test images.

On the other hand, video datasets depict much richer information and enable the reflection of real-life traffic situations. However, it normally requires more human resources to label ground-truth lanes. To deal with this problem, Borkar *et al*. [95] proposed a semi-automatic method to label lane pixels in video sequences. They used the time-sliced (TS) images and interpolation method to reduce the labelling workload. The time-sliced images were constructed by selecting the same rows from each video frame and re-arranging these row pixels according to the frame order. Two or more TS images were required, and the accuracy of ground truth lanes was directly proportional to the number of images. The lane labelling tasks are converted to point labelling in the TS images. After the labelled ground truth points were selected from each TS image, the interpolated ground-truth lanes can be recovered into the video sequence accordingly. The authors significantly reduced the ground truth labelling workload by converting lane

labelling into few points labelling tasks. This method was further improved in [49] by using a so-called modified min-between-max thresholding algorithm (M2BMT) applied to both time-slices and spatial stripes of the video frames.

Despite manual annotated ground truth, some researchers use the synthesis method to generate lane images with known position and curvature parameters in simulators [28], [56]. Lopez *et al*. [28] used a MATLAB simulator to generate video sequences and ground truth lanes. Lane frames were created with known lane parameters and positions. This method was able to generate arbitrary road and lane models with an arbitrary number of video frames. Using a simulator to generate lane ground truth is an efficient way to assess the lane detection system under ideal road conditions. However, there are few driving simulators that can completely simulate real world traffic context at this moment. Therefore, the detection performance still has to be tested with real-world lane images or videos after evaluation using simulators. Another way is to test the system on real-world testing tracks to assess the lane detection system compared to the accurate lane position ground truth provided by GPS and high precision maps [79].

3.4.3 Online Evaluation

The online evaluation system combines road and lane geometry information and integrates with other sensors to generate a detection confidence. Lane geometry constraints are reliable metrics for online evaluation. Once the camera is calibrated and mounted on the vehicle, road and lane geometric characteristics such as the ego lane width can be determined. In [96], a real-time lane evaluation method was proposed based on width measurement of the detected lanes. The detected lanes were verified based on three criteria, which are the slopes and intercept of the straight lane model, the predetermined road width, and position of the vanishing point. The distribution of lane model parameters was analysed, and a look-up table was created to determine the correctness of the detection. Once the detected lane width exceeds the threshold, re-estimation is proposed with respect to the lane width constraints.

In [5], the authors used a world-coordinate measurement error instead of using errors in image coordinates to assess the detection accuracy. A road side down-facing camera was used to directly record lane information, generate ground truth, and estimate vehicle position within the lanes. In [50], [51], real-time confidence was calculated based on the similarity measurement of the results given by different detection algorithms. The

evaluation module calculates if the detected lane positions from different algorithms are within a certain distance. If similar results are obtained, then the detection results are averaged, and a high detection confidence is reported. However, this method requires performing two algorithms simultaneously at each step, which increases the computation burden.

In [56], vision and Lidar-based algorithms were combined to build a confidence probability network. The travelling distance was adopted to determine the lane detection confidence. The system was said to have a high estimation confidence at certain meters in front of the vehicle if the vehicle can travel safely at that distance. Other online evaluation methods like estimating the offsets between the estimated centre line and lane boundaries were also used in previous researches. Instead of using single sensor, vision-based lane detection results can be evaluated with other sensors such as GPS, Lidar, and highly accurate road models [56], [77]. A vanishing point lane detection algorithm was introduced in [97]. Vanishing point of lane segments were first detected according to a probabilistic voting method. Then, the vanishing point along with the line orientation threshold were used to determine correct lane segments. To further reduce the false detection rate, a real time inter-frame similarity model for evaluation of lane location consistency was adopted. This real time evaluation idea was also under the assumption that lane geometry properties do not change significantly within a short period of continuous frames.

3.4.4 Evaluation Metrics

Existing studies mainly use visual evaluation or simple detection rates as evaluation metrics since there are still no common performance metrics to evaluate the lane detection performance. Li *et al*. [98] designed a complete testing scheme for intelligent vehicles mainly focusing on the whole vehicle performance rather than just the lane detection system. In [20], five major requirements for a lane detection system were given: shadow insensitivity suitable for unpainted roads, handling of curved roads, meeting lane parallel constraints, and reliability measurement. Kluge [99] introduced feature level metrics that measure the gradient orientation of the edge pixels and angular deviation entropy. The proposed metrics evaluate edge points and required road curvatures and vanishing point information.

Veit *et al*. [100] proposed another feature-level evaluation based on a hand labelled dataset exceeding 100 images. Six different lane feature extraction algorithms were compared. The authors concluded that the lane feature extraction, which combines photometric and geometric features, will achieve the best result. McCall and Trivedi [101] examined the most important evaluation metrics to assess the lane detection system. They concluded that it is not appropriate to view the system as a whole and use detection rates as the metrics. Instead, three different metrics, which include standard deviation of error, mean absolute error, and standard deviation of error in rate of change were used.

Satzoda and Trivedi [102] introduced five metrics to measure different properties of lane detection systems and to examine the trade-off between accuracy and computational efficiency. The five metrics consist of the measurement of lane feature accuracy, ego-vehicle localisation, lane position deviation, computation efficiency and accuracy, and the cumulative deviation in time. Among these metrics, cumulative deviation in time helps determine the maximum amount of safety time and can be used to evaluate if the proposed system meets the critical response time of ADAS. However, all of these metrics pay more attention to the detection accuracy assessment and do not consider the robustness.

In summary, a lane detection system can be evaluated separately from the pre-processing, lane detection algorithms, and tracking aspects. Evaluation metrics are not limited to measuring the error between detected lanes and ground truth lanes but can also be extended to assess the lane prediction horizon, the shadow sensitivity, and the computational efficiency etc. The specific evaluation metrics for a system should be determined based on the real-world application requirements. There are three basic properties of a lane detection system, which are the accuracy, robustness, and efficiency. The primary objective of the lane detection algorithm is to meet the real-time safety requirement with acceptable accuracy and at low computational cost. Accuracy metrics measure if the algorithm can detect lanes with small error for both straight and curved lanes. Lane detection accuracy issues have been widely studied in the past and many metrics can be found in literature. However, the robustness issues of the detection system are still not sufficiently studied. Urban road images are usually used to assess the robustness of the system since more challenges will be encountered in such situations.

TABLE 3-2 SUMMARIZE OF VARIOUS PREVIOUS LANE DETECTION SYSTEMS

Ref.	Preprocessing	Lane Detection	Tracking	Integration	Evaluation	Comments
[11]	IPM	Lane marking clustering	Particle filter	Lidar and CCD camera	Frame images and visual assessment	Avoid strong assumption to lane geometry and use weak tracking models
[5]	IPM, steerable filters, adaptive template	Statistical and motion-based outlier removal	Kalman filter	Cameras, Laser ranger, GPS, CAN	Quantitative analysis using evaluation metrics	Rich experiments and metrics applied to test the VioLET system
[58]	Temporal blur, IPM, adaptive Threshold	RANSAC	Kalman filter	Camera	Quantitative analysis and visual assessment	The proposed ALD 2.0 is used for efficient Video ground truth labeling
[94]	EDF	Hough Transform	None	Camera	Frame images and visual assessment	Road are divided into near field and far field with straight and curve model
[52]	Road detection, centerline estimation	RANSAC	Route Network Description File	Lidar and cameras	Confidence and centerline evaluation	Obstacle detection and free road area is determined before lane detection
[16]	Vanishing point detection, Canny edge detector	Control point detection	None	Camera	Frame images and visual assessment	The proposed B-snake model is robust to shadow and illumination variation
[23]	Ridge feature	RANSAC	None	Camera	Quantitative analysis	Synthesized lane ground truth data are generated with known geometry parameters
[20]	IPM, Gaussian kernel filter	Hough Transform, RANSAC	None	Camera	Quantitative analysis with public Caltech dataset	The proposed method is robust to shadow and curves but can be influence by crosswalks and road painting
[74]	Layered ROI, steerable filter	Hough Transform	Kalman filter	Radar and camera fusion	Visual assessment and correct detect rate metrics	Adaptive ROI created with range data makes lane detection robust to nearby vehicles and other road markings
[95]	IPM, 2nd and 4th steerable filter	RANSAC	None	Camera,	Performed on KITTI dataset using correct and false positive rate	Detection algorithm is robust to shadow, integrate with optical flow for lane departure aware
[69]	ROI, IPM	CNN, RNN	None	Surrounding cameras	Quantitative analysis with ROC curve	Proposed RNN use long-short-term-memory can capture lane spatial structures over a period of time in the video sequences
[82]	ROI, artificial image generating	CNN	None	Two lateral cameras facing down the road	Pixel level distance evaluation	End-to-End lane recognition procedure and able to apply in real time
[21]	IPM, temporal blur	RANSAC	Kalman filter	Camera	Visual assessment and correct detect rate metrics	Lane detection algorithm is designed mainly focus on night vision
[36]	YCbCr colour space transform, vanishing point detection	Lane turning point detection	None	Lane and vehicle integration using single camera	Frame images and visual assessment	Vehicles that have same colour with lanes are distinguished with shape, size, and motion information
[37]	YIQ colour space transform, vanishing point detection	Fan-scanning line detection	None	Lane and front vehicle integration using single camera	visual assessment and correct detection rate	The highway lane departure warning and front collision system is built with straight lane model
[75]	Median filter, ground plane extraction	Lane segmentation	None	Camera and Lidar integration	visual assessment and correct detection rate	Lane position detected with vision and Lidar is fused with a voting scheme
[34]	IPM, adaptive threshold	Morphological filters	None	Stereo camera for lane and obstacle detection	Frame images and visual assessment	Lanes are detected mainly with colour features which may be less robust to illumination change
[35]	IPM, steerable filter	RANSAC	Kalman filter	Lane and nearby vehicles integration using single camera	Evaluate using hand label frames with multiple metrics	Lane detection is robust in heavy traffic situation with improved surrounding vehicle detection and localization
[47]	IPM	Template matching	None	Camera, IMU and GPS fusion	Evaluate using hand label frames with mean absolute error (MAE) metrics	Lanes detected with camera is cross validate with road geometry knowledge given by road map and GPS to improve detection accuracy
[31]	Lane marking texture extraction	Scanning line	Kalman filter	Camera	Frame images and visual assessment	Two low level detections is combined with results similarity comparison
[48]	Dynamic thresholding, Canny edge detector	Hough Transform and least square model fitting	Kalman filter	Camera, IMU, Lidar, and GPS fusion	Spatial and slope criterion for real time assessment and MAE with ground truth position	A robust redundant lane detection and lateral offset measurement is proposed based on the detection given by camera and Lidar
[50]	Prewitt vertical gradient, adaptive threshold	probabilistic Hough Transform	None	IMU, GPS, Lidar and cameras fusion	visual assessment and correct detection rate	Lane marking detection is performed only after road and optimal drivable area is detected based on sensor fusion

Some representative lane detection studies are illustrated in Table 3-2. Specifically, in Table II, the pre-processing column records the image processing methods used in the literature. The integration column describes the integration methods used in the study, which may contain different levels of integration. Frame images and visual assessment in the evaluation column indicate that the proposed algorithm was only evaluated with still images and visual assessment method without any comparison with ground truth information. As shown in previous studies, a robust and accurate lane detection system usually combines detection and tracking algorithms. Besides, most advanced lane detection systems integrate with other objects detection systems or sensors to generate a more comprehensive detection network.

3.5 Discussion

In this part, the current limitation of vision-based lane detection algorithm, integration, and evaluation are analysed based on the context of above sections firstly. Next, the framework of parallel vision-based lane detection system, which is regarded as a possible efficient way to solve the generalization and evaluation problems for lane algorithm design will be discussed.

3.5.1 Current Limitation and Challenges

Lane detection systems have been widely studied and successfully implemented in some commercial ADAS products in the past decade. A large volume of literature can be found, which uses vision-based algorithms due to the low cost of camera devices and extensive background knowledge of image processing. Although vision-based lane detection system suffers from illumination variation, shadows, and bad weathers, it is still widely adopted and will continue dominating the future ADAS markets. The main objective of lane detection system is to design an accurate and robust detection algorithm. Accuracy issues were the main concerns of previous studies and many novel methods that are based on machine learning and deep learning methods are designed to construct a more precise system. However, the robustness issues are the key aspects that determine if a system can be applied in real life. The huge challenge to future vision-based systems is to maintain a stable and reliable lane measurement under heavy traffic and adverse weather conditions.

Considering this problem, one efficient method is to use the integration and fusion techniques. It has been proved that a single vision-based lane detection system has its limitation to deal with the varying road and traffic situation. Therefore, it is necessary to prepare a back-up system that can enrich the functionality of ADAS. Basically, a redundancy system can be constructed in three ways based on algorithm, system, and sensor level integration. Algorithms integration is a choice with the lowest cost and easiest to be applied. A system level integration combines lane detection system with other perception systems such as road and surrounding vehicles detection to improve the accuracy and robustness of the system. However, the two integration methods still rely on camera vision systems and have their inevitable limitations. Sensor level integration, on the other hand, is the most reliable way to detect lanes under different situations.

Another challenging task in lane detection systems is to design an evaluation system that can verify the system performance. Nowadays, a common problem is the lack of public benchmarks and data sets due to the difficulty of labelling lanes as the ground truth. Besides, there are no standard evaluation metrics that can be used to comprehensively assess the system performance with respect to both accuracy and robustness properties. Online confidence evaluation is another important task for lane detection system. For ADAS and lower level automated vehicles, the driver should be alerted once a low detection confidence occurs. In terms of autonomous vehicles, it is also important to let the vehicle understand how it does in the lane detection task, which can be viewed as a self-aware and diagnostic process.

3.5.2 Apply Parallel Theory into Vision-Based Lane Detection Systems

Considering the issues, a novel parallel framework for lane detection system design will be proposed in this part. The parallel lane detection framework is expected to be an efficient tool to assess the robustness as well as the evaluation issues for the lane detection system.

Parallel system is the product of advanced control systems and the computer simulation systems. It was introduced by Fei-Yue Wang and developed to control and manage complex systems [105]−[107]. The parallel theory is an efficient tool that can compensate the hard modelling and evaluating issue for the complex systems. The main objective of parallel system is to connect the real-world system with one or multiple artificial virtual systems that are in the cyberspace. The constructed virtual systems will have similar

characteristics as the real-world complex system but not the same. Here, parallel refers to a parallel interaction between the real-world system and its corresponding virtual counterparts. By connecting these systems together, analysing and comparing their behaviours, the parallel system will be able to predict the future status of both the real-world systems and the artificial one. According to the response and behaviours of the virtual system, the parallel system will automatically adjust the parameters of the real-world model to control and manage the real-world complex system such that efficient solution will be applied.

The construction of parallel system requires the ACP theory as the background knowledge. ACP is short for Artificial Society, Computational experiments, and Parallel execution, which are the three major components of parallel system. The complex system is firstly modelled using a holistic approach, whereas the real-world system is represented using an artificial system. After this step, the virtual system in the cyberspace becomes another solution domain of the complex system, which contributes to the potential complete solution along with the natural system in the physical space. It is hard to say that one solution will satisfy all the real-world challenges. An effective solution for the control of complex system should have the ability to deal with various situations occurring in the future. However, the limited testing scenarios in the real world cannot guarantee the potential solution being comprehensively tested. Therefore, the computation experiment module will execute large amount of virtual experiments according to the constructed artificial system in last step. Finally, considering there are normally no unique solution for complex system, the parallel execution provides an effective fashion to validate and evaluate various solutions. The parallel execution module will online update the local optimal solution to the real-world system that is found in the cyberspace for better control and management [108].

Recently, a parallel vision architecture based on the ACP theory has been summarised and introduced to the computer vision society [109]. The parallel vision theory offers an efficient way to deal with the detection and evaluation problems of the vision-based object detection systems. Similarly, the general ideal of ACP theory within the parallel vision scope is to achieve perception and understanding of the complex real-world environment according to the combination of virtual realities and the real-world information. In terms of lane detection task, the first artificial societies module can be used to construct a virtual

traffic environment and various road scenes using computer graphics and virtual reality techniques. Next, in the computation experiments module, the unlimited labelled traffic scene images and the limited real world driving images can be combined to train powerful lane detector using machine learning and deep learning methods. This process also contains two sub-procedures namely learning and training, testing and evaluating. The large-scale dataset will benefit the model training task, after that, the large amount of near-real data will sufficiently facilitate the model evaluation. Finally, in the parallel execution process, the lane detection model can be trained and evaluated in a parallel scheme in both real world and the virtual environment. The lane detector can be online optimized according to its performance in the two parallel worlds.

In addition, the application of ACP parallel vision system will efficiently solve the generalization and evaluation problems due to the utilization of the large-scale near-real synthesis images. To improve the generalization of the lane detection system, the detectors can be tested on virtual environments that have high similarity with the real world. The performance also can be sufficiently evaluated from the accuracy and robustness perspectives. Various computation experiment and model testing procedure can be continuously executed. In the computational experiments, the cutting-edge deep learning and reinforcement learning techniques can be applied to improve the accuracy and generalisation of the system without considering the lack of labelled data. Meanwhile, some deep learning models like the generative adversarial networks (GAN) can be used to generate near-real road scene images which can reflect the real-world road characteristics such as illumination, occlusion, and poor visualization. In addition, in the virtual computational world, the GAN network can be trained to discriminate whether the lane markings exist in the input image.

Figure 3-4 shows a simplified architecture of the ACP-based lane detection system. The road and lane images are in parallel collected from the real world and the artificial cyberspace. The real-world data is then used as a guidance for generating near-real artificial traffic scenes, which are automatically labelled. Both of the real world data and synthesis data are fed into the data-driven computational level. Machine learning and deep learning methods are powerful tools in this level. For various driving scenarios occurred in both real-world and the parallel virtual world, the model training process will try to come up with the most satisfying model. After that, the lane detection model will be

exhaustively evaluated and validated in the cyberspace world according to large scale labelled data. Once a well-trained and evaluated lane detection model is constructed, the model can be applied in parallel to both real world environment and virtual world for real-time lane detection evaluation. Due to the safety, human resource limitation, and energy consumption, the number of experiments in real world are limited, which may not be able to deal with all the challenges from the road [110], [111]. In contrary, the experiments in the parallel virtual world are safer and economical to be applied, moreover, the virtual world can simulate much more situations that are less possibly occur in the real world. Meanwhile, by using online learning technique, the experience from the continuous learning and testing module in the virtual world will improve the real-world performance.

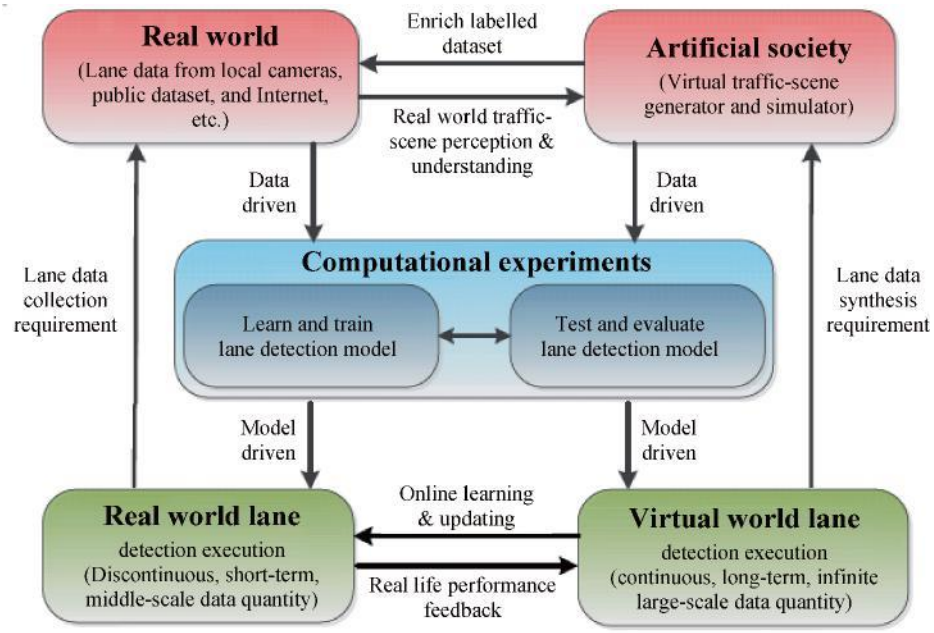

Figure 3-4. Simple Architecture of ACP-based Parallel Lane Detection and Evaluation System.

Some previous literatures have partially applied the parallel vision theory into the construction of lane detection system [28], [56]. These studies try to simulate the lane detection model within the simulation environment and process the lane detection model with the first two steps of ACP architecture. However, to construct an actual parallel system, the ACP architecture should be treated. The final parallel execution step of ACP theory is the core of parallel system. This step will online update the real-world model and adjust the corresponding model parameters according to the testing results in the parallel worlds. This step is also the core step, which guarantees that the learned lane detection model can be satisfied by various real-world driving scenarios. Despite applying

parallel theory into the design of intelligent transport and vehicles, it has been widely used in some other domains. For example, Deepmind use multiple processor to train their AlphaGo based on the deep reinforcement learning methods [112]. The idea behind the reinforcement learning in this case is to construct a parallel virtual world for the virtual go player to do exercise. In summary, the parallel theory is drawing increasing attention from the researchers. The utilization of parallel vision techniques in the future is expected to become another efficient way to solve the generalization and evaluation problems for the lane detection algorithms. The ACP-based parallel lane detection system will not only assist to build an accurate model that is well tested and assessed, but also enable the intelligent vehicles to carefully adjust their detection strategies in real-time. Meanwhile, since there are too many different lane detection methodologies which are hardly evaluated uniformly, a public virtual simulation platform can be used to compare these algorithms in the future. Those algorithms which achieve satisfactory performance in the parallel virtual worlds can then be implemented in the real world.

3.6 Conclusion

In this study, vision-based lane detection systems are reviewed from three aspects, namely algorithms, integration, and evaluation methods. Existing algorithms are summarized into two categories, which are conventional image processing-based, and novel machine learning (deep learning)-based methods. Next, previous integration methods of the lane detection system are divided into three levels, which are algorithm level, system level, and sensor level. In algorithm level, multiple lane detection and tracking algorithms are combined in serial or parallel manner. System level integration combines vision-based lane detection with other road marking or obstacle detection systems. Sensor fusion enhances the vehicle perception system most significantly by fusion of multi-modal sensors. Finally, lane detection evaluation issues are analysed from different aspects. Evaluation methods are divided into offline performance assessment and online real-time confidence evaluation.

As mentioned earlier, although the vision-based lane detection system has been widely studied in the past two decades, it is hard to say that research in this area has been matured. In fact, there are still many critical studies that need to be done, such as efficient low-cost system integration and the evaluation system design, especially the construction of parallel lane detection system. Moreover, increasing amount of advanced object detection

algorithms and architectures have been developed to optimize the lane detection systems. The continuous studies and the application of these techniques will further benefit ADAS and automated driving industry. The ACP-based parallel lane detection approach holds significant potentials for future implementation.

3.7 Reference

[1] Bellis, Elizabeth, and Jim Page. National motor vehicle crash causation survey (NMVCCS) SAS analytical user's manual. No. HS-811 053. 2008.

[2] Gayko, Jens E. "Lane departure and lane keeping." *Handbook of Intelligent Vehicles*. Springer London, 2012. 689-708.

[3] Visvikis C, Smith T L, Pitcher M, *et al*. Study on lane departure warning and lane change assistant systems. *Transport Research Laboratory Project Rpt PPR*, 2008, 374.

[4] Bar Hillel, Aharon, *et al*. "Recent progress in road and lane detection: a survey." *Machine vision and applications* (2014): 1-19.

[5] McCall, Joel C., and Mohan M. Trivedi. "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation." *IEEE transactions on intelligent transportation systems* 7.1 (2006): 20-37.

[6] Yenikaya, Sibel, Gökhan Yenikaya, and Ekrem Düven. "Keeping the vehicle on the road: A survey on on-road lane detection systems." *ACM Computing Surveys (CSUR)* 46.1 (2013): 2.

[7] Fritsch, Jannik, Tobias Kuhnl, and Andreas Geiger. "A new performance measure and evaluation benchmark for road detection algorithms." *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013.

[8] Beyeler, Michael, Florian Mirus, and Alexander Verl. "Vision-based robust road lane detection in urban environments." *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014.

[9] Kang, Dong-Joong, and Mun-Ho Jung. "Road lane segmentation using dynamic programming for active safety vehicles." *Pattern Recognition Letters* 24.16 (2003): 3177-3185.

[10] Suddamalla, Upendra, *et al*. "A novel algorithm of lane detection addressing varied scenarios of curved and dashed lanemarks." *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*. IEEE, 2015.

[11] Collado, Juan M., *et al*. "Adaptative road lanes detection and classification." *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer Berlin Heidelberg, 2006.

[12] Sehestedt, Stephan, *et al*. "Robust lane detection in urban environments." *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007.

[13] Lin, Qing, Youngjoon Han, and Hernsoo Hahn. "Real-time lane departure detection based on extended edge-linking algorithm." *Computer Research and Development, 2010 Second International Conference on*. IEEE, 2010.

[14] Cela, Andrés F., *et al*. "Lanes Detection Based on Unsupervised and Adaptive Classifier." Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on. IEEE, 2013.

[15] Borkar, Amol, *et al*. "A layered approach to robust lane detection at night." Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS'09. IEEE Workshop on. IEEE, 2009.

[16] Kreucher, Chris, and Sridhar Lakshmanan. "LANA: a lane extraction algorithm that uses frequency domain features." *IEEE Transactions on Robotics and automation* 15.2 (1999): 343-350.

[17] Jung, Soonhong, Junsic Youn, and Sanghoon Sull. "Efficient lane detection based on spatiotemporal images." *IEEE Transactions on Intelligent Transportation Systems* 17.1 (2016): 289-295.

[18] Xiao, Jing, Shutao Li, and Bin Sun. "A Real-Time System for Lane Detection Based on FPGA and DSP." *Sensing and Imaging* 17.1 (2016): 1-13.

[19] Ozgunalp, Umar, and Naim Dahnoun. "Lane detection based on improved feature map and efficient region of interest extraction." *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, 2015.

[20] Wang, Yue, Dinggang Shen, and Eam Khwang Teoh. "Lane detection using spline model." *Pattern Recognition Letters* 21.8 (2000): 677-689.

[21] Wang, Yue, Eam Khwang Teoh, and Dinggang Shen. "Lane detection and tracking using B-Snake." Image and Vision computing 22.4 (2004): 269-280.

[22] Li, Xiangyang, *et al*. "Lane detection and tracking using a parallel-snake approach." *Journal of Intelligent & Robotic Systems* 77.3-4 (2015): 597.

[23] Lim, King Hann, Kah Phooi Seng, and Li-Minn Ang. "River flow lane detection and Kalman filtering-based B-spline lane tracking." *International Journal of Vehicular Technology* 2012 (2012).

[24] Jung, Cláudio Rosito, and Christian Roberto Kelber. "An improved linear-parabolic model for lane following and curve detection." *Computer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on*. IEEE, 2005.

[25] Aly, Mohamed. "Real time detection of lane markers in urban streets." *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.

[26] Borkar, Amol, Monson Hayes, and Mark T. Smith. "Robust lane detection and tracking with ransac and kalman filter." *Image Processing (ICIP), 2009 16th IEEE International Conference on.* IEEE, 2009.

[27] Lopez, A., *et al*. "Detection of Lane Markings based on Ridgeness and RANSAC." *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*. IEEE, 2005.

[28] López, A., *et al*. "Robust lane markings detection and road geometry computation." *International Journal of Automotive Technology* 11.3 (2010): 395-407.

[29] Chen, Qiang, and Hong Wang. "A real-time lane detection algorithm based on a hyperbola-pair model." *Intelligent Vehicles Symposium, 2006 IEEE*. IEEE, 2006.

[30] Tan, Huachun, *et al*. "Improved river flow and random sample consensus for curve lane detection." *Advances in Mechanical Engineering* 7.7 (2015): 1687814015593866.

[31] Hur, Junhwa, Seung-Nam Kang, and Seung-Woo Seo. "Multi-lane detection in urban driving environments using conditional random fields." *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013

[32] Bounini, Farid, *et al*. "Autonomous Vehicle and Real Time Road Lanes Detection and Tracking." *Vehicle Power and Propulsion Conference (VPPC), 2015 IEEE*. IEEE, 2015.
[33] Wu, Dazhou, Rui Zhao, and Zhihua Wei. "A multi-segment lane-switch algorithm for efficient real-time lane detection." *Information and Automation (ICIA), 2014 IEEE International Conference on*. IEEE, 2014.
[34] Zhou, Shengyan, *et al*. "A novel lane detection based on geometrical model and gabor filter." *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010.
[35] Niu, Jianwei, *et al*. "Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting." *Pattern Recognition* 59 (2016): 225-233.
[36] He, Bei, *et al*. "Lane marking detection based on Convolution Neural Network from point clouds." *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016.
[37] Li, Jun, Xue Mei, and Danil Prokhorov. "Deep neural network for structural prediction and lane detection in traffic scene." *IEEE transactions on neural networks and learning systems* (2016).
[38] Gurghian, Alexandru, *et al*. "DeepLanes: End-To-End Lane Position Estimation Using Deep Neural Networksa." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016
[39] Li, Xue, *et al*. "Lane detection based on spiking neural network and hough transform." *Image and Signal Processing (CISP), 2015 8th International Congress on*. IEEE, 2015.
[40] Kim, Jihun, *et al*. "Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection." *Neural Networks* (2016).
[41] He, Bei, *et al*. "Accurate and robust lane detection based on Dual-View Convolutional Neutral Network." *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016.
[42] Revilloud, Marc, Dominique Gruyer, and Mohamed-Cherif Rahal. "A new multi-agent approach for lane detection and tracking." *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016.
[43] Bertozzi, Massimo, *et al*. "An evolutionary approach to lane markings detection in road environments." *Atti del* 6 (2002): 627-636.
[44] Tsai, Luo-Wei, *et al*. "Lane detection using directional random walks." *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.
[45] Bai, Li, and Yan Wang. "Road tracking using particle filters with partition sampling and auxiliary variables." *Computer Vision and Image Understanding* 115.10 (2011): 1463-1471.
[46] Danescu, Radu, and Sergiu Nedevschi. "Probabilistic lane tracking in difficult road scenarios using stereovision." *IEEE Transactions on Intelligent Transportation Systems* 10.2 (2009): 272-282.
[47] Kim, ZuWhan. "Robust lane detection and tracking in challenging scenarios." *IEEE Transactions on Intelligent Transportation Systems* 9.1 (2008): 16-26.
[48] Shin, Bok-Suk, Junli Tao, and Reinhard Klette. "A superparticle filter for lane detection." *Pattern Recognition* 48.11 (2015): 3333-3345.
[49] Das, Apurba, Siva Srinivasa Murthy, and Upendra Suddamalla. "Enhanced Algorithm of Automated Ground Truth Generation and Validation for Lane Detection System by $ M^{2} BMT$." *IEEE Transactions on Intelligent Transportation Systems* (2016).
[50] Labayrade, Raphael, S. S. Leng, and Didier Aubert. "A reliable road lane detector approach combining two vision-based algorithms." *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004.
[51] Labayrade, Raphaël, *et al*. "A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance." *IEICE transactions on information and systems* 89.7 2006: 2092-2100.
[52] Hernández, Danilo Cáceres, Dongwook Seo, and Kang-Hyun Jo. "Robust lane marking detection based on multi-feature fusion." *Human System Interactions (HSI), 2016 9th International Conference on*. IEEE, 2016.
[53] Yim, Young Uk, and Se-Young Oh. "Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving." *IEEE Transactions on Intelligent Transportation Systems* 4.4 (2003): 219-225.
[54] Felisa, Mirko, and Paolo Zani. "Robust monocular lane detection in urban environments." *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010.
[55] Bertozzi, Massimo, and Alberto Broggi. "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection." *IEEE transactions on image processing* 7.1 (1998): 62-81.
[56] Huang, Albert S., *et al*. "Finding multiple lanes in urban road networks with vision and lidar." *Autonomous Robots* 26.2 (2009): 103-122.
[57] Cheng, Hsu-Yung, *et al*. "Lane detection with moving vehicles in the traffic scenes." *IEEE Transactions on intelligent transportation systems* 7.4 (2006): 571-582.
[58] Sivaraman, Sayanan, and Mohan Manubhai Trivedi. "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach." *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013): 906-917.
[59] Wu, Chi-Feng, Cheng-Jian Lin, and Chi-Yung Lee. "Applying a functional neurofuzzy network to real-time lane detection and front- vehicle distance measurement." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4 (2012): 577-589.
[60] Huang, Shih-Shinh, *et al*. "On-board vision system for lane recognition and front-vehicle detection to enhance driver's awareness." *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. Vol. 3. IEEE, 2004.
[61] Satzoda, Ravi Kumar, and Mohan M. Trivedi. "Efficient lane and vehicle detection with integrated synergies (ELVIS)." *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014.
[62] Kim, Huieun, *et al*. "Integration of vehicle and lane detection for forward collision warning system." *Consumer Electronics-Berlin (ICCE-Berlin), 2016 IEEE 6th International Conference on*. IEEE, 2016.
[63] Qin, B., *et al*. "A general framework for road marking detection and analysis." *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013.
[64] Kheyrollahi, Alireza, and Toby P. Breckon. "Automatic real-time road marking recognition using a feature driven approach." *Machine Vision and Applications* 23.1 (2012): 123-133.
[65] Greenhalgh, Jack, and Majid Mirmehdi. "Detection and Recognition of Painted Road Surface Markings." *ICPRAM (1)*. 2015.
[66] Oliveira, Gabriel L., Wolfram Burgard, and Thomas Brox. "Efficient deep models for monocular road segmentation." *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016.
[67] Kong, Hui, Jean-Yves Audibert, and Jean Ponce. "Vanishing point detection for road detection." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[68] Levi, Dan, *et al*. "StixelNet: A Deep Convolutional Network for Obstacle Detection and Road Segmentation." *BMVC*. 2015.
[69] Stein, Gideon P., Yoram Gdalyahu, and Amnon Shashua. "Stereo-assist: Top-down stereo for driver assistance systems." *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010.
[70] Raphael, Eric, *et al*. "Development of a camera-based forward collision alert system." *SAE International Journal of Passenger Cars-Mechanical Systems* 4.2011-01-0579 2011: 467-478.
[71] Ma, Bing, S. Lakahmanan, and Alfred Hero. "Road and lane edge detection with multisensor fusion methods." *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*. Vol. 2. IEEE, 1999.
[72] Beyeler, Michael, Florian Mirus, and Alexander Verl. "Vision-based robust road lane detection in urban environments." *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014.
[73] Ozgunalp, Umar, *et al*. "Multiple Lane Detection Algorithm Based on Novel Dense Vanishing Point Estimation." *IEEE Transactions on Intelligent Transportation Systems* 18.3 (2017): 621-632.
[74] Lipski, Christian, *et al*. "A fast and robust approach to lane marking detection and lane tracking." *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*. IEEE, 2008.
[75] Kim, Dongwook, *et al*. "Lane-level localization using an AVM camera for an automated driving vehicle in urban environments." *IEEE/ASME Transactions on Mechatronics* 22.1 (2017): 280-290.
[76] Jung, H. G., *et al*. "Sensor fusion-based lane detection for LKS+ ACC system." *International journal of automotive technology* 10.2 (2009): 219-228.
[77] Cui, Dixiao, Jianru Xue, and Nanning Zheng. "Real-Time Global Localization of Robotic Cars in Lane Level via Lane Marking Detection and Shape Registration." *IEEE Transactions on Intelligent Transportation Systems* 17.4 (2016): 1039-1050.
[78] Jiang, Yan, Feng Gao, and Guoyan Xu. "Computer vision-based multiple-lane detection on straight road and in a curve." *Image Analysis and Signal Processing (IASP), 2010 International Conference on*. IEEE, 2010.
[79] Rose, Christopher, *et al*. "An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS." *IEEE Transactions on Intelligent Transportation Systems* 15.6 (2014): 2615-2629.
[80] Li, Qingquan, *et al*. "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios." *IEEE Transactions on Vehicular Technology* 63.2 (2014): 540-555.
[81] Kammel, Soren, and Benjamin Pitzer. "Lidar-based lane marker detection and mapping." *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.
[82] Manz, Michael, *et al*. "Detection and tracking of road networks in rural terrain by fusing vision and LIDAR." *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011.
[83] Schreiber, Markus, Carsten Knöppel, and Uwe Franke. "Laneloc: Lane marking based localization using highly accurate maps." *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013.
[84] Clanton J M, Bevly D M, Hodel A S. A low-cost solution for an integrated multisensor lane departure warning system[J]. IEEE Transactions on Intelligent Transportation Systems, 2009, 10(1): 47-59.
[85] Montemerlo, Michael, *et al*. "Junior: The stanford entry in the urban challenge." *Journal of field Robotics* 25.9 (2008): 569-597.
[86] Buehler, Martin, Karl Iagnemma, and Sanjiv Singh, eds. *The DARPA urban challenge: autonomous vehicles in city traffic*. Vol. 56. springer, 2009.
[87] Lindner, Philipp, *et al*. "Multi-channel lidar processing for lane detection and estimation." *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*. IEEE, 2009.
[88] Shin, Seunghak, Inwook Shim, and In So Kweon. "Combinatorial approach for lane detection using image and LIDAR reflectance." *Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on*. IEEE, 2015.
[89] Amaradi, Phanindra, *et al*. "Lane following and obstacle detection techniques in autonomous driving vehicles." *Electro Information Technology (EIT), 2016 IEEE International Conference on*. IEEE, 2016.
[90] Dietmayer, Klaus, *et al*. "Roadway detection and lane detection using multilayer laserscanner." *Advanced Microsystems for Automotive Applications 2005*. Springer Berlin Heidelberg, 2005. 197-213.
[91] Hernandez, Danilo Caceres, Van-Dung Hoang, and Kang-Hyun Jo. "Lane surface identification based on reflectance using laser range finder." *System Integration (SII), 2014 IEEE/SICE International Symposium on*. IEEE, 2014.
[92] Sparbert, Jan, Klaus Dietmayer, and Daniel Streller. "Lane detection and street type classification using laser range images." *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001.
[93] Broggi, Alberto, *et al*. "A laserscanner-vision fusion system implemented on the terramax autonomous vehicle." *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006.
[94] Zhao, Huijing, *et al*. "A laser-scanner-based approach toward driving safety and traffic data collection." *IEEE Transactions on intelligent transportation systems* 10.3 (2009): 534-546.
[95] Borkar, Amol, Monson Hayes, and Mark T. Smith. "A novel lane detection system with efficient ground truth generation." *IEEE Transactions on Intelligent Transportation Systems* 13.1 (2012): 365-374.
[96] Lin, Chun-Wei, Han-Ying Wang, and Din-Chang Tseng. "A robust lane detection and verification method for intelligent vehicles." *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*. Vol. 1. IEEE, 2009.
[97] Yoo, Ju Han, *et al*. "A Robust Lane Detection Method Based on Vanishing Point Estimation Using the Relevance of Line Segments." *IEEE Transactions on Intelligent Transportation Systems* (2017).
[98] Li, Li, *et al*. "Intelligence Testing for Autonomous Vehicles: A New Approach." *IEEE Transactions on Intelligent Vehicles* 1.2 (2016): 158166.
[99] Kluge, Karl C. "Performance evaluation of vision-based lane sensing: Some preliminary tools, metrics, and results." *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*. IEEE, 1997.
[100] Veit, Thomas, *et al*. "Evaluation of road marking feature extraction." Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on. IEEE, 2008.
[101] McCall, Joel C., and Mohan M. Trivedi. "Performance evaluation of a vision based lane tracker designed for driver assistance systems." *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*. IEEE, 2005.
[102] Satzoda, Ravi Kumar, and Mohan M. Trivedi. "On performance evaluation metrics for lane estimation." *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014.
[103] Jung, Claudio Rosito, and Christian Roberto Kelber. "A robust linear-parabolic model for lane following." *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*. IEEE, 2004.

[104] Haloi, Mrinal, and Dinesh Babu Jayagopi. "A robust lane detection and departure warning system." *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015.
[105] F. Y. Wang, "Parallel system methods for management and control of complex systems," *Control Decision*, vol. 19, no. 5, pp. 485-489, 514, May 2004.
[106] F. Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans.Intell. Transp. Syst*., vol. 11, no. 3, pp. 630-638, Sep. 2010.
[107] F. Y. Wang, "Artificial societies, computational experiments, and parallel systems: A discussion on computational theory of complex social economic systems," *Complex Syst. Complexity Sci*., vol. 1, no. 4, pp.25-35, Oct.
[108] L. Li, Y. L. Lin, D. P. Cao, N. N. Zheng, and F. Y. Wang, "Parallel learning-a new framework for machine learning," *Acta Automat. Sin.*, vol. 43, no. 1, pp. 1-18, Jan. 2017.
[109] K. F. Wang, C. Gou, N. N. Zheng, J. M. Rehg, and F. Y. Wang, "Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives," Artif. Intell. Rev., vol. 48, no. 3, pp.298-328, Oct. 2017.
[110] F. Y. Wang, N. N. Zheng, D. P. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in CPSS: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 577—587, Sep. 2017.
[111] C. Lv, Y. H. Liu, X. S. Hu, H. Y. Guo, D. P. Cao, and F. Y. Wang, "Simultaneous observation of hybrid states for cyber-physical systems: A case study of electric vehicle powertrain," *IEEE Trans. Cybern.*, 2018, to be published, doi: 10.1109/TCYB.2017.2738003.
[112] Silver, David, *et al*. "Mastering the game of Go with deep neural networks and tree search." *Nature* 529.7587 (2016): 484-489.

4

PAPER III – Integrated Lane Detection System Design

Dynamic Integration and Online Evaluation of Vision-Based Lane Detection Algorithms

Authors:

Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis

This paper has been accepted by

IET Intelligent Transport Systems (2018)

Abstract

Lane detection techniques have been widely studied in the past two decades and applied in many advance driver assistance systems (ADAS). However, development of a robust lane detection system that can deal with various road conditions and efficiently evaluate its detection results in real time, is still of great challenge. Therefore, the construction of evaluation system is one of the primary objectives for lane detection systems. In this article, a vision-based lane detection and evaluation system by integrating two distinct lane detection algorithms is proposed. Firstly, the steerable filter and Hough Transform are used as a primary detection algorithm. A secondary algorithm, which combines Gaussian mixture model (GMM) for unsupervised lane segmentation and RANSAC for lane model fitting, is used to detect lanes when the primary algorithm encounters a low detection confidence. Then, a Kalman filter is applied to track and smooth the detected lane parameters. To detect the colour and line style of ego-lane and evaluate the lane detection system in real time, a sampling and voting technique is proposed, and the corresponding drivable area is determined. By combining the sampling and voting system with prior lane geometry knowledge, the evaluation system can efficiently recognise false detections. The proposed real-time lane detection and evaluation systems use a front-facing monocular camera. The system can work robustly in various complex situations (e.g. shadows, night, and lane missing scenarios). The proposed lane detection system is evaluated with both Caltech lanes dataset and Cranfield local dataset, and test results and comparisons are discussed.

Index Terms - Lane detection, Hough transform, Gaussian mixture model, performance evaluation, ADAS.

4.1 Introduction

Traffic accidents are mainly caused by human operational mistakes, such as inattention, misbehaviour and distraction [1]. A large amount of companies and institutes have participated in improving driving safety and reducing traffic accidents. Among all those techniques, road perception and lanes detection are of great significance. Lane detection is the foundation of many ADAS products such as lane departure warning (LDW) and lane keeping assistance (LKA) [2-6]. Since lane markings are painted on road for human visual perception and the cost of camera devices is acceptable to public transport, thus most of ADAS products adopt vision-based techniques to detect front lanes. Apart from accuracy and robustness improvement of the lane detection system, another important task is performance evaluation. From the functional perspective, lane detection system should be aware of inaccurate detections and properly adjust the detection and tracking algorithms and parameters accordingly [5] [6]. For ADAS products, when encountering a low detection confidence, it should alert the driver and request the driver to concentrate on the driving task. On the other hand, unlike ADAS which allows driver to intervene the vehicle control, highly automated vehicles are responsible for traffic monitoring all the time and must deal with low-accurate detection by themselves. This makes the self-evaluation of the lane detection system more crucial in high level automated vehicles.

System integration is an efficient policy for the construction of robust lane detection systems [7]. In this section, the lane detection algorithms, integration and evaluation methods investigated in literatures are reviewed firstly, and an integrated robust lane detection system combining two distinct detection methods is proposed. The primary algorithm is built with steerable filter and Hough transform with the assumption that lanes are straight in most situations in real life. To further improve the system performance, a second algorithm, which is based on GMM lane segmentation and RANSAC, is designed to detect both straight and curve lanes when the first algorithms fails. Then, the lane position parameters are tracked with Kalman filter. To evaluate the lane detection performance in real-time, an evaluation scheme based on prior lane geometry information, such as lane width and lane orientation range, is applied. Next, another evaluation method based on the proposed sampling and voting technique is used to further evaluate the system performance. Instead of detecting all candidate lanes in the image, our work focusses on ego-lane detection and determine whether the left and right area of ego-lane

are drivable or not on the basis of the lane type recognition. The lane types which contain lane colour and the line style (e.g. solid, dashed, and double solid) are determined with the lane sampling algorithm. Finally, the integrated lane detection system is tested with a local dataset of Cranfiled and a public Caltech lanes dataset. According to the test results, the system achieves steady and robust performance on both datasets. The proposed approach contributes to a robust lane detection with an algorithm level integration and online evaluation method.

4.2 Related Works

Survey of lane detection algorithms and the general framework of the detection systems have been proposed in [7]-[9]. In [8], McCall and Trivedi defined three lane detection objectives, which are lane departure warning, driver attention awareness and automated vehicle control system design. Hillel and Lerner *et al* [7] concluded that road colour, texture, boundaries, and lane markings are the main perception aspects for human drivers, and lane detection system is one of the major perception tasks of ADAS. Previous research usually divided lane detection algorithms into two categories, namely the feature-based [10]-[15] and model-based ones [16]-[20]. Feature-based methods rely on the detection of lane marking features such as colour, texture, and edges. Then, based on these features, Hough Transform is commonly used to detect straight lanes in the image. Model-based methods, on the other hand, usually assume lanes can be described with a specific model such as linear model, parabola model, and various kinds of spline models. Then, a lane model can be fitted using RANSAC or other model fitting methods.

One challenge of vision-based lane detection system is that its robustness usually cannot meet real life requirements due to the highly random properties of the traffic and road conditions. Therefore, to improve the accuracy and stability of ADAS, lane detection system is usually integrated with other object detection systems. Previous integration methods can be summarized into three levels, which are algorithm level, system level, and sensor level. Algorithm level integration has been widely studied in literatures. There are two basic integration architectures, which are the parallel [22] [23] and serial [17]-[21] combination types. The parallel-type integration processes multiple algorithms simultaneously, while the serial type usually combines algorithms in sequence. System integration increases the accuracy of lane detection and reduces the false detection rate by combining other detection systems to remove the non-lane pixels. For example, lane

detection can be integrated with vehicle and road detections to enhance the overall detection accuracy [24]-[26]. The last integration method is sensory fusion. The integration at sensor level improves the detection accuracy most significantly, since more sensors are used that the sensing ability of the vehicle is increased. Despite of using monocular camera, multiple cameras, stereo camera, Lidar, and GPS are also widely used to enhance the lane detection system [27]-[30].

The construction of lane detection evaluation systems is another challenging task. Evaluation of lane detection system is very complex, because detection methods distinct from hardware to algorithms. Besides, road conditions in different scenarios differ significantly. An accurate lane detection system in a certain area is not guaranteed to be accurate in another area, and the performance can also vary from day to night. Previous lane detection system usually uses visual evaluation method to assess the final detection performance because of lacking public dataset and difficulties in labelling ground truth lanes [32]. Popular evaluation methods can be roughly divided into offline evaluation and online evaluation. Offline evaluation has been widely used in literatures and public datasets, such as those studies conducted in KITTI Road and Caltech Road [26] [20]. Online confidence evaluation often combines road and lane geometry properties and other sensors to evaluate the detection performance and provide a confidence measurement. In [31], a real-time lane evaluation method was proposed based on the width of the detected lanes. In [8], a side-facing camera was used to directly sample road lane information and generate ground truth.

In the following, a robust lane detection system with efficient algorithm integration is designed. Two different algorithms are integrated and compensate for each other. Then, drivable area recognition and online system evaluation are proposed based on a lane sampling and voting scheme. The detailed methodology used in this work with test results will be introduced in next sections.

4.3 Experiment Setup

As aforementioned, lane detection evaluation usually suffers from data variation, which means there are no existing standard and metric for lane detection performance evaluation. In this work, a fair performance comparison will be made between the proposed algorithms and previous Caltech lane detection system [20]. Therefore, the open access Caltech road dataset will be adopted in this study. Meanwhile, a local dataset named

Cranfiled local road dataset will be used for further evaluation of our system. The Cranfiled local road dataset contains the video captured in the motorway of M1 from Cranfiled to Northampton. Videos are captured at the frequency of 30Hz using a CMOS camera mounted in the middle of the dashboard. One common passenger car was used as experiment vehicle to collect all the data during day and night. All the images are of VGA resolutions (640×480). Two datasets are used in this work; In this experiment, the algorithms are implemented in an Intel ® Core i7 2.5 GHz computer using MATLAB 2016a. Currently the system works with a frequency of 25fps in the laboratory environment.

4.4 Methodology

In this section, the lane detection algorithms used in this work is introduced. In the first part, the primary lane detection method using Gaussian steerable filter (SF) and HT is described. In the second part, the proposed secondary detection algorithm based on GMM and RANSAC is introduced, which is initiated when the primary algorithm fails or being detected to have a low detection confidence according to the lane sampling and voting method. In part C, a Kalman filter is introduced to track the detected lane positions. Lastly, in part D, the sampling and voting scheme for lane type recognition and lane evaluation is described.

4.4.1 Lane Detection Using SF and HT method

In this work, a primary lane detection algorithm based on the combination of steerable filter and Hough Transform is built. The detection process follows a basic lane detection procedure, which consists of image capture and pre-processing, edge extraction, and lane detection.

1) Image Processing

Three tasks are proposed in this step. The captured images are firstly cropped with a pre-determined ROI, as shown in Figure 4-1. (a). Then decorrelation stretching is used to enhance the colour contrast between lanes makings and road surface. Lastly, the image is converted to grayscale image.

2) Edge Extraction

A second order Gaussian steerable filter is used to extract edge information. Steerable filter is able to detect edges in any orientation [35]. The idea behind is the directional derivative operator of Gaussian Function G shown in equation (1) is steerable.

$$G(x,y) = e^{-\frac{x^2+y^2}{\sigma^2}} \tag{4-1}$$

where x, y are Cartesian coordinates, σ is the variance of Gaussian kernel.

The steerable filter is constructed with a set of base filters. Dues to the separable properties, it is more flexible and faster than other edge detectors. The second order Gaussian steerable filter used in this work for edge extraction is defined as:

$$G_{xx}(x,y) = \frac{\partial}{\partial x^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} = (\frac{4x^2}{\sigma^4} - \frac{2}{\sigma^2}) e^{\frac{-(x^2+y^2)}{\sigma^2}} \tag{4-2}$$

$$G_{xy}(x,y) = \frac{\partial}{\partial x}\frac{\partial}{\partial y} e^{-\frac{(x^2+y^2)}{\sigma^2}} = \frac{4xy}{\sigma^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} \tag{4-3}$$

$$G_{yy}(x,y) = \frac{\partial^2}{\partial y^2}\ e^{-\frac{(x^2+y^2)}{\sigma^2}} = (\frac{4y^2}{\sigma^2} - \frac{2}{\sigma^2}) e^{-\frac{(x^2+y^2)}{\sigma^2}} \tag{4-4}$$

The combination of these base filters makes steerable filter able to calculate the filter response at arbitrary orientations, and the combination method is shown as follows [35].

$$G_2^\theta = cos^2(\theta) G_{xx} + sin^2(\theta) G_{yy} - 2\sin(\theta)\cos(\theta)\, G_{x,y} \tag{4-5}$$

where θ represents the orientation angle.

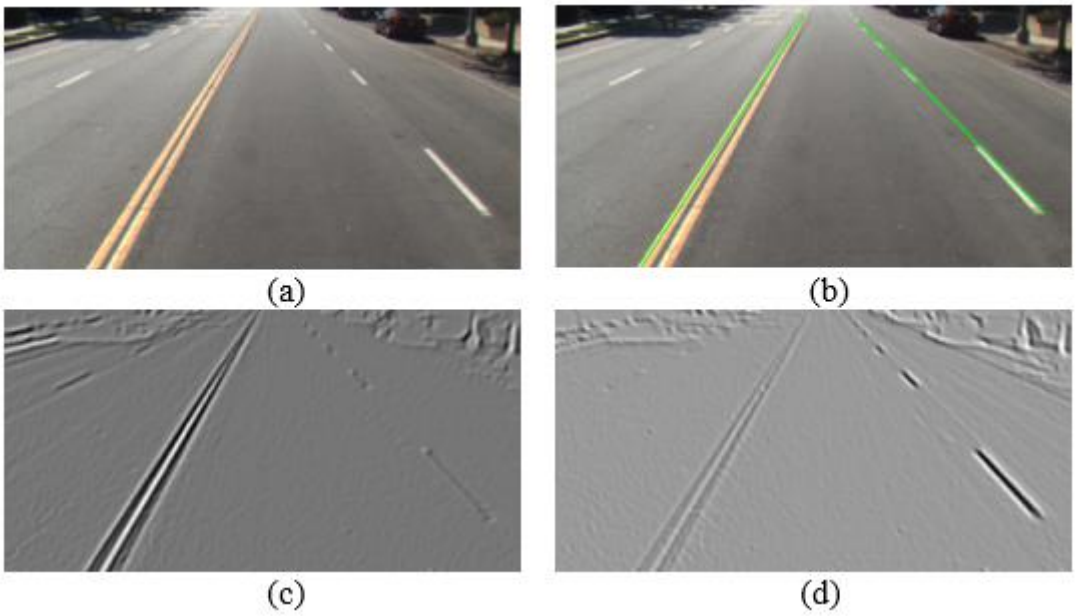

Figure 4-1. Lane detection with steerable filter.

In this work, two steerable filters, which detect left and right lane edge features, are applied individually. Figure 4-1. (c) and (d) represent the lane features given by steerable filter with two different orientation responses

3) Hough Transform

Hough Transform line detection has been successfully used for lane detection in previous studies and has been proved efficiency in lane detection. It was first designed to recognize lines [33], and then improved to detect arbitrary shapes [34]. It is an efficient line detection method, which can be applied on binary images.

To apply standard Hough Transform, lanes are assumed to be straight, which is true in most situations. In terms of curve lanes, lanes in the near field of the image can be viewed as a straight lane and can be detected with Hough Transform. Therefore, Hough Transform is used to detect lanes in the primary algorithm for fast and rough lane detection since no model fitting is required. The Hough Transform convert models in Cartesian coordinate into Polar coordinate, such that the straight-line model in equation (6) can be transformed to equation (7) or equation (8).

$$y = k \cdot x + b \tag{4-6}$$

$$\rho = x\cos\theta + y\sin\theta \tag{4-7}$$

$$y = -\frac{\cos\theta}{\sin\theta}x + \frac{\rho}{\sin\theta} \tag{4-8}$$

where $\theta \in [0°, 180°]$ and $\rho \in \boldsymbol{R}$, θ is the line orientation in Polar space, and ρ is the distance between the original point and the line in Polar space. k and b is the slope of the and the Y-intercept line in Cartesian space, respectively.

The process of the primary lane detection algorithm is illustrated in Table 4-1.

TABLE 4-1 FIRST LANE DETECTION PROCEDURE

Lane Detection Algorithm Based On Hough Transform
Input: Raw image
Outputs: straight lane parameters
Process:
1. Crop the raw image with pre-determined ROI.
2. Enhance colour contrast using decorrelation stretching.
3. Convert RGB image to grey scale image.
4. Perform 2nd order steerable filter to find lane edge map.
5. Using Hough Line Transform to detect left and right lanes.
6. **if** slopes and intercept meet lane geometry constraint
7. Lanes are found.
8. **else**
9. Lane detection error.
10. **end if**
11. Return output value.

Lines in the original image are transferred to single points in the Polar coordinate, and this point can be represented as a sinusoid line in Hough space. An accumulator array which contains the votes of the number of sinusoid Hough lines that pass through a point is used. The points with most Hough lines intersect are determined as a line in the edge

map. Figure 4-1, (b) shows the result of lane detection given by SF and HT. To make a fair illustration and comparison, all the illustration images in this paper are selected from Caltech public dataset [20].

4.4.2 Lane Detection Using GMM and RANSAC method

GMM and RANSAC are combined as the secondary lane detection system when the primary algorithm fails, or a low detection confidence is returned. Hough Transform based method has been proved efficiency for lane detection in a vast amount of the previous literature. However, it is difficult to process curve lanes. Meanwhile, in case to increase the diversity of the lane detection algorithms, a different image segmentation and model fitting based method, which integrates the GMM and RANSAC will be used as a secondary algorithm to enlarge the robustness of the system. The Hough Transform based algorithm can be viewed as a feature-based method, while the RANSAC method belongs to model-based algorithm. Therefore, by dynamically integrating the feature-based algorithm and model-based algorithm, the lane detection system will have the advantages given from these two different algorithms. In this part, GMM is firstly used to segment lane markings, and RANSAC is then applied to fit both straight and curve lane models based on the features given by GMM. A slightly smaller ROI than the one used in the first part is used to increase the detection accuracy and reduce the false detection rate of lane pixels.

1) GMM-based feature extraction

GMM is an unsupervised machine learning method, which can be used for data clustering and data mining [36]. In this part, GMM is used to cluster the image into different parts, such as road, lanes and shadows. In GMM, the distribution of the input data is a mixture of a set of Gaussian distribution as shown in the following:

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sum_k) \tag{4-9}$$

where x represents data point, K is the number of component, μ and $\sum$ are the mean and covariance parameter of multivariate Gaussian function. Each component follows a multivariate normal distribution format that can be described as:

$$N(x|\mu, \sum) = \frac{1}{(2\pi)^{d/2}\sum^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu)) \tag{4-10}$$

where $\pi_k$is the weight for each Gaussian component and their sum equals to one, and $0 \leq \pi_k \leq 1$.

The feature vector of GMM is a four-element vector that contains the RGB values and the Sobel magnitude value to minimize the non-edge noise.

$$x(k) = [R(u,v), G(u,v), B(u,v), S(u,v)] \tag{4-11}$$

where x is the feature vector of a cropped ROI image, k is the index of x. (u,v) is image coordinate.

The GMM-based feature extraction performance is examined by comparing GMM with Otsu's segmentation method and examine GMM with different number of cluster centres. Firstly, the road segmentation result is compared with Otsu's method [37]. Otsu's algorithm segments the image by searching an optimal threshold in the greyscale image and has been widely used in many researches. When it is difficult to segment the lane and road as shown in Figure 4-2. (a), Otsu's segmentation fails to extract the lane pixels, as shown in Figure 4-2. (b). However, GMM can efficiently remove the road texture noise and extract the true lane pixels, as illustrated in Figure 4-2. (c).

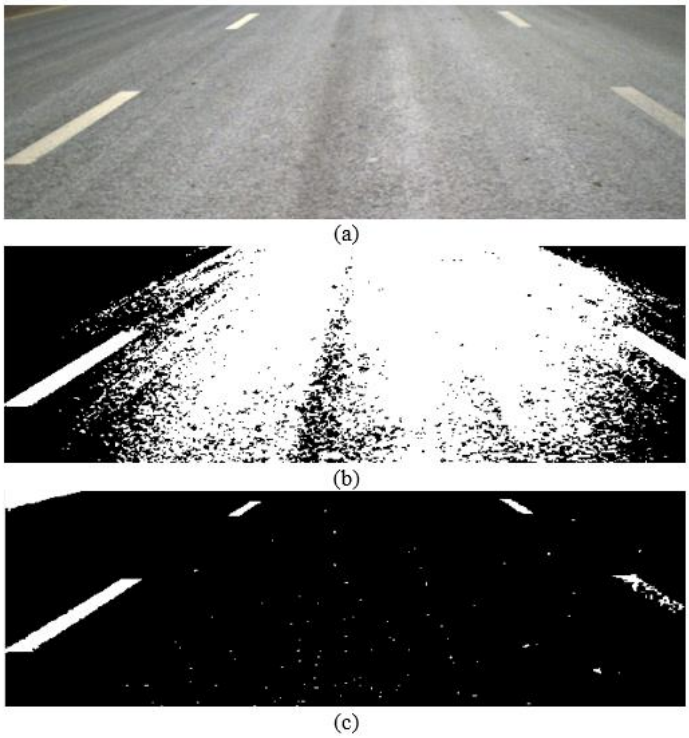

Figure 4-2. A segmentation comparison between Otsu's method and GMM in a complex texture road.

A GMM with two cluster centres sometimes clusters non-lane pixels into the lane group, such as the bright area shown in the top-left corner in Figure 4-2. (c). However, the effect of these lane outlier pixels can be removed by the following RANSAC algorithm. In addition, another powerful property of GMM-based lane segmentation is that shadows

can be removed by controlling the number of cluster centres. By increasing the number of clustering centres, GMM will generate to a more accurate segmentation result as shown in Figure 4-3. The subplot (c) and (e) are the segmentation results obtained with three and four cluster centres. Graph (d) and (f) are the corresponding binary images which only contain lane and non-lane pixels.

2) RANSAC model fitting

RANSAC algorithm is widely used in computer vision systems due to the good robustness property. RANSAC iteratively estimates the parameters of the given model by dividing data into outlier and inlier sets. A sample subset containing the minimal points to describe the model is chosen iteratively. Then, all the data are tested with the fitted model, and data items which are consistent with the model within a certain threshold are selected as inliers. RANSAC keeps fitting the given model unless the maximum iteration number is reached or a best model that has most inliers is found.

RANSAC algorithm is used to fit a Hyperbola-pair lane models in this work. Lane models are divided into near field and far field part. Near field model is a common straight-line model as shown in Equation 4-12, and a hyperbola model, which is given by equation (12) [16], is used to represent lanes in far field:

$$u = \frac{C}{v-h} + k \times (v - h) + vp \tag{4-12}$$

where (u, v) is the coordinate in image plane, C is the road curvature in far field, k is the slope of the straight line in near filed, h is the y-coordinate of road horizon, and vp denotes the estimated vanishing point of straight lanes.

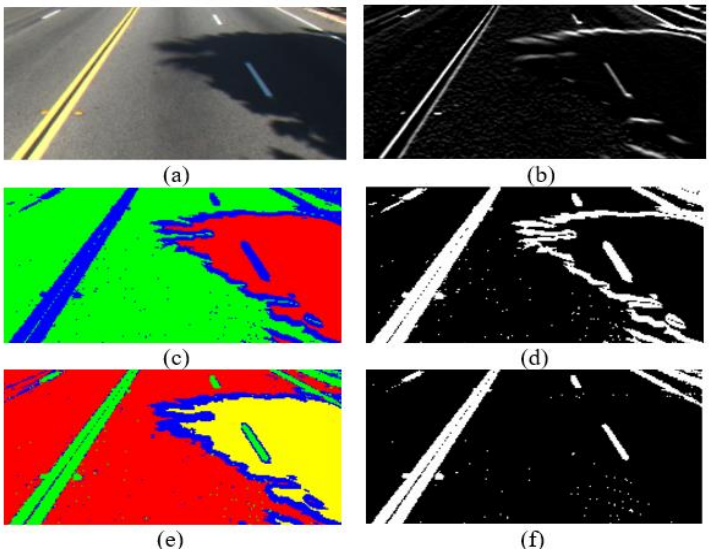

Figure 4-3. Results of lane segmentation using GMM on shadow road.

In Figure 4-3, (a) is raw RGB image, (b) is the magnitude image calculated with Sobel operator, (c) and (e) are GMM segmentation result with 3 and 4 clustering centres while (d) and (f) are the binary value images of (c) and e) respectively. As shown in this graph, the more clusters used, the more accurate lane extraction can be achieved.

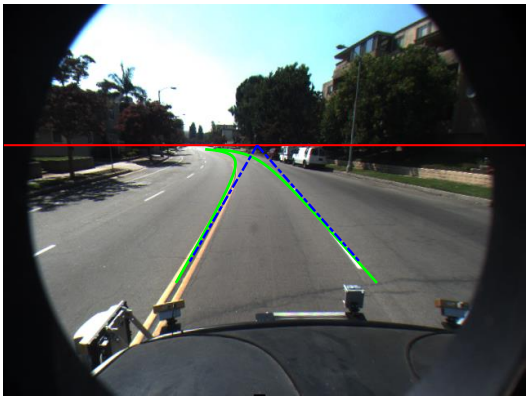

Figure 4-4. Caltech road lane detection result based on Hyperbola-pair model.

As shown in Figure 4-4, Red line represents the estimated horizon line in the image, blue line is the estimated straight line for both right and left line, the intersection of blue lines and red horizon line is the regarded as lane vanishing point, green lines are the estimated lanes for both left and right lanes following road curvature.

The left and right lane model are fitted using RANSAC, separately. Figure 4-4 shows an example of curve lane detection result based on GMM and RANSAC lane detection. Table 4-2 illustrates the lane detection procedure using the proposed GMM and RANSAC algorithms.

TABLE 4-2 SECOND LANE MODEL FITTING PROCEDURE

GMM and RANSAC Lane Model Fitting Algorithm

Input: ROI image

Outputs: lane curvatures C_l, C_r, straight line models of left and right lanes

Process:

1. Construct feature vectors based on the ROI image.
2. Fed into GMM clustering module.
3. Extracted lane features from GMM result and binarization.
4. Fit near field straight line model and calculate slopes of both left and right lane if no slopes exist.
5. **if** slopes and intercept meet lane geometry constraint
6. Fit far field curve line model using the slopes.
7. **else**
8. Lane model fitting error.
9. **end if**
10. Return output value.

4.4.3 Lane Tracking with Kalman Filter

Lanes are tracked with Kalman filter after a successful detection to smooth and improve the stability of the detection results. Kalman filter is used for many applications such as noise removal, generating non-observable states, and object navigation and tracking. It provides the optimal state estimation for the linear systems. Kalman filter involves two steps, prediction and update. The prediction step uses previous states to estimate the current states, along with their uncertainties. The update step uses the current measurements to correct the estimation results given by the prediction step using a weighted average which give more weights to the estimates that with high certainty. The prediction and update steps run in a recursive way and estimates the optimal states at each time step.

The Kalman filter can be simply described as the following process.

Firstly, based on the linear dynamic model, the Kalman filter model assumes the true states at time t have the following relationship with the states at t-1:

$$x_k = F_k x_{k-1} + B_k u_k + w_k \tag{4-13}$$

where

F_k is the state transition model.

B_k is the control-input model.

and w_k is the process noise with zero mean multivariate normal distribution N, with covariance Q_k.

The observation z_k at each time step is described as:

$$z_k = H_k x_k + v_k \tag{4-14}$$

where H_k represents the observation model, and v_k is the Gaussian noise model that exist in the measurement with the covariance of R_k.

The two-stages process can be described as follows.

At predict step, the state and error covariance are estimated,

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} B_k u_k \tag{4-15}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{4-16}$$

In the update stage, the system output will be corrected according to the estimated *a prior* state as well as the measurements.

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{4-17}$$

$$S_k = R_k + H_k P_{k|k-1} + H_k^T \tag{4-18}$$

$$K_k = P_{k|k-1}H_k^T S_k^{-1} \tag{4-19}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k \tag{4-20}$$

$$P_{k|k} = (I - K_k H_k)P_{k|k-1}(I - K_k H_k)^T + K_k R_k K_k^T \tag{4-21}$$

$$\tilde{y}_{k|k} = z_k - H_k\hat{x}_{k|k} \tag{4-22}$$

where $\tilde{y}_k$ is the innovation pre-fit residual, S_k is the innovation covariance, K_k represents the Kalman gain, $\hat{x}_{k|k}$ and $P_{k|k}$ are the estimated a posterior state and covariance, respectively. $\tilde{y}_{k|k}$ is the measurement post-fit residual.

In this part, the Kalman filter is used to track and smooth the detected lane parameters. The lane detection model is selected as a simple constant velocity model, which ignore the variation of the vehicle velocity. The discrete state-space model for Kalman filter is described as:

$$x[t] = Fx[t-1] + w(t-1) \tag{4-23}$$

$$y[t] = Hx[t] + \mathrm{v(t)} \tag{4-24}$$

The selection of the covariance matrices did not rely complex covariance matrix estimation algorithm such as the autocovariance least-square technique, instead, they are selected merely based on manually selection and the performance are simplify verified according to the visual evaluation on the public dataset and Cranfield local dataset. Kalman filter tracks the lane positions with a linearized system dynamic model and Gaussian noise to estimate the optimal states at each time step. In this work, the left and right lanes are tracked separately while using the same system model. For straight lane case, the detected line slope and intercept are tracked. For curve lane case, extra curvature parameters C_l, C_r and their variation are added. State vector $x(t)$ and output vector $y(t)$ of the straight lane model are shown as:

$$X = [k(t), \dot{k}(t), b(t), \dot{b}(t)]^T \tag{4-25}$$

$$Y = [k(t), b(t)]^T \tag{4-26}$$

where t denotes for t, k and b are the slope and intercept, $\dot{k}$ and $\dot{b}$ are the derivatives. The state transition matrix F and output matrix H are defined as:

$$F = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-27}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4\text{-}28)$$

4.4.4 Lane Sampling and Voting for Lane Recognition

In this part, a lane type recognition method based on lane sampling and voting (LSV) is proposed. Lane type contains two properties which are the colour and line style of the lanes. Three lane styles, namely the solid, double solid, and dashed ones, and two-lane colours, i.e. yellow and white, are recognized. A set of sampling points with same interval is selected according to the detected lanes. The two recognition steps are explained in detail as follows.

1) Lane Colour Detection

To detect lane colours of both the left and right lanes, the RGB images are first converted into the YCbCr colour space to increase the contrast between yellow and white colour. Also, converting raw images to YCbCr space improves the detection efficiency, since the two colours can be efficiently distinguished only with the Cb channel of the YCbCr images. The sub graph (a) and (b) in Figure 4-5 illustrate the image in YCbCr space and the sampling points on the lanes. Figure 4-6. shows the value of the sampling points in Cb space. A clear boundary between the yellow lane and the white lanes in the Cb image can be found. Then, Lane colour is determined according to the voting results of the sampling points. If the values of most sampling points are below the threshold, the lane is determined to be yellow, otherwise the colour is determined to be white.

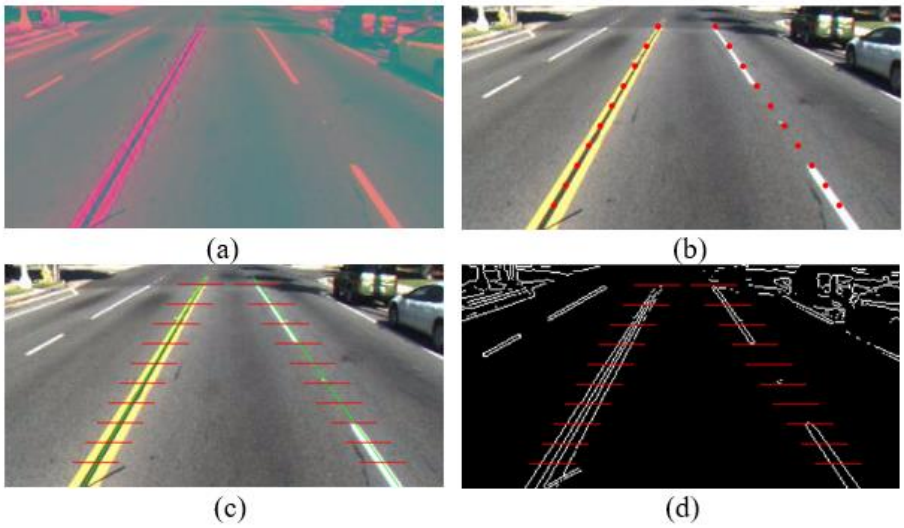

Figure 4-5. Illustration of lane sampling and voting.

In Figure 4-5, subgraph (a) shows traffic image in YCbCr colour space, (b) shows the sampling points on both lanes in original image. (c) and (d) are the sampling lines generated based on the sampling points in original and edge images, respectively.

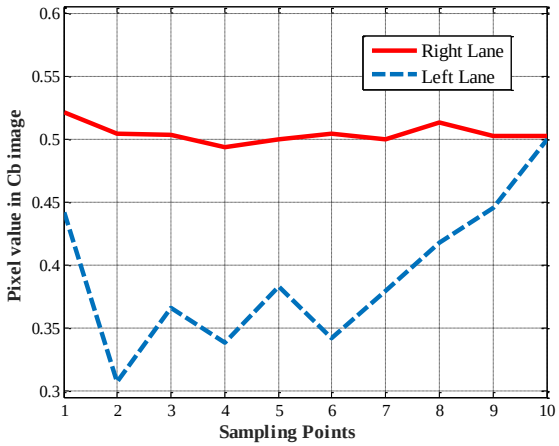

Figure 4-6. Illustration left and right lane sampling points in Cb image.

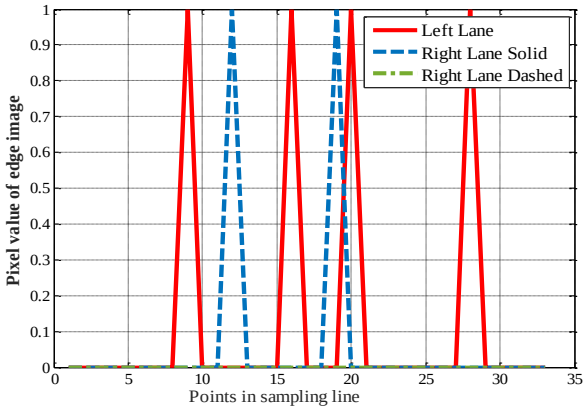

Figure 4-7. Sampling points values along sampling lines.

2) Lane Type Detection

After the lane sampling points have been determined, line style of left and right lanes can be recognized by extending the sampling points into short sampling lines. The line width of each sampling line is chosen as about three times of the original one to cover the double lanes. Besides, the width of the sampling lines can control the robustness of the style recognition algorithm. Sometimes the detected and tracked lanes will drift or become inaccurate, due to the disturbances on road, such as a nearby vehicle appears. A slightly longer sampling line can still cover and recognize the line style at this moment. The scanning is proposed on the edge image given by the Sobel edge detector. The lane style is determined by counting the rising edge along each sampling line. Figure 4-5. (c) and (d) illustrate the sampling lines in the raw and edge image. Figure 4-7. shows the value distribution of the sampling line points on edge image. As shown in Figure 4-7, the

double-solid left lane is expected to have four maximums. The dashed right lane consists of two parts, which are a short single-solid line segment and no line segment, respectively. The single-solid line part is expected to have two maximums while the empty segment should have zero values along the corresponding sampling line. The line style determination logic based on LSV method is shown in Table 4-3.

TABLE 4-3 LANE TYPE RECOGNITION PROCEDURE

Lane points sampling and voting for lane recognition

Input: ROI image, lanes position

Outputs: lane line style and colour

Process:

```
Convert RGB image into YCbCr colour space.
Edge extraction and binarization of RGB image.
Lane points sampling based on lanes position.
for Lines equal to left lane and right lane
    if more points value in Cb image larger than threshold
        Lane colour is determined as white.
    else
        Lane colour belongs to yellow.
    end if
end for
Generate scanning lines based on sampling points.
for Lines equal to left lane and right lane
    if most minimum value of sampling lines equals to zero
        Lane style belongs to dash line.
    else if most scanning lines values less than four
        Lane style belongs to solid line.
    else if most scanning lines values larger or equals to four
        Lane style belongs to double line
    else if all scanning lines values equals to zero
        Detection fails.
    end if
end for
Return output value.
```

4.5 Lane Algorithms Integration and Evaluation

In this section, the approach used to integrate the proposed algorithms is described in detail. As mentioned in Section II, algorithm integration methods in literatures can be divided into the parallel and serial methods. In this work, the two existing detection algorithms are combined in a dynamic manner to create a robust lane detection and evaluation system.

1) Integration of Lane Detection Algorithms

The main processing procedure follows a serial sequence. The GMM based model fitting algorithm is applied as redundancy algorithm to increase the robustness of lane detection in a dynamic way. Instead of running multiple algorithms simultaneously, the GMM based detection system will only be activated when the first algorithm reports a low detection confidence. The real time evaluation system combines prior lane geometry

knowledge and the proposed LSV method. Figure 4-8 illustrates the architecture of the proposed integration and evaluation system.

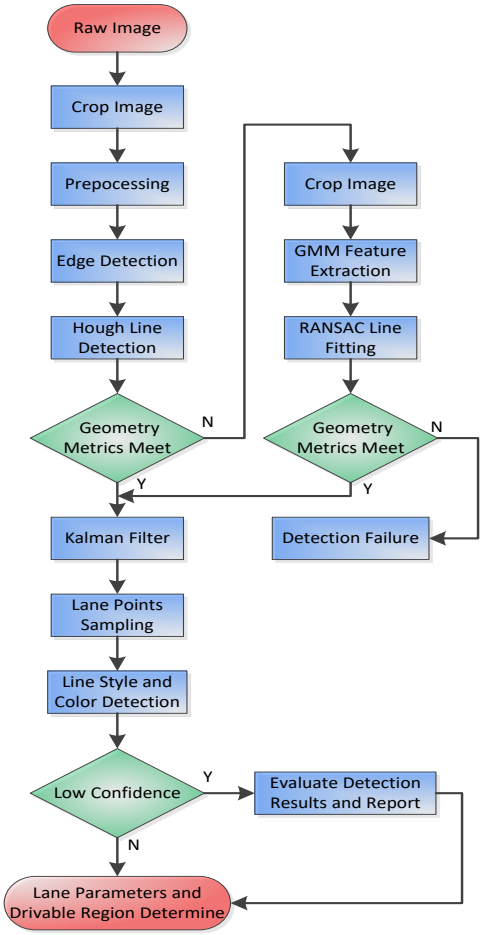

Figure 4-8. Integrated lane detection system architecture.

Hough line detection is used as the primary lane detection method, and GMM and RANSAC method is used as a redundancy one which is only activated in certain situations. There are two situations when the secondary lane detection system needs to be activated. The first case is that Hough line detection continuously fails to meet the road geometry constraints in a period. To reduce the impact of noisy measurement, the secondary lane detection system is initiated, and the detection results are tracked with Kalman Filter. The other condition to active the secondary algorithm is when the LSV system detects a low confidence of current measurement. At this moment, the detection results given by the two-lane detection systems are evaluated together. If straight line parameters are within a vicinity, then the detection result is adopted, otherwise the low detection confidence is

reported to the user, and current tracked results are maintained. The HT based lane detection system will be re-initiated if the estimated curvatures are small enough.

2) Integration of Lane Detection Algorithms

The online evaluation system combines the prior lane geometry constraints and the lane sampling method. In this study, since inverse perspective mapping is not used in this work, lanes cannot be assumed to be parallel. However, the left and right lanes still meet some geometry constraints. For example, normally the detected slope of the left and right lanes with the straight-line model are opposite to each other and their summation is around zero. In this work, the prior lane position knowledge is used as the first evaluation metric. The prior knowledge contains the reasonable line slopes and the intercepts of both lanes. These parameters can be determined after mounting the front-face camera in the vehicle. If the camera does not change its focal length or other parameters automatically, the adjacent lanes are expected to be appeared in a certain range of locations. When the detected line parameters locate in a reasonable interval (i.e. $[\frac{1}{3}k, 3k]$ and $[\frac{1}{3}b, 3b]$ in this work), the detection results can be seen as reasonable. The reasonable interval is not necessary to be very restricted, because the results will be further filtered with Kalman filter. When the primary lane detection continuously fails within a period, the secondary algorithm will be activated. Similar limitation is also applied to the second algorithm. Another assumption is that the reasonable road curvature should be less than 90 degrees and the two lanes' curvatures are in the same direction. A fail detection is marked when the primary detection system continuously fails, and the secondary algorithm returns unreasonable results.

The second evaluation method is based on the proposed LSV method. A set of points will be sampled along each lane and acts as virtual sensors. As mentioned in Section III part D, LSV is used to recognize the lane types. In this part, this method is used to evaluate the lane detection system. The idea is that all the points should locate on the real lanes ideally. If the tracked lanes drift due to false detections or road curvature changes, the sampling points will no longer locate on the lanes anymore.

In this part, both the sampling points and the extended scanning lines are used to evaluate the detection result. The distance between two parallel points on the adjacent lanes is normally within a certain range. This distance is used to meet the ego-lane constant width constraints. When the distance is out of range, a low confidence is reported.

As mentioned before, scanning lines proposed on the edge image are used to detect the lanes. If detected lanes drift a little from the real position, the scanning is still able to find the peak values. However, if the detected results are far away from the real position, there will be no peak values along the line and the values of each points in the line equals to zero, as the bottom green dashed line shown in Figure 4-7. At this time, a low confidence of the current detection will be reported, and the secondary algorithm will be initiated. Figure 4-9 shows the tracked lanes positions given by Kalman filter (the green lines) drifted when facing curve road, and the sampling points fail to detect the lanes, especially the left lane. Thus, GMM-RANSAC is activated to detect lanes by fitting a curve lane model, as the red lines shows in Figure 4-9.

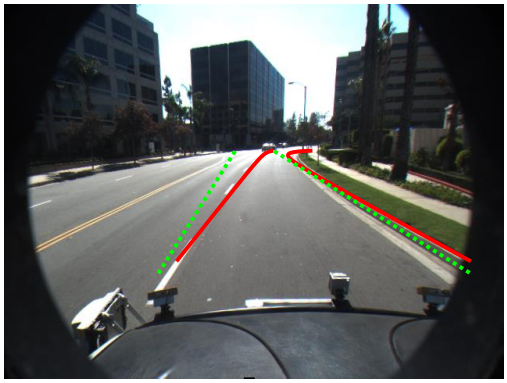

Figure 4-9. Compensation of the secondary lane detection algorithm.

The green dashed line is a false detection given by the HT detection and tracking system, and the red solid line shows the detection results given by the secondary GMM based lane detection algorithm. Although road curb in the right effect the accuracy of right lane, the red lines maintain a correct prediction of road curve. A low confidence is reported if a certain number or all the points vote. The threshold and the number of sampling points can be adjusted to control the sensitivity of the evaluation system. When the evaluation system reports a low confidence, the detection results given by different algorithms will be compared. If the result given by the secondary algorithm meets the lane geometry constraints, the detection results will be updated, and tracking will be kept until the road can be modelled as a straight line.

4.6 Experiment Results

To make a fair comparison and evaluation, in this section, our algorithm is compared with the one reported in [17] using Caltech public dataset. There are four clips of sequences in the dataset with 1224 manual labelled in total.

TABLE 4-4 DETECTION RESULT USING ALGORITHM IN [17]

Clip	Frames	Total	Detected	Correct Rate	False Pos. Rate
1	250	466	467	97.12%	3.00%
2	406	472	631	96.16%	38.38%
3	336	639	645	96.70%	4.72%
4	232	452	440	95.13%	2.21%
Total	1224	2026	2183	96.34%	11.57%

The proposed algorithms are evaluated using the ground truth lanes in the dataset and the same lane similarity measurement method as described in [17]. Lane detection system based on HT only and the integrated method based on HT and GMM are evaluated separately. Unlike previous research, the road curbs are regarded as a lane boundary in this work. This is a reasonable assumption, which has been discussed in [17]. In addition, it can be found in the dataset that sometimes lanes are not labelled when the lanes are close to the road curb. Therefore, the detected lanes using the proposed algorithm are with more numbers than that of the ground truth as shown in Table 4-4. For example, the quantity of our detected lanes in clip two is significantly larger than the given number in Table 4-4. The reason is that some lanes are not given a ground truth values because they are divided into road curbs.

In this work, a lane tracking after the detection process to keep tracking the lane position and minimize the influence of noisy detection. This is another reason that our system detects more lanes than previous method. By using the tracking algorithm, the lanes are tracked when there are no detected lanes, such as at intersections.

TABLE 4-5 DETECTION RESULT OF PROPOSED METHODS

Clip	Total	Detected	HT only		HT + GMM	
			Correct Rate	False Pos. Rate	Correct Rate	False Pos. Rate
1	466	492	94.85%	15.45%	98.50%	7.98%
2	472	806	92.16%	74.2%	96.19%	70.40%
3	639	669	93.15%	9.54%	97.18%	7.51%
4	452	459	94.47%	7.08%	98.01%	3.54%
Total	2026	2434	94.37%	25.77%	97.58%	22.56%

Table 4-5 shows the performance of the proposed integrated algorithm. The results given by the primary algorithm with tracking achieve comparable performance with previous research. When applying the redundancy algorithm, the correct rate increases, which shows that the secondary algorithm can compensate the primary algorithm. The

general performance of proposed algorithm achieves a high precision in both Caltech road dataset and the local one. The main issue of the result is that a high value of false positive detection rate occurs in second clip. As explained before, this is because some lanes, which are close to the curb, are not labelled and curbs are not regarded as a lane in the dataset. However, a curb recognition is as important as lane detection. Therefore, this high value of the false positive is not a big problem in real world application.

4.6.1 Experiment Results

Some result samples of the proposed integrated lane detection system are shown in Figure 4-10 and Figure 4-11. Figure 4-10 shows some successful detections according to Caltech and Cranfiled datasets. Caltech dataset was captured in urban area, while Cranfiled dataset was captured in both urban and highway roads. To make a fair illustration of the proposed system, we majorly adopt Caltech images for verification. As shown in Figure 4-10 the integrated lane detection system is robust to shadows, surrounding vehicles, curvy roads, intersections, and other road markings. The system can successfully detect lanes with large shadows on the road and is robust to non-lane road markings. Besides, with the redundancy algorithm and tracking algorithm, the system can successfully detect lanes with poor visualization, for example when the camera is facing towards sun. However, this monocular vision-based algorithm still has its limitation under some conditions.

Figure 4-10. Detection samples show the robustness and accuracy under different conditions.

Figure 4-11. Inaccurate and false detection samples showing the influence of curbs, no or poor lane marks, lane changing and model weakness.

Figure 4-11 shows some samples when the algorithm cannot find the lanes correctly. The first case shows the drawback of the curvy road detection algorithm. Hyperbola model is a second order model, which is suitable for most curvy lane conditions. However, it cannot give a precise estimation of very complex lanes comparted with higher order models. There always exist a trade-off between computational efficiency and accuracy. Most curvy lanes in real world, especially on highway, can be modelled with hyperbola model in a relatively faster fitting time. Therefore, in this work the second order hyperbola model is chosen to fit curvy lanes. The second case shows that lane detection would fail when there are no clear lane markings because of the reflection of light or shadows. But at these situations, it is also very difficult to distinguish the lane markings even with human eyes. Therefore, both of the two detection algorithms fails to find the lane. The third case shows that no lane marking is next to the road boundary and the detection and tracking system choose the guardrail as the lane boundary. Guardrail and lane marking distinction is one of the most common and difficult tasks, because guardrail has similar properties with lanes from the perspective of computer vision. In this case, the distance is beyond the pre-determined range and secondary algorithm is imitated to find the lane boundaries. The last two case shows that no lane marking is in the middle of the road and the detection system finds two wider lanes. Since there is no clear ego-lane marking, the secondary algorithm cannot find a correct lane position.

4.7 Discussion

The proposed two algorithms in this work are expected to enrich the system diversity. A robust straight lane detection and tracking system is used as the primary algorithm under the assumption that in most situations lanes can described with straight line model. This is reasonable in real life since curvy roads doesn’t appear frequently in most of the situations. Besides, in the near field part of curves, lanes can be modelled as straight ones.

Moreover, this method is computational efficiency and can be easily implemented to the on-board control system. The GMM used in the secondary detection algorithm has been successfully applied to many image segmentation tasks. As shown in section III part B, GMM can distinguish the road, shadow and lane markings accurately. Features extracted by GMM carry more information, which not only give accurate estimations of lane markings but are also able to segment the road and shadows. Another property of GMM and RANSAC model fitting method is that it can find lane marks with very low image resolution due to the robustness characteristics. The proposed algorithm is also tested with images in the format of 60×80 pixels, and the algorithm is still able to find lanes in most situations. However, colour-based image segmentation can be affected by those objects that have similar colour distribution. In that case, the extracted lane mark pixels can be noisy. Therefore, a smaller ROI image is used in this work to reduce the noise influence.

Lane type recognition is proposed based on the sampling points and lines. Currently, the lane type is estimated frame by frame, which increases the computation burden. In real world applications, lane type is not necessary to be detected at each step. A more efficient way is to combine a lane departure or lane change aware system and only initiate lane type recognition after a lane change manoeuvre.

Previous lane detection evaluation method often uses visual check without quantity measurement. This is mainly because ground truth labelling is a tough work that costs many human resources. Another reason that makes lane detection system difficult to be evaluated that is no common metric exists in the literatures. This give rise to the utilisation of different testing images and ROI. Therefore, an adaptive lane detection system, which can work under various application situation without or with only a few modifications, is expected to be developed. For example, an automatic lane marking recognition can be combined with a road area detection system. There are many approaches to achieve an adaptive lane detection system, and with such a system, lane detection will be much easier to be evaluated and implemented.

4.8 Conclusions

In this work, a novel real-time integration and evaluation architecture towards a robust lane detection system is proposed. Robust lane detection is achieved by increasing the diversity of detection algorithms. Two distinct lane detection algorithms which based on Hough Transform and GMM-RANSAC model fitting methods are integrated in an

efficient way. The detected lane positions are further tracked and smoothed with Kalman filter. Instead of detecting all the lane markings on the road, the boundaries of ego-lanes with corresponding lane type and drivable area recognition is constructed. This will reduce the computation cost compared with the methods that detect all the candidate lanes in the image. Finally, a real time lane detection evaluation method is constructed based on the combination of prior lane geometry information and LSV scheme. The contribution of this paper is as follows:

GMM based lane feature extraction is proposed and analysed in detail. The results show that with more cluster centres, GMM can classify road, lane marks, shadow and other objects accurately.

A novel efficient integration method to combine two lane detection algorithms is designed. Unlike previous parallel architecture, the secondary algorithm is only activated when the primary algorithm continuous fails or a low detection confidence occurs.

A novel online evaluation system based on prior lane geometry constraints and sampling and voting scheme is proposed. The system continuous evaluate the lane detection result and try to keep the detection as accurate as possible.

The lane detection and evaluation system designed in this work use monocular camera and algorithm level integration achieve an accurate and robust detection performance in real world. However, there are still some limitation exist in both hardware and software aspects since camera cannot deal with all the kinds of serious situations. Future works will combine lane detection system with other object detection system as mentioned in Section II to increase the robustness of the lane detection system. Besides, more works is expected to be done towards an efficient real time lane detection evaluation system

4.9 References

[1] National Highway Traffic Safety Administration. "National motor vehicle crash causation survey: Report to congress." National Highway Traffic Safety Administration. "National motor vehicle crash causation survey: Report to congress." *National Highway Traffic Safety Administration Technical Report DOT HS* 811 (2008): 059.

[2] Martinez, Clara Marina, Xiaosong Hu, Dongpu Cao, Efstathios Velenis, Bo Gao, and Matthias Wellers. "Energy Management in Plug-in Hybrid Electric Vehicles: Recent Progress and a Connected Vehicles Perspective." IEEE Transactions on Vehicular Technology (2016).

[3] Son, Young Seop, *et al*. "Robust multirate control scheme with predictive virtual lanes for lane-keeping system of autonomous highway driving." *IEEE Transactions on Vehicular Technology* 64.8 (2015): 3378-3391.

[4] Joerer, Stefan, *et al*. "A vehicular networking perspective on estimating vehicle collision probability at intersections." *IEEE Transactions on Vehicular Technology* 63.4 (2014): 1802-1812.

[5] Qu, T., Chen, H., Cao, D., Guo, H., Gao, B. Switching-based stochastic model predictive control approach for modeling driver steering skill. IEEE Transactions on Intelligent Transportation Systems, 16(1): 365-375, 2015.

[6] Lv, Chen, Junzhi Zhang, and Yutong Li. "Extended-Kalman-filter-based regenerative and friction blended braking control for electric vehicle equipped with axle motor considering damping and elastic properties of electric powertrain." *Vehicle System Dynamics* 52.11 (2014): 1372-1388.

[7] Bar Hillel, Aharon, *et al*. "Recent progress in road and lane detection: a survey." *Machine vision and applications* (2014): 1-19.

[8] McCall, Joel C., and Mohan M. Trivedi. "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation." *IEEE transactions on intelligent transportation systems* 7.1 (2006): 20-37.
[9] Yenikaya, Sibel, Gökhan Yenikaya, and Ekrem Düven. "Keeping the vehicle on the road: A survey on on-road lane detection systems." *ACM Computing Surveys (CSUR)* 46.1 (2013): 2.
[10] Xiao, Jing, Shutao Li, and Bin Sun. "A Real-Time System for Lane Detection Based on FPGA and DSP." *Sensing and Imaging* 17.1 (2016): 1-13.
[11] Suddamalla, Upendra, *et al*. "A novel algorithm of lane detection addressing varied scenarios of curved and dashed lanemarks." *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*. IEEE, 2015.
[12] de Paula, Maurício Braga, and Cláudio Rosito Jung. "Automatic detection and classification of road Lane markings using onboard vehicular cameras." *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015): 3160-3169.
[13] Cela, Andrés F., *et al*. "Lanes Detection Based on Unsupervised and Adaptive Classifier." Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on. IEEE, 2013.
[14] Bounini, Farid, *et al*. "Autonomous Vehicle and Real Time Road Lanes Detection and Tracking." *Vehicle Power and Propulsion Conference (VPPC), 2015 IEEE*. IEEE, 2015.
[15] Suddamalla, Upendra, *et al*. "A novel algorithm of lane detection addressing varied scenarios of curved and dashed lanemarks." *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*. IEEE, 2015.
[16] Tan, Huachun, *et al*. "Improved river flow and random sample consensus for curve lane detection." *Advances in Mechanical Engineering* 7.7 (2015): 1687814015593866.
[17] Wang, Yue, Eam Khwang Teoh, and Dinggang Shen. "Lane detection and tracking using B-Snake." *Image and Vision computing* 22.4 (2004): 269-280.
[18] Lim, King Hann, Kah Phooi Seng, and Li-Minn Ang. "River flow lane detection and Kalman filtering-based B-spline lane tracking." *International Journal of Vehicular Technology* 2012 (2012).
[19] Li, Jun, Xue Mei, and Danil Prokhorov. "Deep neural network for structural prediction and lane detection in traffic scene." *IEEE transactions on neural networks and learning systems* (2016).
[20] Aly, Mohamed. "Real time detection of lane markers in urban streets." *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.
[21] Borkar, Amol, Monson Hayes, and Mark T. Smith. "A novel lane detection system with efficient ground truth generation." *IEEE Transactions on Intelligent Transportation Systems* 13.1 (2012): 365-374.
[22] Shin, Bok-Suk, Junli Tao, and Reinhard Klette. "A superparticle filter for lane detection." *Pattern Recognition* 48.11 (2015): 3333-3345.
[23] Labayrade, Raphaël, *et al*. "A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance." *IEICE transactions on information and systems* 89.7 (2006): 2092-2100.
[24] Sivaraman, Sayanan, and Mohan Manubhai Trivedi. "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach." *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013): 906-917.
[25] Nguyen, Vinh Dinh, *et al*. "A fast evolutionary algorithm for real-time vehicle detection." *IEEE Transactions on Vehicular Technology* 62.6 (2013): 2453-2468.
[26] Fritsch, Jannik, Tobias Kuhnl, and Andreas Geiger. "A new performance measure and evaluation benchmark for road detection algorithms." *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013.
[27] Rose, Christopher, *et al*. "An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS." *IEEE Transactions on Intelligent Transportation Systems* 15.6 (2014): 2615-2629.
[28] Danescu, Radu, and Sergiu Nedevschi. "Probabilistic lane tracking in difficult road scenarios using stereovision." *IEEE Transactions on Intelligent Transportation Systems* 10.2 (2009): 272-282.
[29] Li, Qingquan, *et al*. "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios." *IEEE Transactions on Vehicular Technology* 63.2 (2014): 540-555.
[30] Cui, Dixiao, Jianru Xue, and Nanning Zheng. "Real-Time Global Localization of Robotic Cars in Lane Level via Lane Marking Detection and Shape Registration." *IEEE Transactions on Intelligent Transportation Systems* 17.4 (2016): 1039-1050.
[31] Lin, Chun-Wei, Han-Ying Wang, and Din-Chang Tseng. "A robust lane detection and verification method for intelligent vehicles." *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*. Vol. 1. IEEE, 2009.
[32] Satzoda, Ravi Kumar, and Mohan M. Trivedi. "On performance evaluation metrics for lane estimation." *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014.
[33] VC, Hough Paul. "Method and means for recognizing complex patterns." U.S. Patent No. 3,069,654. 18 Dec. 1962.
[34] Ballard, Dana H. "Generalizing the Hough transform to detect arbitrary shapes." *Pattern recognition* 13.2 (1981): 111-122.
[35] Freeman, William T., and Edward H. Adelson. "The design and use of steerable filters." *IEEE Transactions on Pattern analysis and machine intelligence* 13.9 (1991): 891-906.
[36] Xu, Lei, and Michael I. Jordan. "On convergence properties of the EM algorithm for Gaussian mixtures." *Neural computation* 8.1 (1996): 129-151.
[37] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." *Automatica* 11.285-296 (1975): 23-27.

PART IV:

DRIVER BEHAVIOUR REASONING.

Driving actions and secondary tasks recognition

5

PAPER IV – Driver Behaviour Recognition with Feature Evaluation

Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition

Authors:

Yang Xing, Chen Lv, Zhaozhong Zhang, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang

This paper has been published by

IEEE Transactions on Computational Social Systems (2018)

Abstract

Driver decisions and behaviours regarding the surrounding traffic are critical to traffic safety. It is important for an intelligent vehicle to understand driver behaviour and assist in driving tasks according to their status. In this study, the consumer range camera Kinect is used to monitor drivers and identify driving tasks in a real vehicle. Specifically, seven common tasks performed by multiple drivers during driving are identified in this study. The tasks include normal driving, left, right, and rear mirror-checking, mobile phone answering, texting using a mobile phone with one or both hands, and the setup of in-vehicle video devices. The first four tasks are considered safe driving tasks while the other three tasks are regarded as dangerous and distracting tasks. The driver behaviour signals collected from the Kinect consist of a colour and depth image of the driver inside the vehicle cabin. Additionally, three-dimensional head rotation angles and the upper body (hand and arm at both sides) joint positions are recorded. Then, the importance of these features to behaviour recognition is evaluated using Random Forests (RF) and Maximal Information Coefficient (MIC) methods. Next, a Feedforward Neural Network (FFNN) is used to identify the seven tasks. Finally, the model performance for task recognition is evaluated with different features (body only, head only, and combined). The final detection result for the seven driving tasks among five participants achieved an average of greater than 80% accuracy, and the FFNN tasks detector is proved to be an efficient model that can be implemented for real-time driver distraction and dangerous behaviour recognition.

Index Terms - Driver Behaviour, driver distraction, Kinect, Random Forest, Feedforward Neural Network.

5.1 Introduction

5.1.1 Motivation

Driver behaviour is the most important factor for on-road driving safety [1]-[6]. Since humans are the major users of roads, their driving behaviours influence traffic safety and efficiency. More than 90% of traffic accidents for light-vehicles in the US were reported to be caused by driver errors such as misbehaviour and inadvertent errors, which is similar to other countries worldwide. It was also mentioned in [7]-[11] that traffic accidents could be reduced by 10% to 20% by correctly recognizing driver behaviours. Therefore, it is critical to have a clear perspective of driver behaviour and the tasks being performed.

Human drivers have been extensively studied since the 1970s. The study of human drivers is a massive project with many aspects. Most of the existing research lies in the scopes of driver behaviours, driver attention and intention, driver drowsiness and fatigue, driver cognitive and neural muscles, etc. All of these studies have a common objective, which is to gain a better understanding of driver status from either a psychological or physiological aspect so as to assist in driving tasks and increase driving safety [12]-[14].

Understanding human drivers is necessary both for conventional vehicles and for automated vehicles. In the US and China, accidents have occurred when a Tesla driver trusted or solely relied on the autopilot system while driving. For lower level automated vehicles, especially for level two and level three automated vehicles (based on the automation definition in SAE standard J3016), human drivers need to sit in the driver seat and are responsible for the safety issues. In these vehicles, the driver is allowed to perform secondary tasks for entertainment; however, due to the partially automated limitation, the driver has to take control in emergencies. Therefore, the monitoring of human drivers and determining whether they can return to the driving task is more important than in conventional vehicles.

In this study, a driver monitoring system is designed to detect driving and secondary tasks in real time. Specifically, the recognition model is designed to identify seven tasks performed by different drivers. There are four tasks considered as normal driving tasks: normal driving (front looking), right mirror checking, left mirror checking, and rear mirror checking. Meanwhile, according to [13], the three most common secondary tasks in automated vehicles are selected, which are using a video device, answering a mobile phone, and texting using a mobile phone. To identify the driver postures, multimodal data

is collected using a Kinect consumer RGB-D camera including the head rotation and body joint positions. The main objective of this study is to design a real-time driver behaviour model that does not require any history information for the recognition of normal driving and secondary driving tasks. Additionally, the importance of driver posture features to the identification of driving tasks is evaluated.

5.1.2 Related Works

In this study, the research scope is narrowed to the range of driving task recognition towards a normal driving and secondary task monitoring system for lower level and middle level automated vehicles. According to previous studies, driver behaviour can be classified into intended and non-intended behaviours [15][45]. The intended behaviour of the driver is the extension of the driver's mental thought, which can be used to infer the mental state and intent of the driver. In contrast, non-intended behaviours are usually caused by distractions due to outside and inside disturbances. Driver behaviour has been widely studied in previous literature. General driver behaviours include the study of driver head pose [15][16], eye gaze dynamics [17][18], hand motions and gestures [19], body movement [20][21], and foot dynamics [22]. This behaviour information has been successfully used to estimate driver fatigue, driver distraction, driver attention, etc. In this study, driver head and upper body information detected using a Kinect will be evaluated for normal driving and distraction identification.

When drivers are performing secondary tasks while driving, they are regarded as being distracted and many studies use the duration of eye-off-road to detect whether a driver is distracted by the secondary tasks. Therefore, the most common features for driver distraction detection are head pose and eye gaze information. Along with the driver behaviour, information of the vehicle such as vehicle speed, heading, and acceleration are important features for evaluating the level of driver distraction. In [23], an integration method combining the driver's hand, head, and eye for driver activity recognition was proposed. Rezaei and Klette introduced an intelligent driver assistance system to prevent rear-end crashes based on driver monitoring and front vehicle detection [24]. The head pose was estimated based on the proposed face appearance model and 3D head model mapping. In [25], a driver drowsiness alert system was proposed according to the driver head and eye dynamics. The driver head pose was estimated based on an Euler angle comparison between a single head region image and a 3D head model with known

rotations. In [28], the authors analysed the relationship between head pose and eye gaze. A strong correlation was found between the head and gaze direction. The study showed that during natural driving, the participants tend to have less head rotation but more gaze searching to maintain safe driving.

In [45], a comprehensive in-vehicle perception system for driver surveillance and assistance was proposed. Multi-modal sensors were fused to integrate the major driver physical cues and traffic situations. In [29], driver acceleration profiles for a car following scenario on a highway were generated using recurrent neural networks. Specifically, a LSTM recurrent network was adapted since it can automatically learn the spatial and temporal features of the naturalistic driving data. In [30], a LSTM based recurrent neural network was proposed to detect driver distraction behaviours based on the simulated CAN bus signals. Thirty participants performed eight typical secondary tasks independently and the distraction levels were classified into binary, three levels, and six levels. In [31], the authors claimed that applying eye tracking is much more difficult in real vehicles than in the simulator. Therefore, gaze estimation was not adopted and only driving information through the CAN bus was used for driver visual searching distraction detection. In [32], a driving behaviour model for teenage drivers was studied. Different machine learning methods were evaluated based on the driving data, which was collected with a driving simulator. The authors reported that instead of predicting driving behaviour (steer, throttle, and brake) directly, more accurate results can be achieved using context-based prediction and indirect prediction methods.

Despite the driver and driving behaviours, other studies have used physiological sensors to identify driver distraction and other abnormal statuses. According to the study in [35], driver monitoring systems for drowsiness and distraction detection can be classified into visual-based and non-visual-based methods. Visual-based methods monitor driver head pose, eye movement and blinking, yawning, and facial expression. In contrast, non-visual based systems detect driver status with physiological sensors such as EEG, ECG, and EOG, along with the vehicle CAN bus signals. However, the effects of hand, arm, and body on the recognition of driver status were not discussed. Similarly, a stress detection system for drivers was studied in [34]. A specific type of continuous recurrent neural network named cellular neural network was used for the binary classification task.

The Kinect sensor, a low-cost range camera, has been successfully applied to human and driver behaviour detection since it was first made available by Microsoft in 2012. Kinect was first designed for indoor motion sensing and provides a colour image, depth image, and infrared image. In [36], the general architecture for human activity recognition was proposed using Kinect. The human activities were viewed as the spatiotemporal evolution of body postures. The estimated postures are classified using support vector machines, and finally, the HMM was used to model the activities as a time sequence of the different estimated postures. In [37], a Kinect-based wearable face recognition system for people with low-vision or blindness was proposed. The colour and depth images were simultaneously captured to identify the face and generate the 3D location for the user. In [27], a seven-point skeleton-based driver upper body tracking system using Kinect depth images was applied. The proposed system is efficient for detecting driver merging and turning behaviours according to the detected body pose and arm motion. The system can also be used to analyse and compare the driving manoeuvre styles of different drivers.

In this study, Kinect is adopted as the driver monitoring sensor to identify normal driving and secondary tasks. Similar research can be found in [26], where driver mirror-checking behaviour during normal driving and performing secondary tasks were analysed. The authors reported that mirror-checking behaviour is one of the most important driving perception processes and reflects the attention level of the driver. In addition, mirror-checking behaviours are highly detectable manoeuvres and can achieve 95% detection accuracy using machine learning. However, that work only studied the binary classification scenarios without reporting the recognition accuracy for each task. Additionally, that study did not analyse the impact of body postures to the recognition of complex driving behaviours.

5.1.3 Contribution

In this study, driver head and body posture information is used for driving and non-driving related task recognition. The contributions of this study are threefold.

First, a driver posture detection method using a Kinect, a consumer range camera, inside a vehicle is introduced. The data characteristics of Kinect are analysed and the data processing technique for in-vehicle application is proposed. In addition, the head rotation signals from the Kinect are calibrated with a precise orientation sensor.

Second, the importance of head and body features to task prediction is estimated using an integrated algorithm. The feature importance estimation given by RF and MIC are compared and integrated. Then, the most important posture features for task recognition are determined. Unlike previous studies that use time sequence data for driver behaviour recognition [29, 30], this study focuses on identifying behaviour in a more natural way, only based on the instance samples. The objective is to design a human-like task detector that can identify driver behaviours according to a single image. Therefore, a FFNN model is evaluated and compared with multiple machine learning methods.

Finally, quantity analyses of the impact of the driver's head and body features to driver task recognition are performed, and the predicted important posture features are evaluated separately. Since the existing literature seldom considers the driver's body features, this study quantitatively proved that head and body features are required for driver behaviour recognition.

5.2 Experiment Design and Data Analysis

5.2.1 System Architecture

The procedure taken to construct a behaviour recognition model is described in this section. The driver monitoring system architecture is shown in Figure 5-1. The general structure of this study consists of three parts. First, the driver head and body data are collected and time stamped. Then, the signals are smoothed, and noise is filtered. Second, feature importance prediction is proposed using a combination of Random Forests and MIC, the feature importance given by the two algorithms show strong consistency. The 'model selection' block in Figure 5-1 processes the feature evaluation based on the feature importance provided by the 'feature importance estimation' block. Meanwhile, the influence of depth, head, and body features to the driver status detection will be studied. Then, real-time driver behaviour identification will be conducted using a FFNN model with *leave-one-out (LOO)* cross-validation. Finally, the performance with different features is analysed and a behaviour classification performance comparison between different algorithms will be proposed.

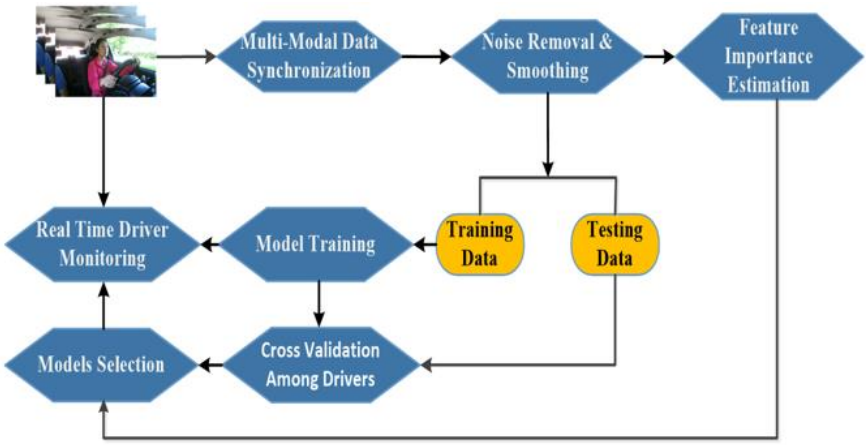

Figure 5-1. Proposed driver task recognition architecture.

5.2.2 Experiment Setup and Data Collection

In this part, the experiment setup and data collection methods are introduced. Driver behaviour data is collected using the low-cost range camera Kinect, which was developed by Microsoft. In this study, the second version of Kinect (V2) was adopted. Kinect is a consumer camera that supports colour images, depth images, audio, and infrared information. It was first designed for indoor human interaction with computers and has been successfully applied in vehicles for driver monitoring [36][37].

Kinect supports tracking the head and the body skeletons of as many as six individuals. In this work, the head and upper body joint detection functions are integrated for collecting driver head and body signals. The head detection provided by the Kinect requires tracked body information. Therefore, to use the Kinect inside a vehicle, it must be mounted above the dashboard to have full vision of the driver's body. Considering the mounting requirements in [38], the Kinect is mounted in the middle of the dashboard, facing the driver, which does not interfere with the driver's field of view and allows for monitoring of the driver's entire upper body. Figure 5-2 illustrates the detected head centre and upper body joints using Kinect and an example depth image.

Figure 5-2. Driver body joints detection with Kinect.

In this study, the head and body signals and colour and depth images are collected and synchronized with a time stamp. The sampling rate is eight frames per second. The data is sampled with an Intel ® Core i7 2.5 GHz computer and the code is written in C++ based on the Windows Kinect SDK and OpenCV. The size of the colour image captured using a Kinect is 1920 x 1080. However, to increase computational efficiency, the stored colour image was compressed to 640 x 360. According to [26], short-term driver mirror checking actions last from 0.5 s to 1 s. Therefore, the sampling frequency is fast enough to capture normal driver actions and behaviour. The three-dimensional head rotation vector contains yaw, pitch, and roll angles. The upper body joints are recorded using X and Y coordinates in the colour image and the corresponding depth value in the depth map. The 42 signals collected are shown in Table 5-1.

TABLE 5-1 MULTIMODAL FEATURES GIVEN BY KINECT

Multimodal Features from Kinect (42 Features)		
Head (12 Features)	Head Pitch Angle (Pitch)	Head Yaw Angle (Yaw)
	Head Roll Angle (Roll)	Left Eye (X, Y, Z)
	Right Eye (X, Y, Z)	Nose (X, Y, Z)
Body (30 Features)	Left Hand (X, Y, Z)	Right Hand (X, Y, Z)
	Left Wrist (X, Y, Z)	Right Wrist (X, Y, Z)
	Left Elbow (X, Y, Z)	Right Elbow (X, Y, Z)
	Left Shoulder (X, Y, Z)	Right Shoulder (X, Y, Z)
	Left Hand Tip(X, Y, Z)	Right Hand Tip(X, Y, Z)

5.2.3 Data Processing

The Kinect data processing methodologies used in this study are described in this section. The two data processing steps are head rotation calibration with an orientation sensor and noise removal and smoothing based on a combination of a median filter and an exponential filter.

5.2.3.1 Kinect Head Rotation Data Calibration

In [39], Kinect head rotation data were evaluated and compared with a high-precision head rotation detection device. The author concluded that the average errors in absolute yaw, pitch, and roll angles were $2.0 \pm 1.2^{\circ}, 7.3 \pm 3.2^{\circ}$, and $2.6 \pm 0.7^{\circ}$, respectively. However, the experiment and data calibration were proposed for indoor environments in standard conditions. However, in this work, the Kinect V2 was implemented inside a vehicle, which is a more challenging environment. During the experiment, the Kinect detection signals inside the vehicle have more noise and are less stable than the signals collected inside the room. Therefore, the first step was to calibrate the Kinect head rotation data with a high-precision head rotation sensor. Since driver head rotation is a very important signal for determining the driver's attention and distraction status, only

the head rotation signals were evaluated in this study and the detected body positions provided by the Kinect were not calibrated.

To calibrate the estimated head rotation results of the Kinect, a head-mounted head tracker was used and three-degree rotation data from the head tracker was used as the ground truth. The head tracker is based on an Arduino microcontroller board and an intelligent nine-axis absolute orientation sensor (BNO055) designed by BOSCH. The sampling frequency of the orientation sensor is up to 100 HZ. The rotation sensor and Arduino data-recording sensor are fixed on a head-mounted harness belt strap, as shown in Figure 5-3. And seven driver behaviours studied in this research are shown in Figure 5-4.

Figure 5-3. Experiment setup for driver behaviour detection.

The Kinect sensor is mounted in the middle of the front dashboard. The optical axis of the Kinect camera is not perpendicular to the yaw axis of driver's head, which will influence the detected yaw angles. The rotation angle of the Kinect sensor in world coordinates is reflected by a constant bias of the detected yaw angle, as shown in Figure 5-5. The blue line is the original yaw angle. The yellow line is the shifted yaw angle, which shifts the original signal by a constant offset (30°). The red line shows the ground truth results of the head tracker. The calibrated Kinect signal and ground truth have similar variations, which means that the head rotation angle detected by Kinect is reliable and can be used for further analysis.

The data recording frequency for the head tracker is 30 Hz, which is approximately three times greater than Kinect, therefore, the head tracker yaw angle shown in Figure 5-5 is the smoothed version of the original signal. Finally, the mean error and standard deviation between the calibrated Kinect signal and the head tracker for yaw, pitch, and roll angles are $1.93 \pm 11.55^{\circ}$, $1.47 \pm 5.98^{\circ}$, and $1.44 \pm 6.98^{\circ}$, respectively.

Figure 5-4. Illustration of Seven driver behaviours.

5.2.3.2 Noise Removal and Data Smoothing

The temporal spikes due to noise can cause more serious problems. The body and head detection results using the Kinect can be influenced by lighting conditions or the location and distance to the driver and human gesture or body pose can influence joint detection, especially inside the vehicle. Due to the less precise detection results using the Kinect, an integrated signal process scheme combining two different filtering techniques is adopted in this study.

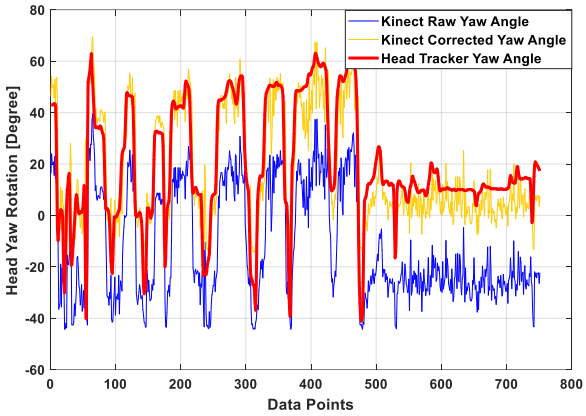

Figure 5-5. Head yaw angle detection results given by Kinect and the orientation-based head tracker.

Specifically, an abnormal data removal and exponential smoothing filter are applied to the raw signals to smooth and track the detection results

$$\hat{x}_n = \begin{cases} x_n, & x_n \neq 0 \\ mean(X_{pre}), & x_n = 0 \end{cases} \quad (5\text{-}1)$$

where $\hat{x}_n$is the filtered data value, $x_n$is the raw data, and X_{pre} represents all the non-zero data before step n. The exponential smoothing filter is defined as (2):

$$\begin{cases} s_0 = x_0 \\ s_t = \alpha \sum_{i=0}^{W} (1-\alpha)^i x_{W-i} \end{cases} \quad (5\text{-}2)$$

where s_t is the smoothed version of raw signal x_t, W is the sliding window size that depends on the number of previous inputs used for smoothing, α is called the dampening factor, which controls the weight of previous inputs and $0 \leq \alpha \leq 1$.

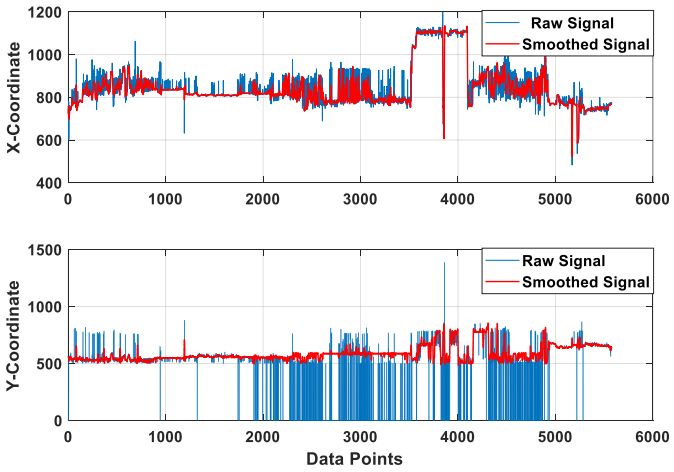

Figure 5-6. Date processing using exponential smoothing filter.

As shown in Figure 5-4, the driver's right arm is partially blocked by the steering wheel, which causes the Kinect to detect inaccurate body joints. During the data recording process, some data points will be lost or unreasonable due to the driver pose, lighting conditions, or the Kinect algorithms. First, these data points are recorded as zeroes to indicate abnormal detection status. Then, the data are fed into the hierarchical filter module to smooth the original signals. To track the signals, an abnormal data removal algorithm is applied. The zero data points are replaced by the mean value of the non-zero data. Then, the exponential smoothing filter is applied to further smooth the noisy signal. Figure 5-6 shows the smoothing result of right wrist signal.

5.3 Evaluation and Identification Algorithms Design

In this section, driver feature evaluation is proposed to study the relationship between driver features and driver behaviour estimation. The most relevant features for driver behaviour recognition are detected. Then, a feedforward neural network is adopted as the driver behaviour classifier to identify the driver actions based on the selected feature vectors.

5.3.1 Feature Importance Evaluation using RF and MIC

For some machine learning tasks, feature vector dimensions can be very high (hundreds or thousands, or even larger). Although machine learning methods are particularly suitable for modelling large datasets, they are always viewed as a black-box where it is difficult to analyse the intrinsic structure. Therefore, it is important to understand how the input features influence or are associated with the model output. In this study, to understand how the driver signals influence behaviour detection, the relationship between body signals and driver behaviour is analysed. Such feature evaluation and selection enable subjective understanding of the relationship between driver body signals and behaviour.

Feature selection is a major research area of feature engineering. By selecting a subset of feature vectors, machine learning models can be trained more efficiently, and better results can be obtained. In this section, to understand how driver features influence the corresponding behaviour detection and which features are important for the behaviour recognition task, two distinct feature selection methods are applied and compared. First, a random forest was used to estimate the driver feature importance with an out-of-bag (OOB) dataset. Second, MIC is used as another indicator for the association between features and the behaviour class. The final conclusion of feature importance will be summarized according to the results given by these two distinct algorithms.

5.3.1.1 RF for Feature Importance Estimation

Random forests, introduced by Breiman in 2001, were built on classification and regression trees [40]. It has proven to be a powerful machine-learning tool for many applications: In [41], the author evaluated the RF classification performance on 121 public datasets and the RF algorithm achieved the best classification result among 179 algorithms. RF is an ensemble learning machine that integrates multiple decision trees. One decision tree is constructed with one root node and multiple middle leaf nodes. The

prediction ability for a single tree is limited, and given a large dataset, overfitting is common for a single decision tree. According to the drawbacks of a single decision tree, RF combines multiple decision trees and uses average or voting schemes to calculate the final results.

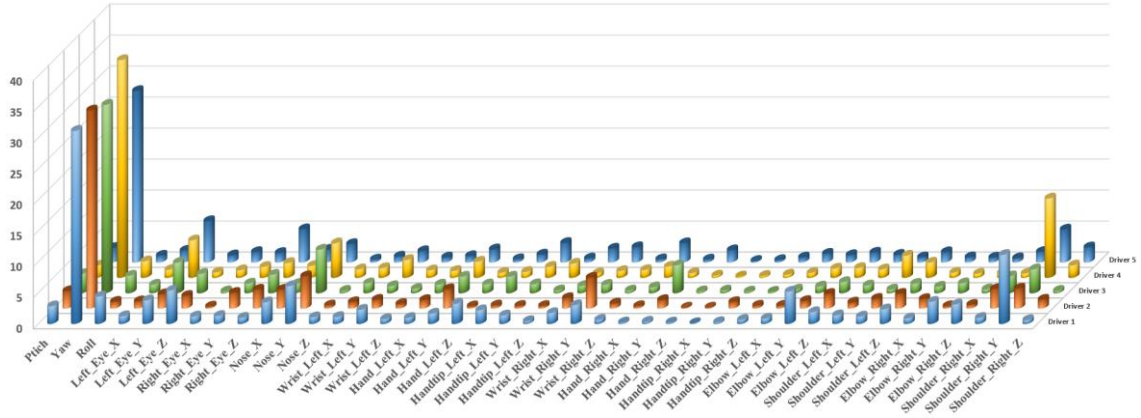

Figure 5-7. Feature importance prediction using random forests based on the permutation method.

Random forest is specific application of ensemble learning. The RF is built with multiple decision trees. To increase the diversity of each tree in the forest, RF is trained using a bootstrap aggregating (Bagging) technique. Specifically, the number of trees B in the RF is selected. Then, according to this number, B separate training datasets are chosen from the original dataset. Since Bagging is a random sampling technique with replacement, approximately one-third of the data is not used for training each subtree. The remaining dataset for each tree is the OOB dataset. Normally cross-validation is not necessary for training RF since the OOB can be used to evaluate the model performance by evaluating the OOB errors [40]. Moreover, the OOB dataset can be used to evaluate the feature importance for model accuracy. The Bagging technique causes the initial training dataset only contain 63.2% samples, the rest 36.8% samples will construct the OOB dataset. To train each decision tree, it is necessary to record the training sample for the tree. Consider the training set as $D = \{(x_1, y_1), (x_2, y_2), \cdots (x_m, y_m)\}$, D_t as the training data for tree h_t, and $H^{oob}(x)$ represents the OOB estimation result on sample x, hence:

$$H^{oob}(x) = arg \max_{y \in Y} \sum_{t=1}^{T} \boldsymbol{I}(h_t(x) = y) \tag{5-3}$$

And the generalizing error for the OOB data is:

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(x,y) \in D} \boldsymbol{I}(H^{oob}(x) \neq y) \tag{5-4}$$

The base learner for RF is single decision tree and the training method for RF rely on the utilization of Bagging technique. Moreover, the trees among the forest are expected to be as diversity as possible to that the model has a good generalization ability. Therefore, a random branch choosing is applied during the training for each tree. Specifically, for each node of the base decision tree learner, a subset that contain k properties are chosen from the total property set. Then, the optimal property will be selected in this subset for generating branch. The parameter k will control the randomness of the random forest and normally it can be selected as $k = \log_2 d$ [40].

To increase the diversity of each tree in the forest, RF is trained using a bootstrap aggregating (Bagging) technique. Specifically, the number of trees B in the RF is selected. Then, according to this number, B separate training datasets are chosen from the original dataset. Since Bagging is a random sampling technique with replacement, approximately one-third of the data is not used for training each subtree. The remaining dataset for each tree is the OOB dataset. Normally cross-validation is not necessary for training RF since the OOB can be used to evaluate the model performance by evaluating the OOB errors [40]. Moreover, the OOB dataset can be used to evaluate the feature importance for model accuracy. To obtain the feature importance, for each variable X_i, the variable is randomly permuted. The feature importance is calculated as follows:

$$\mathrm{I}(\mathrm{X_i}) = \frac{1}{\mathrm{B}}\sum_{\mathrm{t}}^{\mathrm{B}} \mathrm{O}\widetilde{\mathrm{OBerr}}_{\mathrm{t^i}} - \mathrm{OOBerr_t} \qquad (5\text{-}5)$$

where X_i is the permuted i[th] feature in the feature vector X, B is the number of trees in the random forest, $\widetilde{OOBerr}_{t^i}$ is the model prediction error of the perturbed OOB sample with the permuted feature $\mathrm{X_i}$ for tree t, and $\mathrm{OOBerr_t}$ is the untouched OOB data sample with permuted variable.

The concept of permutation feature importance is that a large importance value indicates the feature is influential in the prediction and permuting the feature value will influence the model prediction. In contrast, a small influential feature will have no or less impact on the model prediction. The predicted feature importance for the 42 driver signals using RF are illustrated in Figure 5-7. From the importance estimation results, the driver yaw angles are extremely important for action classification for all five drivers. To verify the prediction results given by RF, the next section proposes another feature evaluation technique called the maximal information coefficient, which uses a completely different method to estimate feature importance.

5.3.1.2 MIC for Feature Importance Estimation

The MIC is designed to efficiently solve the mutual information estimation problem for continuous variables and continuous distributions. The MIC provides an equitable measurement for the linear or nonlinear strength association between two variables. The MIC introduced a maximal mutual information searching technique by varying the grid that drawn on a scatterplot of two variables [42]. Mutual information usually can be used to evaluate the mutual dependence between different variables and assess the amount of information the two variables share, or more generally, the correlation between the joint distribution of the two variables and the product of the independent distribution of the two variables [43]. The mutual information for two discrete vectors is defined as:

$$MI_D(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (5\text{-}6)$$

where MI_D is the mutual information of two discrete vectors, $p(x,y)$ is the joint probabilistic distribution of x and y. $p(x)$ and $p(y)$ are the marginal probability distribution functions of x and y, respectively. For continuous variables, the mutual information format is slightly changed to:

$$MI_C(X,Y) = \iint p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)dxdy \quad (5\text{-}7)$$

where MI_C is the mutual information for two continuous vectors, and $p(x,y)$, $p(x)$, and $p(y)$ represent the corresponding probabilistic density functions.

As shown in (2), calculating the mutual information of continuous variables is difficult. Therefore, the maximal information coefficient technique, which concentrates on the optimal binning method, is applied to assess the mutual information of the continuous case. Meanwhile, MIC enables the mutual information score to be normalized into the range [0,1], which makes assessing the dependency and co-relationship between two variables more convenient. In this study, in addition to the first three continuous head rotation angles, the remaining features are discrete image coordinates and depth values. Therefore, the MIC can be efficiently used for feature association predication.

5.3.1.3 Comparison of the Feature Importance Prediction

To evaluate the prediction results of feature importance using the two algorithms, the ten most important features for each subject are extracted and compared. Specifically, for each driver, the ten most important features are selected. Then, five selected feature vectors are fused into 42 bins and the count in each bin represents the number of

occurrences for each feature of the five subjects. Therefore, the highest value, 5, indicates the feature is the one of the ten most important features for all five drivers. The statistical results are shown in Figure 5-8. Blue bars represent the results of RF while red bars standard for maximal information coefficient method.

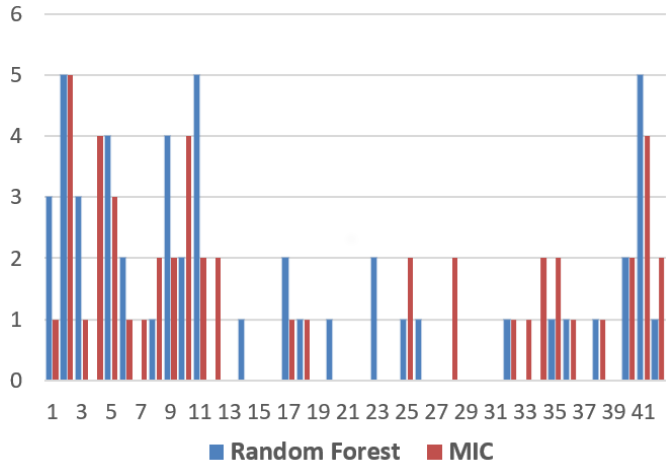

Figure 5-8. Feature importance prediction results using random forest OOB permutation.

TABLE 5-2 FEATURE IMPORTANCE ESTIMATION RESULT

	Importance Order	Features
RF	1-5	Yaw, Left_eye_Y, Red_eye_Y, Nose_Y, Right_shoulder_Y
	6-12	Pitch, Roll, Left_eye_Z, Nose_X, Left_hand_Y, Right_wrist_Y, Right_shoulder_X
MIC	1-5	Yaw, Left_eye_X, Left_eye_Y, Nose_X, Right_shoulder_Y
	6-12	Right_eye_Y, Right_eye_Z, Nose_Y, Nose_Z, Right_hand_X, Left_shoulder_X, Right_shoulder_X

As shown in Figure 5-8, although the prediction results of the two algorithms are not identical, there is some consistency in the results of the two algorithms. For example, the driver yaw and y-coordinate of the right shoulder features (No. 2 and No. 41) are both significant. According to Figure 5-8, the 12 most important features (marked as the ten most important features by at least two drivers) are listed in Table 5-2.

According to Table 5-2, the importance predictions given by RF and MIC are similar. The most important features are the head rotation angles (yaw, pitch, and roll), eye and nose position, shoulder position, and hand position. The remaining features such as the wrist, hand tip, and elbow positions are less likely to influence the behaviour detection result. A quantitative analysis of the feature impact on behaviour recognition based on a feedforward neural network is proposed in the next section.

5.3.2 Feedforward Neural Network for Driver Behaviour Classification

In this section, an ANN is used for driver behaviour pattern recognition. Specifically, a one-way FFNN is adopted. The FFNN passes the input vectors to the output layer-by-layer without any feedback connections. The FFNN is a powerful tool for solving complex nonlinear mapping problems. By learning the neuron parameters and the connection width, the FFNN model can construct a nonlinear mapping between the input and output.

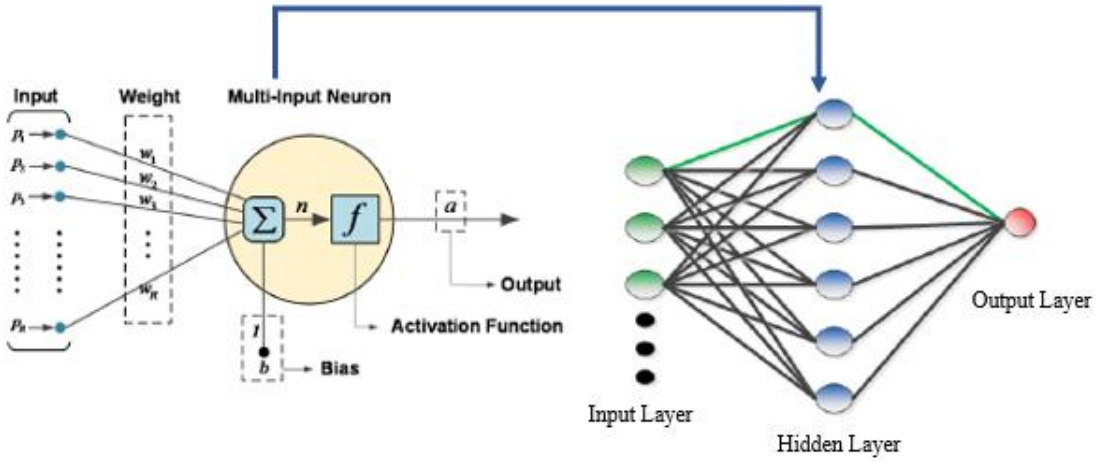

Figure 5-9. Illustration of multi-input neuron and multi-layer FFNN.

Artificial neural network model is inspired by the biological neural networks that constitute the animal brains. The basic units of ANN model are called the neurons, each ANN model is constructed with multiple neurons, which connected loosely through the weighted connections. The connection between the artificial neurons are called “edges”. The neurons will compute the weighted inputs with some non-linear activation function. Normally the activation function can be represented as a $sigmoid$ or $tanh$ function. Once the computed result exceeds a certain threshold, then this neuron is called to be activated, otherwise output the zero value. Usually the ANN is structured with multiple layers and each layer contains multiple neurons. The layers between the input layer and output layer are called hidden layers of the ANN model. Normally, the more hidden layers the ANN model have, the more powerful the model will be. However, increasing the number hidden layers will significantly increase the computational burden and the requirement for data volume. Therefore, deeper ANN models are more easily to be overfitted. The training process of the ANN model is to judge the weights of the edges

and bias in each neuron to minimize the total loss of the output layer. In this part, the feedforward neural network is used, which is a specific ANN model with all the edges have the feedforward connection.

The left part of Figure 5-9 shows the structure of single neuron and the right part indicates the architecture of a multilayer FFNN with one hidden layer. The neuron unit will weight the R inputs with a bias b to form the activation input n, which has the following form:

$$n = \sum_j^R w_j p_j + b = \boldsymbol{Wp} + b \tag{5-8}$$

Then n passes through the activation function f to generate the neuron output a.

$$a = f(n) \tag{5-9}$$

The total FFNN model can be approximately represented as follows:

$$y = f(X, \boldsymbol{\theta}) + \epsilon \tag{5-10}$$

where y is the output of FFNN, $f()$ is the learned model mapping function with model parameter $\boldsymbol{\theta}$, X is the input data vectors, and ϵ is the bias between the actual output and the target.

For the FFNN, parameter $\boldsymbol{\theta}$ represents the set of activation function parameters and the width set between neurons. In this study, a multi-layer FFNN with one hidden layer is used to train the driver behavior recognition model. The sigmoid transfer function in the hidden layer is chosen. The sigmoid activation function for a single neuron is represented as:

$$f = \frac{1}{1+e^{-X}} \tag{5-11}$$

where f is the neuron output and X is the neuron input, which has the following form:

$$X = \sum_{i=1}^N \omega_i x_i + b \tag{5-12}$$

where ω_i is the weight of the ith input, and normally each neuron has a bias parameter b. An important reason for using the sigmoid activation function is the computation efficiency during model training. The backpropagation requires the derivative of the neuron transfer function to be calculated. While the sigmoid function has the convenient derivative form [44]:

$$f' = f \cdot (1 - f) \tag{5-13}$$

Although the sigmoid function will cause the loss of a gradient problem in most scenarios, it is not a serious problem in this shallow network case. In this case, the supervised FFNN is trained with driver head and body signals as the input and an output

of the corresponding behaviour among the seven actions. Unlike some existing research that uses time-series models, the FFNN used in this study does not consider the previous step status of the driver. The reason for this is that humans can normally distinguish the current driving behaviour using one image and do not require video sequences. Unlike the inner mental states of the driver, which is a long-term process and depends on previous states, the outer behaviours can be considered a transient state and are not highly dependent on prior information. Therefore, the FFNN is applied to detect the driving tasks frame-by-frame based on the collected driver body information.

Since time information is not considered in the model construction procedure, the training and testing dataset are reordered randomly. For model training, cross-validation is used. Specifically, the leave-one-out (Loo) method is adopted. For the five-driver dataset, data of four drivers are used for model training and validation, the data of the remaining driver is used to test the classification performance. The general classification accuracy is the average of the five classification results. Another hyper-parameter for FFNN is the number of neurons in the hidden layer. To evaluate the influence of neuron quantity on classification performance, different neuron numbers and cross-validation are studied. A boxplot of the classification results is shown in Fig. 9. The neuron numbers vary from 10 to 100 with an interval of 10. The red line represents the mean accuracy of the five drivers with different neurons. As shown in Figure 5-10, variation in the number of neurons does not significantly influence performance. The most accurate detection occurs at the 100-neuron cases, with an accuracy of approximately 81.2%. In the next section, more detailed statistical results using FFNN with 60 neurons are proposed, and the results are compared with multiple machine learning methods.

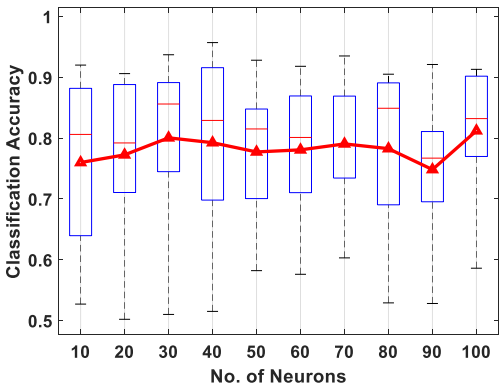

Figure 5-10. Boxplot of the FFNN classification results with cross-validation of the neuron numbers in the hidden layer.

The importance estimation for the 42 features given by two feature importance estimators (RF and MIC) are illustrated in Figure 5-8. As shown in Figure 5-8, although the results are not the same, the importance estimation results given by the two algorithms show the consistency. For example, the head rotation angles as well as the arm positions are estimated the most important features for the two algorithms while some other features like wrist and hand tips are regarded as less important by the two algorithms.

5.4 Experiment Results and Analysis

In this section, the task recognition results are discussed. Specifically, task classification with FFNN is compared with other machine learning methods. In addition, the impact of the head, body and depth information on the classification results will be evaluated separately in part two.

5.4.1 Behaviour Recognition Results

In this section, the identification accuracy for driving and non-driving tasks is analysed. As mentioned in the previous section, the classification model is trained using the LOO-cross validation method. The prediction results for the five drivers are illustrated in Table 5-3. The first four mirror checking tasks are divided into driving-related tasks, while the remaining three tasks are divided into non-driving and distraction tasks.

TABLE 5-3 CLASSIFICATION RESULTS USING FFNN WITH ENTIRE FEATURES

	Driving Tasks				Non-Driving Tasks			
	T1	T2	T3	T4	T5	T6	T7	Ave
D1	0.905	0.856	0.843	0.925	0.686	0.896	1.00	0.883
D2	0.380	0.557	0.498	0.985	0.877	0.617	0.684	0.630
D3	0.985	0.976	0.690	0.998	0.988	1.00	0.662	0.898
D4	0.720	0.973	0.994	0.999	1.00	1.00	0.858	0.927
D5	0.583	0.977	0.801	1.00	0.798	0.969	0.991	0.871
Mean	0.715	0.838	0.747	0.981	0.867	0.884	0.838	0.824

The seven driver tasks are ordered as {Normal Driving, Right Mirror-Checking, Rear Mirror-Checking, Left Mirror-Checking, Using Video Device, Texting, and Answering Mobile Phone}. As shown in the far-right column of Table 5-3, the average classification result (Ave) for each driver is defined as the average of the seven tasks. The mean values shown in the bottom row represent the average classification accuracy for each task of the five drivers. Detection results equal to 1.00 shown in Table 5-3 indicate an accuracy of 100%. The FFNN classification model is trained with 60 neurons using the entire feature vector (42 features). The classification results for driver 2 are much lower than the other four drivers, with an average of only 0.630. This is due to imprecise detection of the driver

skeleton during data collection. To have a clear perspective of detection performance, the confusion matrix for driver 2 is shown in Figure 5-11.

In the confusion matrix, the green diagonal shows the number of correct detection cases for that class. The bottom row shows the classification accuracy with respect to the target value, and the far-right column shows the classification accuracy with respect to the predicted labels. As shown in Figure 5-11, the normal driving behaviour for driver 2 only achieved 38% detection accuracy and 289 cases are classified into the phone answering task. This is mainly due to the similar postures between normal driving and phone answering behaviour. Once hand detection is inaccurate, it is very difficult to classify these two tasks only according to head pose. Detailed discussion will be proposed later. In addition, the low detection accuracy means the trained model using the other four drivers is not sufficient to precisely recognize all the behaviours for driver 2 due to the diversity of the drivers. However, once driver 2 is included in the training data, the model will obtain better detection results for the other four drivers. The most accurate detection occurs for driver 4, the relative results are shown in Figure 5-12.

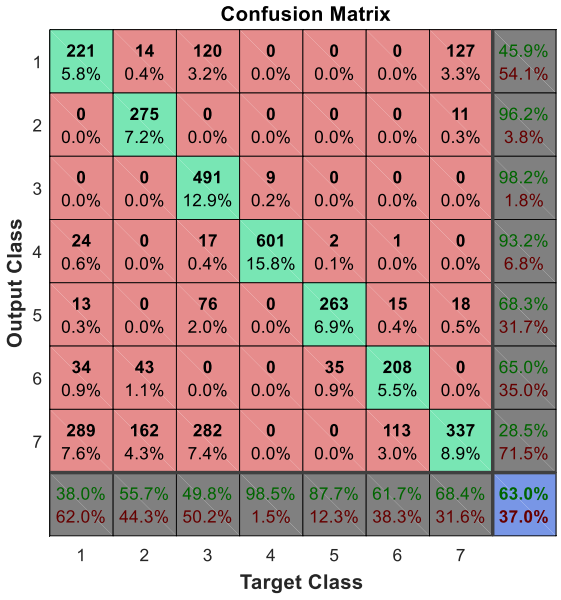

Figure 5-11. Confusion matrix of driving tasks classification results for driver 2.

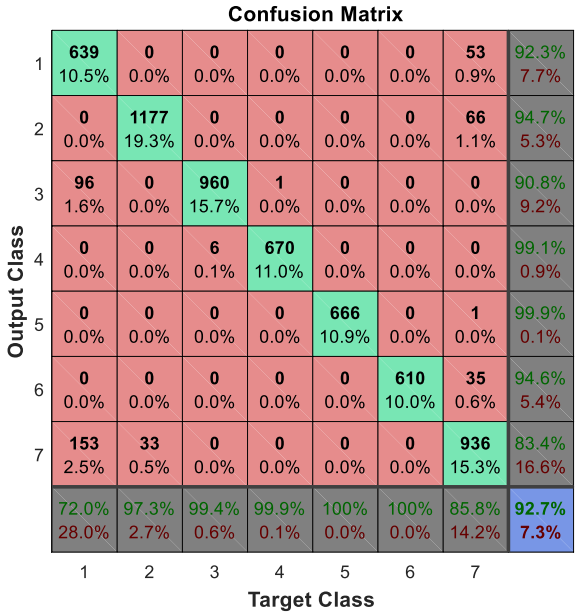

Figure 5-12. Confusion matrix of driving task classification results for driver 4.

As shown in Figure 5-12, the classification results for the seven tasks for driver 4 are much better than for driver 2. False detection between different classes decreased significantly. Similar results are achieved for the remaining three drivers. In conclusion, although very accurate results were not achieved for driver 2 compared with the other drivers, the general classification accuracy for the seven tasks was 82.4% (the mean value of the average column), which indicates efficient classification results.

In Table 5-4, the classification results of FFNN are compared with four other machine learning methods, which are random forest (RF), support vector machine (SVM), naïve Bayes (NB), and K-nearest neighbour (KNN, K equals 5 in this case). The accuracy in Table 5-4 is defined as the average detection result for the five drivers, i.e., the average of the Ave column in Table 5-3. Meanwhile, to evaluate the driver distraction detection performance, the seven classification tasks are merged into a binary classification. Here, the negative group is defined as the combination of the first four normal driving tasks, and the true distraction group consists of the remaining three distracted driving tasks.

TABLE 5-4 CLASSIFICATION RESULTS USING DIFFERENT MACHINE LEARNING METHODS

	Accuracy	TPR	FPR	Training Cost [s]	Testing Cost [s]
FFNN	0.824	0.939	0.088	4.92	0.05
RF	0.736	0.900	0.144	33.55	0.41
SVM	0.747	0.913	0.177	2.85	0.03
NB	0.767	0.922	0.171	0.188	0.02
KNN	0.623	0.771	0.090	0.049	1.08

The TPR and FPR in Table 4 represent the true positive rate (sensitivity) and false positive rate, respectively. TPR and FPR are calculated as:

$$TPR = \frac{TP}{P} \tag{5-14}$$

and

$$FPR = \frac{FP}{N} \tag{5-15}$$

where TP is the number of correctly detected distracted cases, P is the total number of distracted cases, which is the total quantity of the three distracted cases. FP is the number of false detections. In this case, it represents the number of normal driving tasks that are classified in the abnormal driving group. Finally, N is the total amount of normal driving cases.

According to Table 5-4, FFNN binary classification outperforms the other four models, indicating that FFNN is a powerful model suitable for driver behaviour modelling. Note that there are no optimization algorithms used in the other four models. These models are used with their default setup in MATLAB. The RF is constructed with 100 decision trees, and SVR uses a radial-based kernel. Better results may be obtained with parameter tuning and optimization, however, this is beyond the scope of this study. The binary classification model can distinguish normal driving behaviour and distracted behaviour. From the perspective of safety, although it may annoy the driver, it is safe to classify normal driving behaviour into distracted behaviour and warn the driver. On the other hand, if the model classifies distracted behaviour into the normal driving group, it is more dangerous than the previous case and this misclassification should be avoided. In the real world, in terms of non-driving tasks, the time constants are always much longer than normal driving tasks, texting or answering a phone can last for a few minutes. However, the mirror-checking actions usually last for one to two seconds. These time-properties of the different tasks can be adopted to predict the correct states in the future.

5.4.2 Feature Evaluation for Behaviour Classification Performance

In this section, the impact of the driver's head and body features on driving task classification will be analysed. The feature evaluation is divided into three parts. First, the depth information of the detected joints and facial landmarks (eyes and nose) are evaluated. Then, task classification using only head signals or only body signals is proposed. The classification results for these three parts are illustrated in Table 5-5.

TABLE 5-5 TASKS CLASSIFICATION BASED ON DIFFERENT FEATURES

		Driving Tasks				Non-Driving Tasks			
		T1	T2	T3	T4	T5	T6	T7	Ave
2D Only (29)	D1	93.8	48.1	60.5	88.0	100	91.7	70.4	79.4
	D2	66.8	79.6	41.4	88.2	62.7	84.6	37.1	62.7
	D3	58.7	64.4	100	100	97.8	96.0	48.9	77.6
	D4	69.3	73.1	96.9	99.7	99.1	96.7	91.0	87.6
	D5	68.5	54.3	39.3	100	32.7	95.5	99.5	71.1
	Mean	71.4	63.9	67.6	95.2	78.5	92.9	69.4	75.7
Head Only (3)	D1	78.9	92.3	99.7	96.8	5.4	24.9	0.7	61.8
	D2	0.0	85.4	15.7	66.7	93.0	90.8	26.0	44.6
	D3	39.3	4.2	16.6	100	98.5	96.2	40.5	52.7
	D4	94.6	99.8	77.2	100	33.2	99.3	24.8	74.8
	D5	52.8	91.1	33.5	99.6	97.4	99.6	50.0	76.6
	Mean	53.1	74.5	48.5	92.6	65.5	82.1	28.4	62.1
Body Only (30)	D1	76.2	0.0	2.0	0.0	90.2	90.0	96.0	47.0
	D2	12.0	0.0	26.6	0.0	40.3	1.2	32.9	16.3
	D3	1.2	100	0.0	7.4	96.9	94.0	53.5	48.2
	D4	55.4	45.0	0.0	97.0	97.9	95.1	62.1	58.9
	D5	44.0	2.1	0.0	97.3	35.3	75.0	99.3	49.2
	Mean	37.8	29.4	5.72	40.3	72.1	71.1	68.8	43.9

First, the 2D-only case in Table 5-5 represents a feature set only consisting of the head rotation and joint coordinates (X and Y coordinates), and depth information is not used. As shown in Table 5-5, the model trained with 2D information achieves similar accuracy results compared with the model trained with the entire feature set (Table 5-3). The results indicate that depth information has very limited impact on the model classification task.

The second block in Table 5-5 illustrates driving task classification using only head pose information. Specifically, the three head rotation angles: yaw, pitch, and roll are used to construct the feature set. The classification accuracy using head pose is much less than the accuracy in previous cases. For the left mirror-checking and texting tasks, which have significantly different characteristics than other tasks, the detection is accurate. However, for the other tasks, using only head pose information is not sufficient for accurate detection. For example, the driver rear mirror-checking behaviour (T3) is like the task of using a video device. Moreover, without considering body information, the phone answering behaviour cannot be detected accurately since the driver is usually looking forward to the road and the head pose is very similar to normal driving.

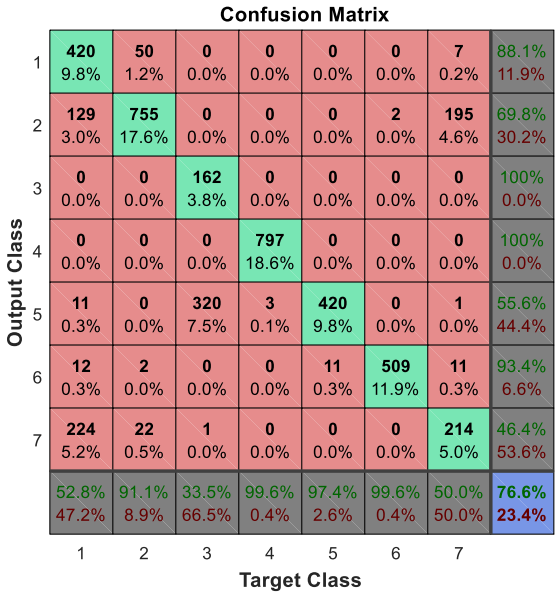

Figure 5-13. Confusion matrix of the classification of the seven tasks for driver 5 using 3D head pose features only.

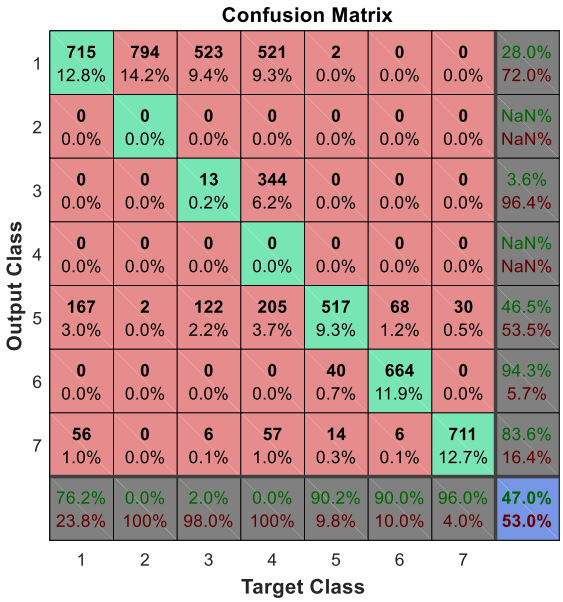

Figure 5-14. Confusion matrix of the classification of the seven tasks for driver 1 using only body features.

The confusion matrix for driver 5, which has the most accurate results among the five drivers, is shown in Figure 5-13. In terms of driver 5, FFNN is not able to accurately distinguish tasks 1, 3, and 7, which are normal driving, rear mirror checking, and answering the phone. Approximately one-third of normal driving cases (224 samples) are

classified into the phone answering task. For rear mirror checking, more samples are falsely detected as the video device using task. It is obvious from the confusion matrix that, without using body features and using only head pose features, it is difficult to identify the actual driver behaviour. The reason may because when using the head only features, the training data from other drivers can be representative to the fifth driver. Meanwhile, after analyses the testing data from driver 5, it was found that the data are less noisy compared with the other drivers as less missing points were founded. Therefore, the system performance is expected to be improved with clearer dataset or more efficient noise removal filter.

The third block indicates the behaviour detection using only body features. There are 30 total features used, containing the X, Y, and Z coordinates of the hand, wrist, elbow, and shoulder joints. As shown in Table 5-5, ignoring the 3D head pose features and the eyes and nose location information, the detector fails to identify the mirror-checking behaviours. By using only body features, the distraction behaviour can be detected with a certain degree of accuracy, while the detection accuracy for the four mirror checking behaviours is quite low. The worst case is the rear-mirror checking behaviour, which only achieved 5.72% accuracy in general.

TABLE 5-6 CLASSIFICATION RESULT USING FFNN WITH 18 SELECTED FEATURES

	Driving Tasks				Non-Driving Tasks			
	T1	T2	T3	T4	T5	T6	T7	Ave
D1	90.6	91.8	98.2	98.1	100	88.3	100	95.1
D2	1.4	20.9	25.5	91.1	97.3	60.2	93.5	49.3
D3	97.8	98.7	82.5	99.8	98.1	100	28.5	85.0
D4	65.2	78.1	96.7	99.9	98.3	97.9	80.4	86.2
D5	51.4	100	89.9	99.9	87.9	94.9	100	88.0
Mean	61.3	80.0	78.6	97.8	96.3	88.2	80.5	80.7

Based on the above evidence, to obtain a better understanding of the tasks that the driver is undergoing, both the head and body features are necessary. From the feature comparison, head pose features are more useful than body features since the 3D head pose information leads to better detection results (62.1% average) compared with the 3D body features (43.9%). Figure 5-14 shows the model classification results for driver 1 when the model is only trained with body features. The three distraction behaviours are accurately detected using body features, while the four mirror checking detections are difficult to identify.

Finally, the important features for driver task classification are selected according to the integrated feature extraction technique in section 5.3 and these features are input to the FFNN model. In total, 18 features are selected as important features. The feature set contains the following features: {yaw, pitch, roll, nose (X, Y, Z), left hand (X, Y, Z), right hand (X ,Y, Z), left shoulder (X, Y, Z), and right shoulder (X, Y, Z)}. The classification results are shown in Table 5-6. The overall accuracy of task detection is 80.7%, which is slightly less than the model trained with the entire feature set. However, the selected 18 features still yield an acceptable accurate detection, also, the time cost of the training and testing process is less than when using the entire feature set. Therefore, the driver tasks can be detected using the small feature set.

5.5 Discussion and Future Work

Based on the results shown in the previous section, driver task recognition can be achieved with a feedforward neural network. The FFNN could reasonably detect seven tasks for different drivers and achieved high-precision detection for secondary tasks. The FFNN has advantages for driver task detection over other machine learning methods. Classification for different tasks resulted in different detection accuracy. The results indicate that for tasks such as texting, left and right mirror checking, which have obvious distinct features, the detection results are accurate. However, for tasks that have similar postures, the model can be confused. In this study, normal driving behaviour has similar characteristics to rear mirror checking behaviour and phone-answering tasks; therefore, the detection results for these behaviours are slightly worse than for other behaviours. In addition to the similar characteristics of these behaviours, another reason for less accurate detection results is driving style. Although accurate detection results can be achieved for some drivers, the FFNN cannot obtain a universal accuracy for all drivers. For example, task detection for driver 2 is less accurate than for other drivers due to driving style and sensor noise. A driver has a unique driving and mirror checking style. Some drivers prefer to use significant head and body movement during mirror checking while others may try to use less body movement and use eye movement to capture information. Therefore, the following aspects are discussed and can be improved to achieve higher task detection accuracy.

First, the driver head and body signals captured with a Kinect are very noisy. Sometimes the detection is less precise, and the detected joint positions are shifted and

unreasonable. This phenomenon is particularly worse for the seated driver inside the vehicle. In this study, a simple integrated tracking and smoothing technique is used, which consists of a jitter removal filter and an exponential filter. Although the integrated filter can recover unreasonable detection and smooth the signals, important information can be lost, and the filter can be further improved by using more advanced filters such as the Kalman filter or particle filter for joint position tracking. Therefore, the quality of Kinect signals, as well as the model detection results, can be further improved. Moreover, in this study, only colour and depth images are collected; however, Kinect also supports audio recording. Therefore, in the future, audio information in the cabin can be captured as another important data source to assist in the detection of non-driving-related tasks.

Second, in this study, the feature selection and extraction methods are constructed based on random forests and the maximal information coefficient technique. This integrated method estimates the importance of the driver body features and the FFNN using these features achieved accurate detection results for some drivers. However, detection accuracy decreased significantly for the second driver for a few reasons. To obtain universal accurate task detection results, more drivers must to be studied in the future. Increasing the dataset volume and data diversity is an efficient way to solve the problem. Meanwhile, more driver features can be used. In the study, only the position and depth information for the eyes are used. The driver gaze movement and gaze tracking technique have been successfully adopted in some research on driver fatigue, inattention, and distraction monitoring. Gaze information can be very useful when the drivers prefer not to move their body when performing mirror-checking tasks.

Finally, on-road data collection can be performed in the future for the study of real-time driver behaviour detection within normal driving environments. Currently, for safety considerations, the drivers were asked to perform the experimental tasks without driving the vehicle because secondary tasks such as texting and playing a video device are extremely dangerous when driving and should be avoided. Therefore, the most naturalistic data is difficult to collect. However, in the future, with the help of ADAS and the mid-level automated vehicle technique, drivers are allowed to remove their hands from the steering wheel. Therefore, more distraction behaviours can be collected and the study for real-time driver distraction detection in a real vehicle can be performed. The

real-time driver monitoring study will significantly improve the driving safety for both conventional vehicles and highly automated vehicles.

5.6 Conclusions

In this study, driving behaviours for different drivers are studied. The driving behaviours are classified into two categories, normal driving tasks and distracting tasks. A feedforward neural network is trained to distinguish the four mirror-checking behaviours from the three secondary driving tasks. Both depth information and the 2D location of the body joints are collected using Kinect. The noisy data is processed with an integrated filtering system. Then, the importance of each driver feature to behaviour recognition is evaluated using random forests and maximal information efficiency. The feature importance prediction with these two feature evaluation techniques shows consistent results. The most important driver features for driver behaviour among all the drivers are determined. The FFNN has been proven to have advantages for behaviour detection tasks over other popular machine learning methods. The model achieved an average of greater than 80% accuracy for the five drivers. With the evaluation of feature importance and their influence on the classification task, the head pose feature, hand position, and shoulder positions for the driver are selected as the most important features. In addition, based on the evaluation of the depth, head, and body features, it is found that the depth information for the body joints and facial markers have very limited influence on behaviour recognition. Meanwhile, the head and body features should be combined with a comprehensive driver behaviour understanding since only using the head or body features will lead to large false detection rates.

The conclusion is made that for future driving monitoring and behaviour understanding, the head and body signals are equally important and necessary. Future works will focus on the collection of more real-world dataset and recognize more sophisticate driver behaviours. This study will benefit future ADAS design and improve driving safety by real-time driver status monitoring.

5.7 Reference

[1] Wang, Fei-Yue. "Computational Social Systems in a New Period: A Fast Transition Into the Third Axial Age." IEEE Transactions on Computational Social Systems, 4(3): 52-53, 2017.

[2] Arbabzadeh, Nasim, and Mohsen Jafari. "A Data-Driven Approach for Driving Safety Risk Prediction Using Driver Behaviourand Roadway Information Data." IEEE Transactions on Intelligent Transportation Systems (2017).

[3] Xing, Yang, *et al*. "Driver Workload Estimation using a Novel Hybrid Method of Error Reduction Ratio Causality and Support Vector Machine." Measurement, 2017, in press.

[4] Michon, John A. "A critical view of driver behaviourmodels: What do we know, what should we do." Human behaviourand traffic safety (1985): 485-520.
[5] Wang, F-Y., Zheng, N., Cao, D., *et al*. "Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence." IEEE/CAA Journal of Automatica Sinica, 2017.
[6] Castignani, German, *et al*. "Driver behaviourprofiling using smartphones: A low-cost platform for driver monitoring." IEEE Intelligent Transportation Systems Magazine 7.1 (2015): 91-102.
[7] Lv, C., Liu, Y., *et al*. Simultaneous Observation of Hybrid States for Cyber-Physical Systems: A Case Study of Electric Vehicle Powertrain, IEEE Transactions on Cybernetics, 2017.
[8] Foroutan, N.,Hamzeh, A. "Discovering the Hidden Structure of a Social Network: A Semi Supervised Approach." IEEE Transactions on Computational Social Systems, 3(4): 151-163, 2016.
[9] Tong, G., *et al*. "Effector Detection in Social Networks." IEEE Transactions on Computational Social Systems, 4(1): 14-25, 2017.
[10] Lv, C., Wang, H., and Cao, D. "High-Precision Hydraulic Pressure Control Based on Linear Pressure-Drop Modulation in Valve Critical Equilibrium State." IEEE Transactions on Industrial Electronics, 2017.
[11] Liang, Yulan, and John D. Lee. "A hybrid Bayesian Network approach to detect driver cognitive distraction." Transportation research part C: emerging technologies 38 (2014): 146-155.
[12] Lv, C., Cap, D., *et al*. Analysis of Autopilot Disengagements Occurring During Autonomous Vehicle Testing, IEEE/CAA Journal of Automatica Sinica, 2017.
[13] Sivak M, Schoettle B. Motion sickness in self-driving vehicles. UMTRI-2015-12. (2015). Ann Arbor: University of Michigan Transportation Research Institute.
[14] Jain, Ashesh, *et al*. "Car that knows before you do: Anticipating manoeuvres via learning temporal driving models." Proceedings of the IEEE International Conference on Computer Vision. 2015.
[15] Murphy-Chutorian, Erik, and Mohan Manubhai Trivedi. "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness." IEEE Transactions on intelligent transportation systems 11.2 (2010): 300-311.
[16] Murphy-Chutorian, Erik, and Mohan Manubhai Trivedi. "Head pose estimation in computer vision: A survey." IEEE transactions on pattern analysis and machine intelligence 31.4 (2009): 607-626.
[17] Vicente, Francisco, *et al*. "Driver gaze tracking and eyes off the road detection system." IEEE Transactions on Intelligent Transportation Systems 16.4 (2015): 2014-2027.
[18] Fletcher, Luke, and Alexander Zelinsky. "Driver inattention detection based on eye gaze—Road event correlation." The international journal of robotics research 28.6 (2009): 774-801.
[19] Das, Nikhil, Eshed Ohn-Bar, and Mohan M. Trivedi. "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics." Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on. IEEE, 2015.
[20] Cheng, Shinko Y., Sangho Park, and Mohan M. Trivedi. "Multi-spectral and multi-perspective video arrays for driver body tracking and activity analysis." Computer Vision and Image Understanding 106.2 (2007): 245-257.
[21] Du, Haiping, Weihua Li, and Nong Zhang. "Vibration control of vehicle seat integrating with chassis suspension and driver body model." Advances in Structural Engineering 16.1 (2013): 1-9.
[22] Tran, Cuong, Anup Doshi, and Mohan Manubhai Trivedi. "Modeling and prediction of driver behaviourby foot gesture analysis." Computer Vision and Image Understanding 116.3 (2012): 435-445.
[23] Ohn-Bar, Eshed, *et al*. "Head, eye, and hand patterns for driver activity recognition." Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014.
[24] Rezaei, Mahdi, and Reinhard Klette. "Look at the driver, look at the road: No distraction! No accident!" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
[25] Mbouna, Ralph Oyini, Seong G. Kong, and Myung-Geun Chun. "Visual analysis of eye state and head pose for driver alertness monitoring." IEEE transactions on intelligent transportation systems 14.3 (2013): 1462-1469.
[26] Li, Nanxiang, and Carlos Busso. "Detecting drivers' mirror-checking actions and its application to manoeuvre and secondary task recognition." IEEE Transactions on Intelligent Transportation Systems 17.4 (2016): 980-992.
[27] Kondyli, Alexandra, *et al*. "Computer Assisted Analysis of Drivers' Body Activity Using a Range Camera." IEEE Intelligent Transportation Systems Magazine 7.3 (2015): 18-28.
[28] Jha, Sumit, and Carlos Busso. "Analyzing the relationship between head pose and gaze to model driver visual attention." Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. IEEE, 2016.
[29] Morton, Jeremy, Tim A. Wheeler, and Mykel J. Kochenderfer. "Analysis of recurrent neural networks for probabilistic modeling of driver behavior." IEEE Transactions on Intelligent Transportation Systems 18.5 (2017): 1289-1298.
[30] Wollmer, Martin, *et al*. "Online driver distraction detection using long short-term memory." IEEE Transactions on Intelligent Transportation Systems 12.2 (2011): 574-582.
[31] Tango, Fabio, and Marco Botta. "Real-time detection system of driver distraction using machine learning." IEEE Transactions on Intelligent Transportation Systems 14.2 (2013): 894-905.
[32] Lee, Yi-Ching, *et al*. "Learning to Predict Driver Behaviourfrom Observation." (2017). The AAAI 2017 Spring Symposium on Learning from Observation of Humans Technical Report SS-17-06.
[33] Harbluk, Joanne L., Y. Ian Noy, and Moshe Eizenman. The impact of cognitive distraction on driver visual behaviour and vehicle control. No. TP 13889 E. 2002.
[34] Mühlbacher-Karrer, Stephan, *et al*. "A Driver State Detection System—Combining a Capacitive Hand Detection Sensor With Physiological Sensors." IEEE Transactions on Instrumentation and Measurement 66.4 (2017): 624-636.
[35] Ranft, Benjamin, and Christoph Stiller. "The role of machine vision for intelligent vehicles." IEEE Transactions on Intelligent Vehicles 1.1 (2016): 8-1.
[36] Gaglio, Salvatore, Giuseppe Lo Re, and Marco Morana. "Human activity recognition process using 3-D posture data." IEEE Transactions on Human-Machine Systems 45.5 (2015): 586-597.
[37] Neto, Laurindo Britto, *et al*. "A Kinect-based wearable face recognition system to aid visually impaired users." IEEE Transactions on Human-Machine Systems 47.1 (2017): 52-64.
[38] Mehran Azimi. Skelton Joint Smoothing White Paper [Online]. Available: https://msdn.microsoft.com/en-us/library/jj131429.aspx.
[39] Darby, John, *et al*. "An evaluation of 3D head pose estimation using the Microsoft Kinect v2." *Gait & posture* 48 (2016): 83-88.

[40] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[41] Do we need hundreds of classifiers to solve real world classification problems." J. Mach. Learn. Res 15.1. (2014): 3133-3181
[42] Reshef, David N., *et al*. "Detecting novel associations in large data sets." science 334.6062 (2011): 1518-1524.
[43] Cover, T.M.; Thomas, J.A. (1991). Elements of Information Theory (Wiley ed.). ISBN 978-0-471-24195-9.
[44] Duch, Włodzisław, and Norbert Jankowski. "Survey of neural transfer functions." Neural Computing Surveys 2.1 (1999): 163-212.
[45] Ohn-Bar, Eshed, *et al*. "On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems." Computer Vision and Image Understanding 134 (2015): 130-140.

6

PAPER V – Driver Behaviour Detection with an End-to-End Approach

Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach

Authors:

Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang

This paper is under review in:

IEEE Transactions on Vehicular Technology

Abstract

Drivers' decision and their corresponding behaviours are important factors that can affect the driving safety. Therefore, it is necessary to understand the driver behaviours in real time. In this study, a driving-related activities recognition system is designed based on the deep CNN. Specifically, seven common driving activities are identified, which are the normal driving, left mirror checking, right mirror checking, rear mirror checking, using in-vehicle radio device, texting, and answering mobile phone activities. Among these behaviours, the first four are regarded as normal driving tasks, while the rest three are classified into the distraction group. The experimental images are collected using a consumer range camera named Kinect, and five drivers are involved in the data collection procedure. Before training the CNN model, the raw images are segmented using GMM to extract the driver from the background. Then, two different pre-trained CNN models are adopted, which directly takes the processed RGB images as input and outputs the identified behaviour label. In this work, the AlexNet and GoogLeNet are selected as the two pre-trained CNN models. To reduce the training cost, transfer learning is applied to these CNN models. The detection results for the seven tasks achieve an average of 79% accuracy for AlexNet and 73.5% accuracy for GoogLeNet. Finally, the two models are trained to solve the binary classification tasks and identify whether the driver is in normal driving status or being distracted. The binary detection rate achieved 94.2% accuracy.

Index Terms - Driver Behaviour, driver distraction, convolutional neural network, transfer learning.

6.1 Introduction

6.1.1 Motivations

Driver is the most important part within the TDV loop. Their decision and behaviours are the major aspects that can influence the driving safety. It is reported that more than 90% light vehicles accidents are caused by human driver misbehaviour in the United States, and the accident rate can be reduced by 10% to 20% with a precise driver behaviour monitoring system [1]-[5]. Therefore, the recognition of driver behaviours is one of the most important tasks for the intelligent vehicles. For the conventional vehicles, driver is in the centre of the TDV loop. The understanding of driver behaviour enables the driver assistance system (DAS) to figure out the optimal vehicle control strategy that can satisfy the driver states [6]-[9]. In terms of the intelligent and partially automated vehicles such as the Level-3 automated vehicles (according to the definition in Society of Automotive Engineers standard J3016), driver is responsible for taking over the vehicle control authority under certain emergency condition. At this moment, the real time driver behaviour and activity monitoring will help to decide whether the driver is able to takeover or not.

According to these, in this study, a vision-based driver activities recognition system is proposed to monitor and understand the driver in real-time. The recognition model is trained to identify seven common driving-related tasks and whether the driver being distracted or not. With this system, the intelligent vehicles can better interact with the human drivers so that the decisions made by the intelligent vehicles can be more reasonable.

6.1.2 Related Works

Driver behaviours has been widely studied in the past two decades. Previous studies mainly focus on driver attention [42] [43] and distraction (either physical distraction or cognitive distraction) [37], driver intention [8] [10], driver drowsiness and fatigue [11]-[13], etc. The National Highway Traffic Safety Administration (NHTSA) defined the driver distraction as a process that the driver shift their attention away from the driving tasks. Four types of distraction are clarified by NHTSA, which are the visual distraction, auditory distraction, biomechanical distraction, and cognitive distraction [38]. To understanding the driver behaviours, most of the studies require capturing driver status

information, such as the head pose [14], eye gaze [9], hand motion [15], foot dynamics [16], and even physiological signals [17] [18].

Specifically, in [19], the video information for driver head movement along with the audio signals were collected to identify the secondary tasks made by drivers. In [20], driver's head pose, eye gaze direction, and hand movement were combined together to identify driver activities. In [21], driver head pose estimation was proposed and applied to the rear-end crash avoidance system. Despite the vision-based driver feature extraction methods, the physiological signals, such as the electroencephalogram (EEG) and electrooculography (EOG) are also widely used for real-time driver status monitoring. In [22], EEG signals were collected to predict driver braking intention. In [23], the EEG and EOG signals were used to monitor driver drowsiness and fatigue status. The EEG signals are proved to be closely related to the driver behaviour and normally show its earlier response to the human mental status compared with the outer behaviours.

However, as mentioned earlier, most of the existing driver behaviour and activity monitoring systems require extracting specific feature vectors such as the head pose angle, gaze direction, and the position of hand and body joints [24]. These features are not always easily to be obtained and some even require specific hardware devices, which will increase the temporal and financial cost. In this work, an end-to-end driver activity recognition system is proposed based on the CNN models. To study the distraction behaviours, the visual distraction, auditory distraction, and biomechanical distraction are involved. In this work, the cognitive distraction is not considered since it has been richly studied in [39] [40], which can be detected without using a vision-based deep learning method.

In terms of the current development of deep learning techniques, significant progress has been made in the computer vision area dues to the development of deep CNN models, parallel computing hardware, and the large-scale annotated dataset. CNN models have achieved the state-of-art results in many object detection, object classification, and segmentation tasks. Meanwhile, it has been successfully applied to the driver monitoring tasks [25] [26]. Therefore, in this work, different CNN models are evaluated for driver activities recognition. The only sensor required in this work is an RGB camera. Based on the report in [27], seven common in-vehicle activities are selected. These tasks contain both normal driving activities and secondary tasks. The CNN models take the processed

images directly without any complex feature extraction. With the transfer learning method, the pre-trained CNN models are fine-tuned to satisfy the behaviour detection task.

6.1.3 Contributions

The contribution of this study can be summarized as follows.

Firstly, a novel CNN model-based method is applied to identify the driver behaviours. Unlike previous literatures that require complex algorithms to estimate the driver status information, such as the head pose, eye gaze, and hand dynamics, etc. The proposed algorithm in this work simply take the colour image as input and directly output the driver behaviour information. By using CNN model, the manually feature extraction process can be avoided and the features will be learned automatically.

Secondly, transfer learning method is applied to the two pre-trained deep CNN models. The models are fine-tuned to deal with both the multiple classification task and the binary classification task. The algorithm is proved an efficient solution for non-intrusive driver activity detection.

Finally, a GMM-based segmentation is applied to the images to extract the driver location from the background. It is found that by increasing the cluster number of the GMM, more precise driver contour can be found from the raw images. By using GMM-based image segmentation method, the model performance on the driving activities recognition increases dramatically compared with using the raw images.

6.2 Experiment and Data Collection

This section describes the experiment design of this study. Figure 6-1. illustrates the general system architecture. The raw images are firstly collected using Kinect. Then, the cropped images are segmented using GMM algorithm. Finally, the CNN models are applied to the activities recognition task.

In this study, the driver behaviour images are collected Kinect. Kinect is able to collect multi-modal signals, such as the colour image, depth image, and audio signals. It was originally designed for indoor human interaction and has been successfully applied to driver monitoring systems [28] [29]. As described in [24], the driver head pose, and upper body joints also can be detected using Kinect, while in this part, only the RGB images are captured.

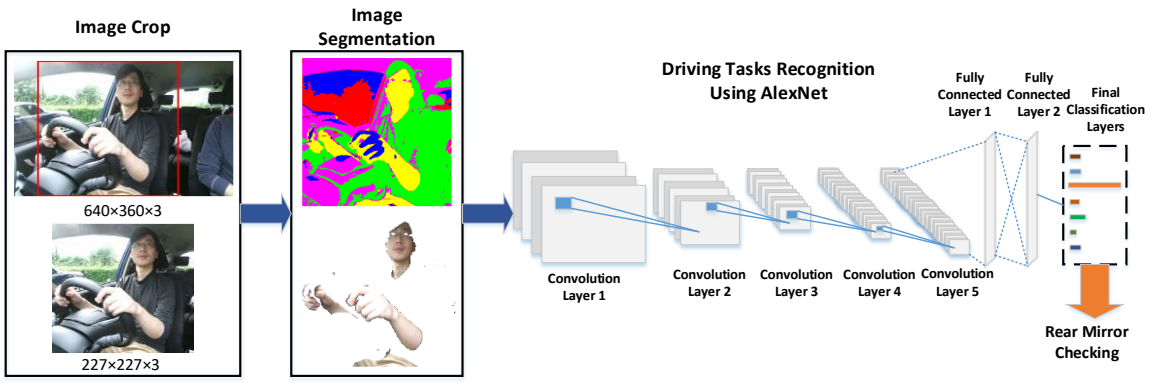

Figure 6-1. Overall System Architecture for driver behaviour detection using AlexNet.

According to the Kinect mounting requirements [30], it is mounted in the middle of the front window, facing the upper body of the driver so that not to interfere driver's field of view while driving. The device setup is shown in Figure 5-3. The sampling rate for the image collection is eight frames per seconds. According to the study in [19], short-term driver behaviours like mirror checking lasts from 0.5 to 1 second. Therefore, the sampling rate is fast enough to capture these behaviours. The data are recorded with an Intel Core i7 2.5GHz CPU and the codes are written in C++ based on the Windows Kinect SDK and OpenCV. To store the images, the raw images are compressed to $640 \times 360 \times 3$ format to increase the computation efficiency.

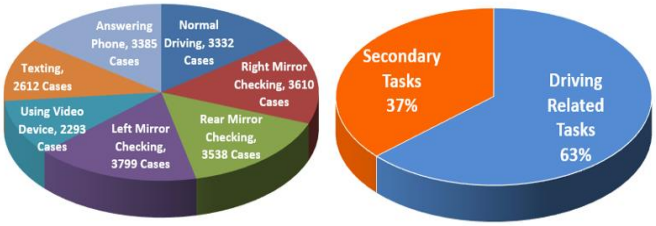

Figure 6-2. Illustration of the collected dataset.

Five drivers are involved in the experiment. They were required to perform seven common activities, which are four normal driving tasks (normal driving, left mirror checking, right mirror checking, and rear mirror checking) and three secondary tasks (using in-vehicle radio/video device, answering mobile phone, and texting). It takes around 20 minutes for each driver to perform all the tasks, and there are 2.4k images are captured in total. The number of images for each task and the quantity comparison between normal driving and secondary tasks are shown in Figure 6-2. Unlike some human activity recognition studies that require temporal information, in this study, no temporal

information are considered and each image is processed individually. The reason is because during driving, driver always perform explicit tasks, such as mirror checking and the secondary tasks. Therefore, no temporal information is required to infer the driver behaviours and each image carries enough information for the activity recognition.

6.3 Methodologies

This section describes the algorithms that used in this study. Specifically, Section III.A introduce the image pre-processing and segmentation based on the GMM algorithm. Section III.B describes the two deep CNN frameworks and the transfer learning methods.

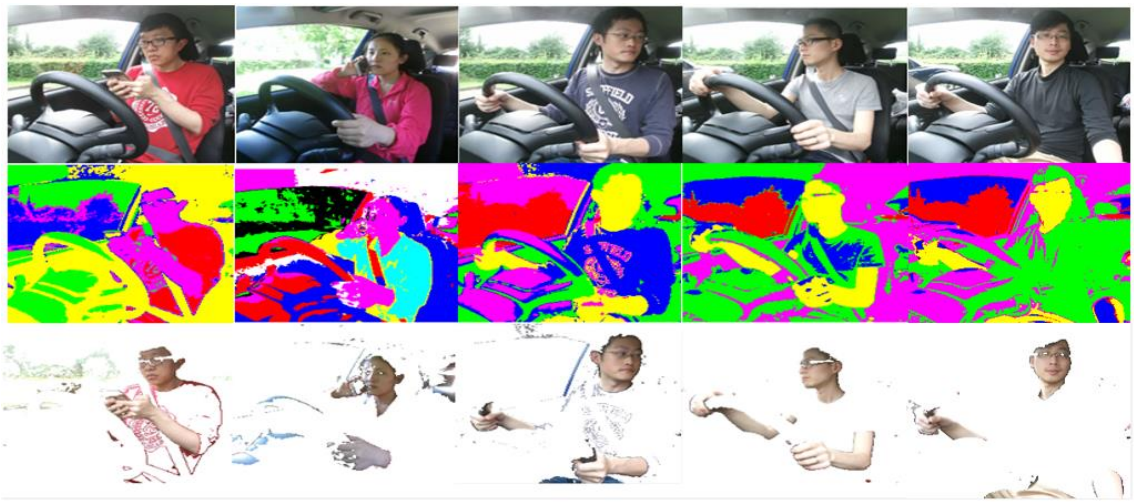

Figure 6-3. Image Segmentation Results using GMM

6.3.1 Image Pre-processing and Segmentation

As aforementioned, the raw images are stored in the format of $640 \times 360 \times 3$. To speed up the CNN training process and increase the classification accuracy, the raw images are cropped in the beginning. An interest of region (ROI) which mainly contains the driver is selected. Figure 6-1. indicates the raw image and the selected ROI. After the raw images are cropped, these images are transformed into the size of $227 \times 227 \times 3$ to satisfy the input requirement of AlexNet and $224 \times 224 \times 3$ for the GoogLeNet, respectively.

Before directly putting the raw colour images into the CNN model, the raw images are firstly processed using the GMM segmentation. The driver body will be detected based on this unsupervised machine learning method in case to have a more precise driver body estimation and behaviour recognition. The main reason of using GMM for image segmentation is the unsupervised machine learning process require no manually labelling,

which is more convenient to use than the supervised learning methods. Although the unsupervised learning results can be less accurate than the supervised learning methods that with accurate ground truth labelling, the classification accuracy of the GMM can be control by choosing the proper number of clustering components. GMM is an unsupervised machine learning method, which can be used for data clustering and data mining. GMM is a probability density function that represented by a weighted sum of sub Gaussian components [31]. As an unsupervised learning method, one of the advantage of using GMM to segment the images is GMM requires no training labels and very flexible to select different number of clusters. To train a GMM for segmentation, each image is represented by a feature vector according to the pixel intensity. The feature vector for the GMM is a three-dimensional vector that contains the RGB intensity of each pixel.

$$x_k = [R(u,v),\ G(u,v),\ B(u,v)] \tag{6-1}$$

where x is the feature vector of a cropped image, k is the index of x which has a maximum value of 227×227 for AlexNet and 224×224 for GoogLeNet. (u, v) is the image coordinates.

The GMM can be represented as the following equation:

$$p(x_i|\theta) = \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sum_k) \tag{6-2}$$

where x_i is the 3-Dimensional feature vector, θ is the parameters of GMM, K is the total number of components in the model (five in this case), π_k is the weight of each Component Gaussian distribution function and the sum of π_k equals to one. μ and $\sum$ are the mean and covariance parameter of multivariate Gaussian function. $N(x_i|\mu_k, \sum_k)$ is the univariate Gaussian distribution function in this case with the form as follows.

$$N(x|\mu, \sum) = \frac{1}{(2\pi)^{D/2}|\sum|^{1/2}} \exp[-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu)] \tag{6-3}$$

where D is the dimension of the data vector. A complicate GMM contains three parameters and can be represented as:

$$\theta = \{\pi_k, \mu_k, \sum_k\} \tag{6-4}$$

The most common method for GMM training is the Expectation-Maximization (EM) maximum likelihood estimation algorithm. It computes the maximization of the cost function in an iterative manner. The detail description of EM algorithm can be found in [32]. Figure 6-3 illustrates the image segmentation results using GMM, driver body and skin can be identified using GMM. After GMM segmentation, only the head and hand

pixels are remained for CNN model training. As shown in next section, this is an efficient method to increase the activity recognition accuracy.

6.3.2 Model Preparation and Transfer Learning

Currently, deep convolutional neural network has gain a tremendous improvement in the domain of computer vision. One of the major reason is the distribution of ImageNet dataset [33]. ImageNet is a large-scale dataset, which contains more than 15 million high-resolution annotated natural images of over 22,000 categories. The large number of annotated images benefit the training of deeper and more accurate CNN models. In this work, two deep convolutional neural network models namely AlexNet and GoogLeNet were chosen as the basic model structures for the recognition of driver behaviour. The AlexNet was first proposed by Alex Krizhevsky in 2012 [34]. The model won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC12) with a much more accurate detection accuracy compared with other models at that time. The model was trained for the classification of 1000 categories. There are five convolutional layers and three fully connected neural network layers with non-linearity and pooling layers between the convolutional layers. In total, AlexNet contains 60 million parameters and 650,000 neurons. The simplified model structure for AlexNet is shown in Figure 6-1.

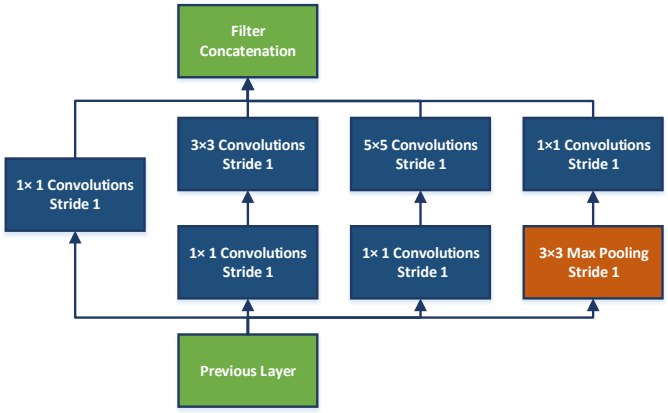

Figure 6-4. Illustration of Inception layer of GoogLeNet.

GoogLeNet is another deep CNN model, which won the ILSVRC14 [41]. GoogleNet is significantly deeper than AlexNet and achieved more accurate classification results on the ImageNet dataset. Despite of the model depth, the main contribution of GoogLeNet is the utilization of Inception architecture. As mentioned in [41], the most common ways of increasing CNN model performance are to increase the network size (either the depth

or the width of the model). However, it gives rise to the requirement for larger scale dataset and more computational burden. They introduce the Inception layers into the CNN model to increase the sparsity among the layers and reduce the number of parameters. Each Inception layers consists of six basic convolution filters and one max pooling filter. With different scales, the parallel-arranged convolution filters will have more accurate detailing and a wider representation for the information from previous layers. A dimension reduction Inception layer is shown in Figure 6-4. In total, there are two traditional convolutional layers at the lower level of the GoogLeNet and nine Inception layers are concatenated at higher level. With the application of Inception layers, the general quantity of parameters in GoogLeNet is 12 times less than that in the AlexNet.

To train the deep convolutional neural networks like AlexNet and GoogLeNet from scratch, a large-scale annotated dataset like ImageNet is need. However, In general, large-scale annotated datasets are not always available. Therefore, common ways to use the pre-trained deep CNN model are to treat the model as a fixed feature extractor without tuning the model parameters or fine-tune the pre-trained model parameters with limited dataset. In this study, the CNN models are used in the second manner, which is to fine tune a few layers of the models with the driver behaviour dataset. Since the two models are trained to classify 1000 categories in the beginning, the last few layers have to be modified so that the model can satisfies the seven objects classification task. In this work, the last three layers of the AlexNet and GoogLeNet are modified.

Specifically, the original last fully connected layer and the output layer, which generate the probabilities for the 1000 categories, are replaced by a new fully connected layer and softmax layer that output the probabilities for the seven categories. The basic structure and properties of the convolutional layers are remained so that these layers can keep their advantages in the feature extraction. Meanwhile, those knowledge that learned from the large-scale ImageNet dataset can be transferred to the driver behaviour domain. For transfer learning, a small initial learning rate is selected to slow down the updating rate of the convolutional layers. On contrary, a larger learning rate factors for the last fully connected layer is chosen to speed up the learning rate in the final layers. With this kind of combination, the new models can be trained to satisfy the new classification tasks.

6.4 Experiment Results and Analysis

In this section, the classification results for the driving tasks are proposed. The system performances are evaluated on three major aspects: the impact of GMM-based image segmentation on the recognition of driver behaviours, the classification results compared with the feature extraction methods, and the binary classification results on the distracted behaviours.

6.4.1 The Impact of GMM Image Segmentation on Driving Tasks Recognition

Firstly, the activities recognition for the five participants are evaluated. The seven driving related tasks are ordered as {normal driving, right mirror checking, rear mirror checking, left mirror checking, using radio/video device, texting, and answering mobile phone}. Table 6-1 illustrates the seven objects classification results using AlexNet and Table 6-2 indicates the results given by the GoogLeNet. Specifically, the upper parts of Table 6-1 and Table 6-2 indicates the classification results with GMM image segmentation, while the lower part illustrates the classification results using the raw RGB images. T1 to T7 represents the seven tasks and D1 to D5 represents the five drivers. The models are trained using MATLAB Deep Learning toolbox. The model is evaluated using LOO cross-validation. To get the activities identification results for a certain driver, the images for this driver are only used as testing images, whereas the images for the rest four drivers are used as training images. Therefore, for these drivers, their images are completely new to the testing CNN models and the identification performances equals to the model generalization on the new dataset.

TABLE 6-1 CLASSIFICATION RESULTS FOR DRIVING TASKS RECOGNITION WITH ALEXNET

	Driving Tasks				Non-Driving Tasks			
GMM	T1	T2	T3	T4	T5	T6	T7	Ave
D1	0.921	0.990	0.929	0.398	0.904	0.897	0.999	0.831
D2	0.996	0.906	0.920	0.449	0.595	0.242	0.936	0.753
D3	0.889	0.989	0.176	1.00	0.982	0.982	0.732	0.832
D4	0.353	0.994	0.229	0.813	1.00	0.982	0.979	0.752
D5	0.958	0.998	0.934	0.236	0.666	0.816	0.998	0.786
Mean	0.823	0.975	0.637	0.579	0.829	0.783	0.929	0.791
Raw	T1	T2	T4	T4	T5	T6	T7	Ave
D1	0.188	0.999	0.998	0.00	0.00	0.942	0.893	0.534
D2	0.593	0.949	0.01	0.00	0.789	0.215	0.957	0.408
D3	0.227	0.779	0.00	1.00	0.843	1.00	0.532	0.623
D4	0.00	1.00	0.00	0.875	1.00	0.98	0.00	0.482
D5	0.292	0.996	1.00	0.00	0.787	1.00	0.993	0.657
Mean	0.260	0.944	0.402	0.375	0.683	0.827	0.675	0.541

TABLE 6-2 CLASSIFICATION RESULTS FOR DRIVING TASKS RECOGNITION WITH GOOGLENET

	Driving Tasks				Non-Driving Tasks			
GMM	T1	T2	T3	T4	T5	T6	T7	Ave
D1	0.937	0.05	0.980	0.957	0.993	0.926	1.00	0.832
D2	1.00	0.902	0.974	0.114	0.953	0.242	0.880	0.732
D3	0.281	0.989	0.456	0.713	1.00	0.987	0.844	0.742
D4	0.562	0.245	0.196	0.960	0.999	0.990	0.843	0.655
D5	0.989	0.998	0.975	0.230	0.515	0.268	0.995	0.715
Mean	0.754	0.637	0.716	0.594	0.892	0.683	0.912	0.735
Raw	T1	T2	T4	T4	T5	T6	T7	Ave
D1	0.711	0.997	0.977	0.721	0.01	0.909	0.995	0.777
D2	0.982	0.902	0.632	0.729	0.813	0.121	0.990	0.734
D3	0.965	0.926	0.00	0.00	0.994	0.991	0.346	0.588
D4	0.00	0.164	0.03	0.979	0.940	0.997	0.775	0.524
D5	0.764	0.999	0.313	0.00	0.00	1.00	0.970	0.588
Mean	0.684	0.798	0.390	0.486	0.551	0.803	0.815	0.642

As shown in Table 6-1, the general identification accuracy for the GMM segmentation-based AlexNet achieved an average of 79.1% accuracy, which is much better than that without using image segmentation (50.3%). The average performance is defined as the average detection results for the five drivers while the mean accuracy represents the average detection rate on each task. In terms of the accuracy on each task, the right mirror checking gets the most accurate detection result among the five drivers, and answering mobile phone achieved the second-best result. The worst result happens in the left mirror checking case (the far-end mirror), both for using GMM segmentation and using raw images. Another evidence that can be drawn from Table 6-1and Table 6-2 is that the CNN model achieved better detection rate on the secondary tasks. This mainly due to the fact that, when performing secondary tasks, the driver has to move the body and hands instead of only rotating his/her head, which makes the detection easier. Similar results are obtained in [24].

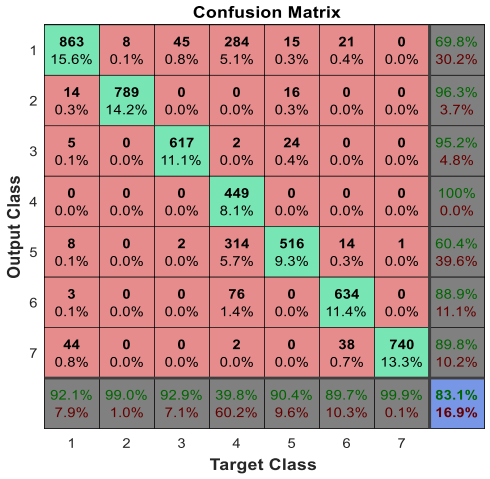

Figure 6-5. Confusion matrix for driver 1 using AlexNet.

Table 6-2 indicates the activity classification results given by the GoogLeNet. The general results are similar to the results in Table 6-1 except that the overall detection accuracy for the five drivers are slightly lower. The GoogLeNet does not achieve a better classification results than the AlexNet on this dataset as it does on the ImageNet dataset. However, the classification results for the GoogLeNet trained with raw images are significantly better than that in the AlexNet case. The general classification results for the GoogLeNet is 64.2% accuracy, which is 10% higher than that for the AlexNet.

Figure 6-5. illustrates the confusion matrix for driver 1 using the AlexNet and GMM segmentation. The green diagonal shows the correct detection cases for the class. The bottom row shows the classification accuracy with respect to the target class, while the right most column shows the classification accuracy with respect to the predicted labels. As shown in Figure 6-5, all the driving tasks except the fourth task (left mirror checking) achieved reasonable detection rates. In terms of the fourth task, there are 284 cases are misclassified into the normal driving case and 314 cases are misclassified into the video device setup group.

Figure 6-6. Illustration of texting behaviour of driver 3.

To have a better understanding about the CNN models and how the model response to the segmented images, the convolutional layer activation and its features are analysed in the following. Figure 6-6 shows the raw image and the segmented image for in-vehicle texting behaviour of one participant. Figure 6-7. shows different activation maps from the CNN models based on the images in Figure 6-6. As shown in the left part of the dash line in Fig. 8, the top activation maps are given by the relu5 layer of the AlexNet models. Specifically, the left one is based on the AlexNet with GMM segmentation (G-AlexNet model), whereas the right one is based on the AlexNet with raw images (R-AlexNet model). The bottom images are the corresponding strongest activation channel. As shown in the Figure 6-7, the relu5 layer for the G-AlexNet model remains much more features

than that in the R-AlexNet. The GMM segmentation extract the driver from the background so that the CNN model can maintain more relevant features for the driver. The strongest activation for the G-AlexNet model keeps the driver head rotation features and other channels keep the arm position information. Figure 6-8. shows the visualized features for the AlexNet models. The G-AlexNet model learns more contrast filters, while the filters in R-AlexNet are over-smooth, especially can be found in the last fully connected layer.

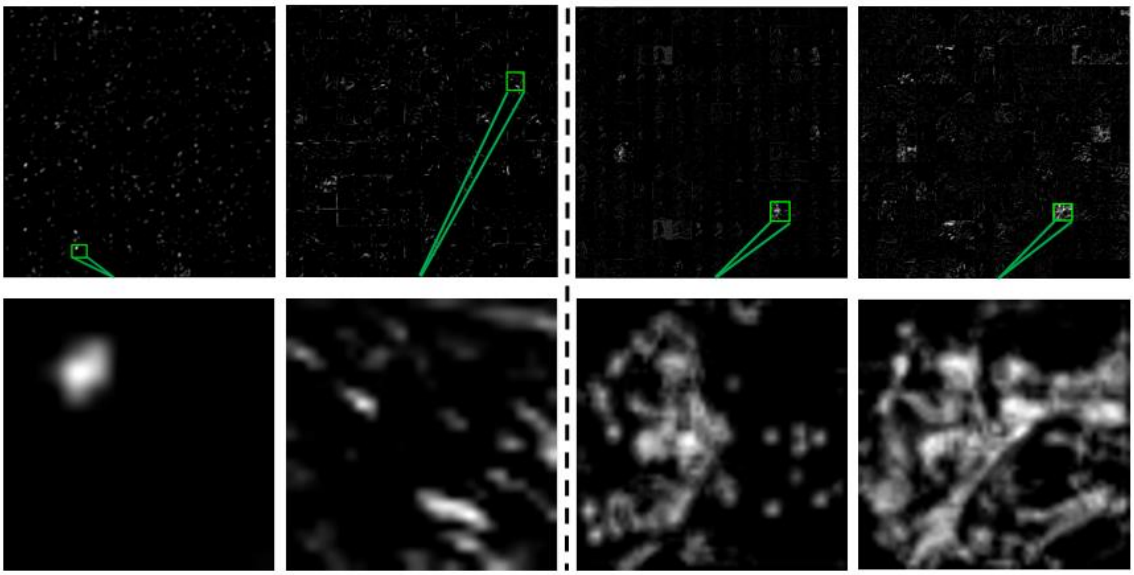

Figure 6-7. Activation map of the two CNN models.

Since GoogLeNet is much deeper than the AlexNet and the deeper Inception layers in the GoogLeNet are complex, in this part, only the activation from the second convolutional layer is analysed. The right part of Figure 6-7. indicates the activation of the conv2-relu layer in the GoogLeNet. As shown in the figure, the GoogLeNet trained with segmented images (G-GoogLeNet) maintain more driver related features instead of the background features. The GoogLeNet model trained with raw images (R-GoogLeNet) cannot give a clear driver position. As driver-related features are not well maintained in the beginning layers, the deeper Inception layers also cannot learn a representative feature for the driver. Based on this, the activation maps explain why the GMM segmentation-based CNN models lead to a better classification results than the raw image-based models.

The left parts of Figure 6-7 is given by the AlexNet model and the right part is given by the GoogLeNet. From left to right, the first activation map is the relu5 layer activation of GMM-AlexNet model, the second map is the relu5 layer of Raw-AlexNet model, the

third one is the conv2-relu layer of GMM-GoogLeNet, and the last one is the same layer activation map for the Raw-GoogLeNet model. The lower part images are the corresponding strongest activation channel.

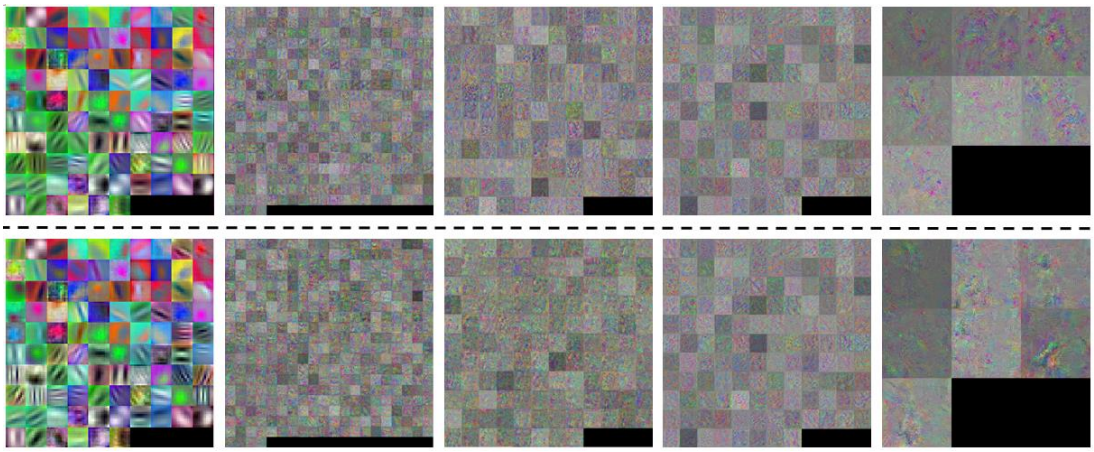

Figure 6-8. Feature visualization for AlexNet.

The upper part of indicates the features for the model trained with segmented images while the lower part is given by the model trained with raw images. From left to right, the feature maps are given by the conv1 layer, conv3 layer, conv5 layer (1-128 filters), conv5 layer (129-256), and final fully connected layer, respectively.

Finally, the time cost for the training and testing of different models are indicated in Table 6-3. As shown in Table 6-3, the general training for GoogLeNet is two times longer than the AlexNet training time. It takes about 12ms to process one image for the AlexNet while the testing time for the GoogLeNet are 45ms. The model training and testing are based on Core i7 2.5GHz CPU and NVIDIA Quadro K1100M 2GB GPU.

TABLE 6-3 TRAINING AND TESTING TIME COST FOR EACH MODEL

	G-AlexNet	R-AlexNet	G-GoogLeNet	R-GoogLeNet
Training Time (s)	~1100	~1200	~2400	~2400
Testing Time per Image (ms)	~13	~12.5	~45	~45

6.4.2 Results Comparison between Transfer Learning and Feature Extraction

As discussed in last section, a pre-trained CNN model normally can be used in two different manners. The first way is to fine-tune the last few layers so that the model can satisfy the new task. The second common usage is treating the pre-trained CNN model as a feature extractor [35]. As introduced in [36], the HOG and SVM method has been one of the most successful method for some image-based feature extraction and classification tasks. Therefore, the HOG and SVM is adopted as one of the conventional method.

Meanwhile, as some of the studies have used the pre-trained CNN as the feature extractor, in this part, the feature set from the conv5 layer of AlexNet is also adopted. The conventional feature extraction and classification scheme will be compared with the proposed end-to-end driver behaviour classification algorithm in case to show the advantage of the proposed method. The deep CNN models are trained on the large-scale dataset, which makes the convolutional layers have a strong representation ability of the objects. The lower level of the convolutional layers is more concentrate on the local features, such as edges, colours, and corners. While the deeper convolutional layers will focus on higher-level features. The combination of feature extraction and conventional machine learning algorithms like Support Vector Machine (SVM) is more convenient to use [35]. Therefore, in this part, the performance of vision-based feature extraction method for driving activities recognition will be evaluated and compared with the transfer learning methods.

Two different features are extracted based on the segmented images. The first feature set is generated using Histogram of oriented gradients (HOG) [36]. A pyramid HOG feature extractor is used, which concatenates two different scale HOG extractors. The block size for the HOG feature extractors is 2×2 and the cell sizes are 8×8 and16×16, respectively. The visualization for the HOG features are illustrated in Figure 6-9. The second features set is generated by the fifth convolutional layer of the AlexNet. The Principal Component Analysis (PCA) algorithm is used to reduce the feature set dimension for the two feature sets. In this work, the dimension for the feature vector is reduced to 50.

Meanwhile, two different classification models are evaluated, which are the SVM and FFNN. The classification results are shown in Table 6-4. In Table 6-4, S+C means SVM and Conv5 feature. S+H is SVM and HOG features. Similarly, F+C means FFNN and Conv5 feature. F+H represents FFNN and HOG features. As shown in Table 6-4, the feature extraction methods are unable to accurately identify the driving tasks as the transfer learning method does. The average results for the four combinations are much lower compared with transfer learning method. Hence, transfer learning is proved to be more suitable for this task. Graph (a) of Figure 6-9 indicates the original segmented image; graph (b) represent the HOG feature map with 8×8 cell, and the graph (c) is the feature map with 16×16 cell.

TABLE 6-4 CLASSIFICATION RESULTS USING FEATURE EXTRACTION

	D1	D2	D3	D4	D5	Ave
S+C	0.497	0.344	0.331	0.236	0.249	0.331
S+H	0.390	0.215	0.184	0.240	0.196	0.245
F+C	0.304	0.298	0.537	0.250	0.289	0.335
F+H	0.428	0.480	0.454	0.173	0.136	0.334

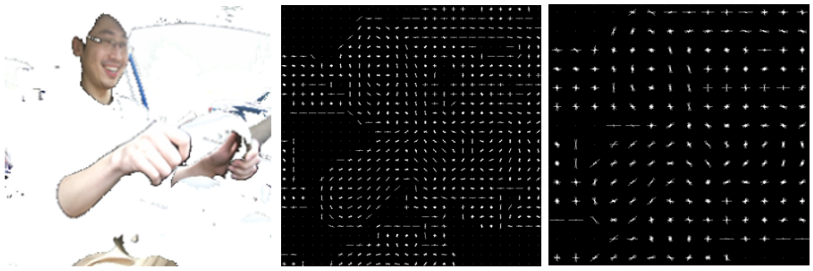

Figure 6-9. HOG feature visualization map.

6.4.3 Driver Distraction Detection Using Binary Classifier

Lastly, the AlexNet and GoogLeNet are modified and trained to detect whether the driver is distracted or not. At this moment, the first four tasks are grouped together, and the last three tasks constitute another group. Unlike previous step where the CNN model is designed for multi-class classification, in this step, the last fully connected layer is tuned to solve the binary classification problem. The distraction detection results for AlexNet and GoogLeNet are shown in Table 6-5 and Table 6-6, respectively.

TABLE 6-5 BINARY CLASSIFICATION RESULTS USING ALEXNET

		D1	D2	D3	D4	D5	Mean
GMM_ AlexNet	Normal	0.991	0.944	0.999	0.944	1.00	0.976
	Distract	0.776	0.997	0.715	0.959	0.992	0.888
	Ave	0.913	0.961	0.890	0.951	0.997	0.942
AlexNet	Normal	0.538	0.213	0.133	0.674	1.00	0.512
	Distract	0.732	1.00	0.999	0.899	0.644	0.855
	Ave	0.608	0.462	0.467	0.771	0.886	0.639

TABLE 6-6 BINARY CLASSIFICATION RESULTS USING GOOGLENET

		D1	D2	D3	D4	D5	Mean
GMM_ GoogLeNet	Normal	0.993	0.979	0.989	0.409	0.999	0.873
	Distract	0.814	0.987	0.977	0.935	0.926	0.923
	Ave	0.928	0.982	0.985	0.634	0.976	0.898
GoogLeNet	Normal	0.995	0.913	0.375	0.09	1.00	0.645
	Distract	0.633	0.984	0.998	0.962	0.991	0.913
	Ave	0.863	0.935	0.615	0.464	0.997	0.775

In Table 6-5, the upper part indicates the detection results based on the GMM image segmentation and CNN model. On the contrary, the lower part indicates the detection results using raw images. As shown in Table 6-5, the GMM-CNN based method lead to a much more accurate result than that using raw image. The general classification

accuracy for the GMM-CNN based method is 94.2%, which is significantly higher than the raw image-based method (only achieved 63.9% accuracy).

Similar results can be found in Table 6-6, where the GMM-GoogLeNet based method also achieved a better result than that using raw image-based method. The general classification result for GMM-GoogLeNet based method achieved 89.8% accuracy, which is slightly lower than the result given by the GMM-AlexNet. However, the distraction detection results (92.3%) for the GoogLeNet are better than that in the AlexNet case (88.8%). One significant low accurate for the Normal driving tasks detection occurs at the fourth driver case, which only achieved 40.9% detection rate. If ignoring this significant low sample, the detection rate would increase to 99% accuracy. It should be pointed out that the fourth driver is the only female driver in this experiment. The gender issue may be a potential explanation for this low detection rate. If ignoring the fourth driver, the GoogLeNet achieved a better classification results on this binary case. Another difference between GoogLeNet and AlexNet is that GoogLeNet achieved better detection result (77.5%) for the raw image-based case, which is about 13% higher than that in the AlexNet case.

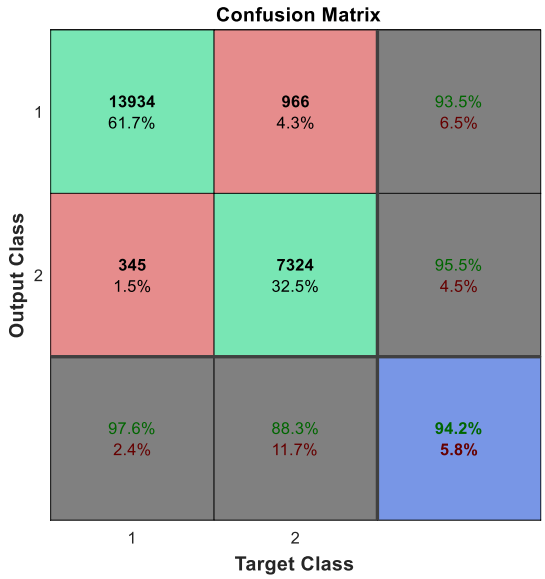

Figure 6-10. Confusion matrix for the binary classification result using AlexNet.

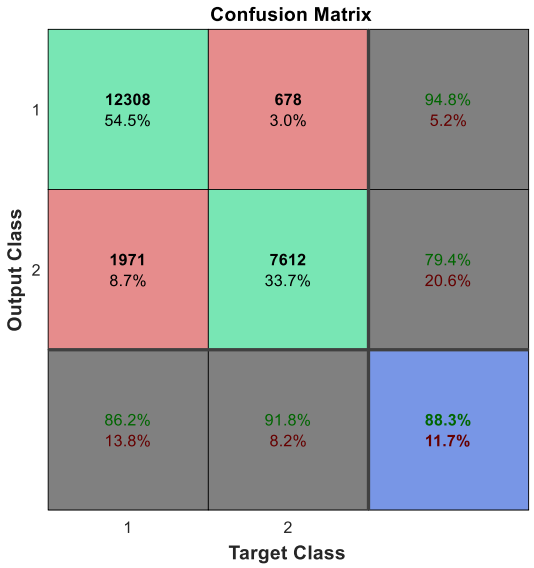

Figure 6-11. Confusion matrix for the binary classification result using GoogLeNet.

The confusion matrix for the GMM-CNN based methods are shown in Figure 6-10 and Figure 6-11, respectively. It should be noticed that, in these cases, there are no smoothing schemes are applied to the distraction warning module. In most situations, the driver assistance system will only warn the driver if distraction cases happen continuously in a short period. Therefore, if applying short period smoothing or voting techniques, the distraction detection system can be more suitable to real-world application.

6.5 Conclusions

In this work, a driving related activities recognition system based on CNN model is proposed. The deep CNN models are trained with transfer learning method. Specifically, the pre-trained AlexNet and GoogLeNet are fine-tuned to satisfy the seven driving tasks recognition task. To increase the identification accuracy, the raw RGB images are first processed with a GMM-based segmentation algorithm. The GMM method can efficiently identify the drivers from the background environment and remove the irrelevant objects. The classification results indicate that the GMM segmentation-based CNN models gives a much precise detection result. Another comparison is made between the transfer learning and feature extraction method. Results show that the HOG and CNN features are not able to accurately identify the driver behaviours. Finally, if using the CNN models as a binary classifier, the driver distraction detection rate can achieve 94% accuracy.

Although the multiple classification accuracy in this work is no better than the result in [24], the end-to-end transfer learning scheme only require RGB images as input. It does not need complicate feature extraction and selection process, which makes it much easier to be implemented in real-time.

Future works will concentrate on the further improvement of the behaviour detection rate. More drivers and data are expected to increase the system robustness and detection accuracy. The system can be also tested and used for driver/passenger behaviour monitoring system on the partially automated vehicles.

6.6 Reference

[1] Wang, Fei-Yue, *et al*. "Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence." IEEE/CAA Journal of Automatica Sinica 4.4 (2017): 577-587.

[2] Wang, Jinxiang, Junmin Wang, Rongrong Wang, and Chuan Hu. "A Framework of Vehicle Trajectory Replanning in Lane Exchanging With Considerations of Driver Characteristics." IEEE Transactions on Vehicular Technology 66, no. 5 (2017): 3583-3596.

[3] Koesdwiady, Arief, *et al*. "Recent Trends in Driver Safety Monitoring Systems: State of the Art and Challenges." IEEE Transactions on Vehicular Technology 66.6 (2017): 4550-4563.

[4] Wang, Xiao, Rui Jiang, Li Li, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. "Capturing Car-Following Behaviors by Deep Learning." IEEE Transactions on Intelligent Transportation Systems (2017).

[5] Zeng, Xiangrui, and Junmin Wang. "A stochastic driver pedal behaviourmodel incorporating road information." IEEE Transactions on Human-Machine Systems 47.5 (2017): 614-624.

[6] Wang, Fei-Yue, *et al*. "Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence." IEEE/CAA Journal of Automatica Sinica 4.4 (2017): 577-587.

[7] Lv, Chen, *et al*. "Analysis of autopilot disengagements occurring during autonomous vehicle testing." IEEE/CAA Journal of Automatica Sinica 5.1 (2018): 58-68.

[8] Butakov, Vadim A., and Petros Ioannou. "Personalized driver/vehicle lane change models for ADAS." IEEE Transactions on Vehicular Technology 64.10 (2015): 4422-4431.

[9] Gou, Chao, *et al*. "A joint cascaded framework for simultaneous eye detection and eye state estimation." Pattern Recognition 67 (2017): 23-31.

[10] McCall, Joel C., *et al*. "Lane change intent analysis using robust operators and sparse bayesian learning." IEEE Transactions on Intelligent Transportation Systems 8.3 (2007): 431-440.

[11] Hu, Jie, *et al*. "Abnormal Driving Detection Based on Normalized Driving Behavior." IEEE Transactions on Vehicular Technology 66.8 (2017): 6645-6652.

[12] Chai, Rifai, *et al*. "Driver fatigue classification with independent component by entropy rate bound minimization analysis in an EEG-based system." IEEE journal of biomedical and health informatics 21.3 (2017): 715-724.

[13] Mandal, Bappaditya, *et al*. "Towards detection of bus driver fatigue based on robust visual analysis of eye state." IEEE Transactions on Intelligent Transportation Systems 18.3 (2017): 545-557.

[14] Murphy-Chutorian, Erik, and Mohan Manubhai Trivedi. "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness." IEEE Transactions on intelligent transportation systems 11.2 (2010): 300-311.

[15] Das, Nikhil, Eshed Ohn-Bar, and Mohan M. Trivedi. "On performance evaluation of driver hand detection algorithms: Challenges, dataset, and metrics." Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on. IEEE, 2015.

[16] Tran, Cuong, Anup Doshi, and Mohan Manubhai Trivedi. "Modeling and prediction of driver behaviourby foot gesture analysis." Computer Vision and Image Understanding 116.3 (2012): 435-445.

[17] Bi, Luzheng, *et al*. "Queuing Network Modeling of Driver EEG Signals-Based Steering Control." IEEE Transactions on Neural Systems and Rehabilitation Engineering 25.8 (2017): 1117-1124.

[18] Teng, Teng, Luzheng Bi, and Yili Liu. "EEG-Based Detection of Driver Emergency Braking Intention for Brain-Controlled Vehicles." IEEE Transactions on Intelligent Transportation Systems (2017).

[19] Li, Nanxiang, and Carlos Busso. "Detecting drivers' mirror-checking actions and its application to manoeuvre and secondary task recognition." IEEE Transactions on Intelligent Transportation Systems 17.4 (2016): 980-992.

[20] Ohn-Bar, Eshed, *et al*. "Head, eye, and hand patterns for driver activity recognition." Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014.

[21] Rezaei, Mahdi, and Reinhard Klette. "Look at the driver, look at the road: No distraction! No accident!" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[22] Kim, Il-Hwa, *et al*. "Detection of braking intention in diverse situations during simulated driving based on EEG feature combination." Journal of neural engineering 12.1 (2014): 016001.

[23] Zhang, Chi, Hong Wang, and Rongrong Fu. "Automated detection of driver fatigue based on entropy and complexity measures." IEEE Transactions on Intelligent Transportation Systems 15.1 (2014): 168-177.

[24] Yang, X, Lv, C., Cap, D., *et al*. Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition. IEEE Transactions on Computational Social Systems 99 (2018):1-14.
[25] Le, T. Hoang Ngan, *et al*. "DeepSafeDrive: A grammar-aware driver parsing approach to Driver Behavioral Situational Awareness (DB-SAW)." Pattern Recognition 66 (2017): 229-238.
[26] Li, Li, *et al*. "Intelligence testing for autonomous vehicles: a new approach." IEEE Transactions on Intelligent Vehicles 1.2 (2016): 158-166.
[27] Sivak M, Schoettle B. Motion sickness in self-driving vehicles. UMTRI-2015-12. (2015). Ann Arbor: University of Michigan Transportation Research Institute.
[28] Gaglio, Salvatore, Giuseppe Lo Re, and Marco Morana. "Human activity recognition process using 3-D posture data." IEEE Transactions on Human-Machine Systems 45.5 (2015): 586-597.
[29] Neto, Laurindo Britto, *et al*. "A Kinect-based wearable face recognition system to aid visually impaired users." IEEE Transactions on Human-Machine Systems 47.1 (2017): 52-64.
[30] Mehran Azimi. Skelton Joint Smoothing White Paper [Online]. Available: https://msdn.microsoft.com/en-us/library/jj131429.aspx.
[31] Rasmussen, Carl Edward. "The infinite Gaussian mixture model." Advances in neural information processing systems. 2000.
[32] Xu, Lei, and Michael I. Jordan. "On convergence properties of the EM algorithm for Gaussian mixtures." Neural computation 8.1 (1996): 129-151.
[33] Deng, Jia, *et al*. "Imagenet: A large-scale hierarchical image database." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.
[34] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[35] Deo, Nachiket, Akshay Rangesh, and Mohan Trivedi. "In-vehicle Hand Gesture Recognition using Hidden Markov models." Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. IEEE, 2016.
[36] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
[37] Tawari, Ashish, *et al*. "Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking." Intelligent Vehicles Symposium Proceedings, 2014 IEEE. IEEE, 2014.
[38] Ranney, Thomas A., *et al*. "NHTSA driver distraction research: Past, present, and future." Driver distraction internet forum. Vol. 2000. 2000.
[39] Liao, Yuan, *et al*. "Detection of driver cognitive distraction: A comparison study of stop-controlled intersection and speed-limited highway." IEEE Transactions on Intelligent Transportation Systems 17.6 (2016): 1628-1637.
[40] Liu, Tianchi, *et al*. "Driver distraction detection using semi-supervised machine learning." IEEE transactions on intelligent transportation systems 17.4 (2016): 1108-1120.
[41] Szegedy, Christian, *et al*. "Going deeper with convolutions." CVPR, 2015.
[42] Pugeault, Nicolas, and Richard Bowden. "How much of driving is preattentive?." IEEE Transactions on Vehicular Technology 64.12 (2015): 5424-5438.
[43] Martinez, Clara Marina, *et al*. "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey." IEEE Transactions on Intelligent Transportation Systems (2017).

PART V:

DRIVER LANE CHANGE MANOEUVRE. Intention inference

7

PAPER VI – Driver Lane Change Intention Inference

Driver Lane Change Intention Inference: Framework, Algorithms, Testing Results, and Analysis

Authors:

Yang Xing, Chen Lv, Huaji Wang, Hong Wang, Dongpu Cao, Efstathios Velenis, Fei-Yue Wang

Abstract

With the development of ADAS and the intelligent vehicles, drivers need to share their control authorities with the intelligent control units. It is critical important for the ADAS to understand the drivers' intent and their following manoeuvres. Therefore, in this study, an intention inference system which particular focus on the lane change manoeuvre on the highway is proposed. Firstly, a general driver intention mechanism and framework are introduced. Then, the vision-based intention inference system, which use multiple low-cost cameras and vehicle data acquisition system to capture the multi-modal sensory inputs are proposed. The vision system is designed to simultaneously capture both the in-vehicle and outer-vehicle features. To predict the driver intention, a deep RNN with LSTM units is proposed to deal with the time-series driving data and temporal pattern. The experiment data are collected with multiple drivers on highway environment. A statistical analysis for the lane change manoeuvre sequences is performed. It is found that drivers tend to perform mirror checking 6s prior to the lane change manoeuvre, and the time interval between steering the wheel and cross the lane is about 2s. Before driver initiate a lane change manoeuvre, the RNN model achieves 95% inference accuracy for the lane change intention.

Index Terms - Lane change, driver intention, ADAS, RNN, LSTM, intelligent vehicle.

7.1 Introduction

7.1.1 Motivation

Traffic accidents can give rise to millions of injuries and death each year in worldwide. Most of the traffic accidents are caused by human misbehaviours, such as the driver cognitive overload, judgement mistake, and operation errors [1]-[3]. Since divers are within the centre of the TDV loop, a proper understanding of driver behaviours and the driving-related intention can largely reduce the amount of traffic accidents [4] [5]. In the past two decades, a large number of ADAS products have been implemented on the commercial vehicles. Currently, most of the ADAS products, such as LDW [6] [7], ACC [8], and SWA [9] are designed to provide additional traffic context information to assist the driver during driving. Although these products can be treated as active safety systems, these functions interact with the human driver in a passive manner, which fail to monitor and understand the driver in real-time. The active interaction between human driver and the intelligent units are the major object for the next generation ADAS products [10] [11].

Therefore, in this study, a driver behaviour monitoring system towards the lane change intention inference is proposed. The reasons to understand human drivers and infer their intentions are summarized as follows. Firstly, the main reason for driver lane change intention inference (LCII) is to increase the driving safety. Driver intention inference enable the ADAS or the intelligent vehicles focus on the potential context as early as possible. Hence, the chance that the driver assistance system prevent accident from happening would largely increase. Meanwhile, as reported in [12], the turn signals are only used in 66% lane changes and less than 50% of the turn indicator activation happens in the initial phase of the lane change manoeuvre. Therefore, the driver intention inference system would benefit the ADAS to efficiently interact with human drivers. Since most of the ADAS tend to share vehicle control authority with the drivers, the LCII system is also helpful to decrease the conflicts between the driver and the vehicle. Finally, a hotpot of the intelligent and automated vehicle study is to design human-like decision making and vehicle control algorithms. A better understanding of human driver intention mechanism will contribute to form a more naturalistic on-board decision system for the automated vehicles. In addition, LCII has close relationship with driver behaviour imitation [13] [14], which can accelerate the development of the parallel automated vehicle evaluation system [15].

7.1.2 Literature Review

Driver intention inference was widely studied in the past two decades from different aspects. Liu and Pentland employed HMM to predict several driver intentions on a car simulator [16]. At that moment, only the vehicle status data such as the steering angle, steering velocity, and the vehicle velocity were used. They reported an average of 88.3% detection rate was achieved after 0.5 second when the action was initiated. As discussed in the next Section, the vehicle status data are hardly to be used for intention prediction but can be used for intention recognition. After that, Oliver and Pentland proposed another intention prediction model using Coupled Hidden Markov model (CHMM) [17]. The driver behavioural signals like head pose and eye gaze were included a long with the front lane and vehicle positions. The CHMM model gave a high prediction accuracy for the start and stop intention one second before the any significant change in the car, while they achieved 29.4% and 6.3% detection rate for the lane change left and right manoeuvre, respectively. The detection rates for lane changes are unreliable in real-life at that moment. After these early studies, it was found that a precise intention inference system should rely on a holistic approach, which need to fuse the multi-modal data in the TDV loop together [18] [33].

The signals from the TDV loop for lane change intention come from different sensors. For the traffic context, the most widely used signals are the lane markings, surrounding vehicles positions, digital maps, and the GPS, etc [19]-[21]. Vision-based LDW also provides the distance to the lane boundaries, vehicle position, vehicle yaw angle, and road curvature. The driver behavioural signals consist of the head motion, eye gaze, body gestures, and even the EEG, etc. [22]-[25]. The vehicle status signals are normally collected from the CAN, which contains vehicle speed, acceleration, steering wheel angle and velocity, turn signal, and pedal position etc. [12][26][27].

Rafael introduced an IMM approach to predict lane change on highway. GPS/IMU sensors are used to collect the highway context data [28]. The GPS device can give the location and time information in complex weather conditions, which is more robust than the camera and Lidar devices. Salvucci introduced a four-step based lane change intention detection system, which contains data collection, model simulation, action tracking and thought inference [27] [29]. The steering wheel angle, accelerator, and front vehicle position were used as the model input. Schmidt and Beggiato proposed a lane change

intention recognition method based on performing an explicit steering wheel angle mathematical model [30]. In [12], early driver intention was detected by observing the easily accessible vehicle and traffic signals. The pedal positions and global vehicle position on a digital map were used to identify an intention at a very early stage after the lane change was started. Campbell and Bajcsy use multiple discriminative models to identify three kinds of driver intention, which were lane keeping, prepare for lane changing and lane changing [31]. Henning *et al*. proposed a lane change intention recognition system, which focused on the analysis of the impact of the environmental indicator on the prediction of lane change intention [32].

Driver behavioural signals such as the head and eye movement can give earlier clues about the driver intention. Many studies have evaluated their influence on the intention prediction problem [34]-[36]. In [37], the authors used the pupil information as the cognitive signals to predict the lane change intent. In [38], the authors proposed a head tracking system for driver lane change intention inference based on the glance area estimation. Zhou *et al.* evaluated how driver behaviour being affected by the cognitive distraction during lane change preparation process through the analysis of driver eye movement [39]. They concluded that a secondary task could affect the intention inference accuracy. In [40], the authors constructed a queuing network based cognitive architecture to modelling the driver behaviour during normal/emergency lane change manoeuvres and the lane keeping. Comparing with those methods that based on the eye gaze and head moment estimation, this method can be easily extended into real-world vehicles with a low cost. In [41], the authors used an HMM as driver lane change intention prediction. A new feature named CDI was introduced to represent the surrounding context of the host vehicle and the lane change willing level. Li *et al*. proposed an integrated intention inference algorithm based on HMM and BF techniques [42]. A preliminary output from the HMM was further filtered using the BF method to make the final decision. The HMM-BF framework achieved a recognition accuracy of 93.5% and 90.3% for the right and left lane change, respectively. In [43], the authors proposed a lane change detection method based on the object-oriented Bayesian networks (OOBN). The system was designed according to the modularity and reusability of the Bayesian network, which makes it easy to extend the system according to different requirements.

McCall and Trivedi designed a lane change intent inference system using a sparse Bayesian learning (SBL) methodology [44]. The lane tracking, driver head motion, and vehicle CAN bus data are fused and fed into the SBL model. They found that the driver head pose signals lead to an early recognition of the lane change intent. Following with this system, Doshi and Trivedi evaluated the impact of eye gaze and head pose on the recognition of driver lane change intent [34]. The eye gaze locations in the front were divided into nine different areas. With various combination of the input data, they found that the eye gaze signals were not as informative as the head motion and did not significantly contribute to a precise intention inference. In [45], a discriminative RVM classifier was used to predict driver lane change intent with sliding-time window. The multimodal signals from ACC, SWA, LDW, and head motion were fused. It was found that the LDW system is more useful to predict the intent between 0 and 1.5 seconds prior to the lane change while the head motion is more informative between 2 and 3 seconds prior to the lane change. The surrounding vehicle position given by the SWA system do not affect the precise of intention inference. At the moment of 2 seconds prior to the lane change manoeuvre, the RVM based intention inference system achieved 82% prediction accuracy.

Jain *et al* [46] [47] designed several intelligent algorithms to anticipate the future manoeuvres for the driver. The Autoregressive Input-Output HMM (AIOHMM) and RNN-LSTM algorithms were used to predict the intention and future manoeuvres, respectively. The algorithms take the inside and outside video streams, vehicle dynamics, GPS, and street map as input signals to anticipate the lane change, turn, and normal driving manoeuvres. By comparing the predict results with different machine learning algorithms, they finally conclude that the RNN model leads to the best precision for 90.5% and can anticipate the manoeuvres 3.5 seconds before they occur. However, the strategical plan about the route and the utilization of digital map reduced the system reliability at unseen street.

7.1.3 Contribution

Although some successful researches about lane change intention inference have been reported in the recent years, far less works have been devoted to the lane change intention analysis in both cognitive level and the action level. The contribution of this study can be summarized as follows. Firstly, a general analysis for the inference of driver intention and

its frameworks are clarified. The framework is designed according to the driver cognitive process and the intention time scale. The hierarchical intention inference scheme would benefit the design of drive assistance system. Secondly, the proposed LCII system is designed with low-cost vision system, which is scalable and easy to be implemented. By integrating the rich inside and outside driving context into the deep RNN model, the temporal LCII model achieved the state-of-art recognition accuracy for the lane change intention compared with other non-temporal inference models. Finally, the naturalistic data collected on highways reflects the real-world driver response to the traffic context and the preparation process according to the lane change intention. The temporal properties of the lane change intention are statistically analysed.

7.2 Framework of Driver Intention Recognition

In this section, a driver lane change intention framework is introduced. The framework describes the general procedure of lane change intention and the relationship between each level. This framework also can be extended to describe any other intention inference problem.

7.2.1 Intention Inference Framework

Michon pointed out that the cognitive structure of human behaviour in traffic environment is a four levels hierarchical architecture, which contains the road user, transportation consumer, social agent, and psycho-biological organism [48]. Among these levels, the road user level is directly connected with drivers and can be further divided into three levels: strategy, tactical, and operational level (also known as control level) according to the time constant property. As shown in Figure 2-1, the strategical level defines the high-level plans for each trip such as the route, destination, and risk assessment, etc. The time constant of this level is much longer than that in the rest two levels, which can last for minutes or hours. The Tactical level tasks are the driving manoeuvres that a driver can take during the normal driving, such as lane change, lane keeping, brake, and acceleration. The time constant of these manoeuvres takes several seconds. Finally, the control level describes the human control actions on the vehicles and stand for the willing of the driver to remain safe and comfortable in the traffic situation. The time constant at this level is normally in milliseconds [26]. As shown in Figure 2-1, each tactical intention and manoeuvre can consist of a series combination of different

control actions. Unlike the strategical level intention that have explicit original and destination, the tactical intention is more random. Driver has to choose different manoeuvres according to the real-time traffic context. Therefore, the study of the tactical level intention is more challenge than the higher-level intention.

A general lane change intention framework is shown in Figure 2-4. Figure 2-4 indicates the relationship between the tactical level and the operational level. As discussed in [49], a human intention is generated by some stimuli. For example, the reason to make a lane change normally can be summarized as following a traffic rule or facing an uncomfortable driving context. The traffic context is the key stimuli for most of the lane change intention. Once an intention occurs, it can be reflected by different driver dynamics, such as the variation of EEG waves, the head and eye motion for mirror checking, etc. These behavioural signals play a key role in the estimation of the intention. After the drivers determine to execute the intention, they will control the vehicle through the steering wheel and the acceleration/brake pedal. Finally, the vehicle response to these control actions with the variation of vehicle dynamics.

After the analysis of the intention framework, it can be found that before the lane change manoeuvre is initiated, the driver has to execute a series of behaviours. The traffic context and the driver behaviours are the two important clues for predicting the lane change intention. The drivers will control the vehicle only after they have decided to finish their lane change intention. Therefore, the vehicle dynamic signals will have limited contribution to the prediction of the intention. However, these signals are still useful for the recognition of the intention before the vehicle cross the lane.

7.2.2 Problem Formulation

Based on the intention inference framework in Figure 2-4, this study will try to solve the following problems. Firstly, to construct an efficient DLII system based on the fusion of multi-modal sensors. The primary object of this study is to infer the lane change intention before the driver initiate the manoeuvre. Therefore, the steering wheel and pedal signals will not be used. Secondly, since the driver intention is not an instant detection task, the inference model should be able to process the temporal information before the lane change getting start and mining the temporal dependency between the features. Therefore, the LSTM based RNN model is adopted in this study. Next, it is important to understand the naturalistic driver behaviours and analysis the statistic roles within the

lane change manoeuvre. The statistic results for the mirror checking moment prior to the lane change and the lane change duration are analysed. Finally, one assumption made in this study is that only the intended lane change manoeuvres are studied. The unintended and aborted lane changes are not considered.

7.3 Methodologies

This section describes the experiment setup and data processing for lane change intention inference. In Section III. A, the experiment design and data collection are introduced. Section III. B and Section III. C describes the feature extraction for the traffic context and driver behaviours information, respectively.

7.3.1 Experiment Setup

In this study, the experimental data are collected from naturalistic driving on highway. The vehicle testbed is a commercial sport utility vehicle (SUV), which is equipped with multiple sensors. The sensory platform includes three low-cost CMOS cameras and a VBOX. The three cameras are all mounted inside the vehicle cabin. One is mounted in front of the driver to capture driver head motion and eye gaze information. The one that facing the front outside traffic context is mounted on the top middle of the front window. The rest one is mounted on the sunroof to monitor driver hand motion and only used for reference. A general system architecture is shown in Figure 7-1.

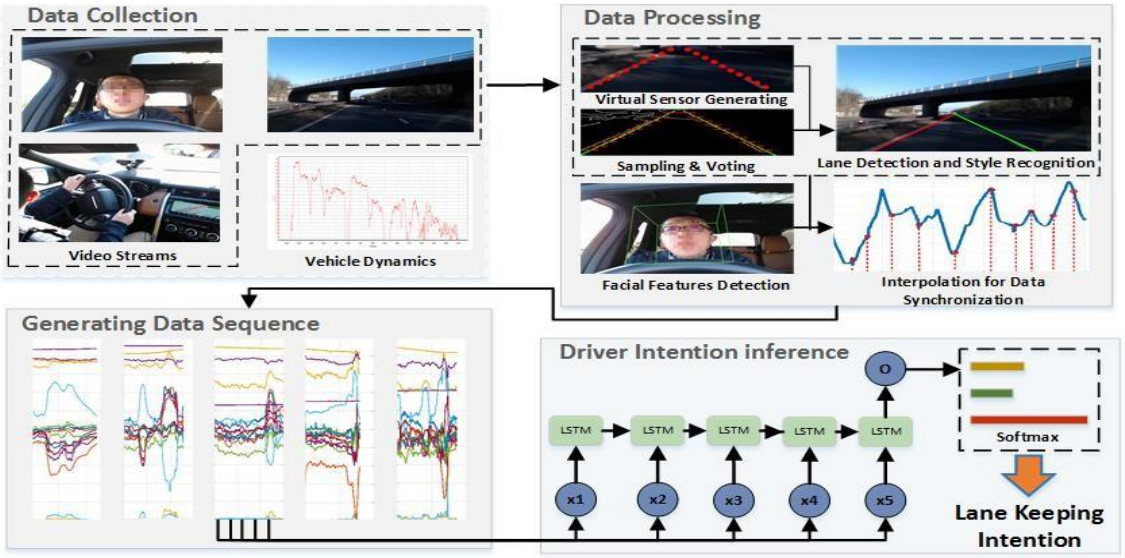

Figure 7-1. System architecture of lane change intention inference system.

All the three cameras are synchronized with timestamp and operate at 25 fps. The video streams are recoded at 640×480 resolution. Two VBOX antennas are fixed on the outside of the roof ceiling to measure the vehicle dynamics such as the velocity, acceleration, and heading, etc. The sampling frequency for the VBOX data logger are 20 HZ. All the signals are collected using a laptop with an Intel Core i7 2.5GHz CPU. Three adult drivers with varying ages and experiences are participated in the data collection process. They were asked to drive as usual without telling the real objective of the experiment. Each driver drove the vehicle on highway for about one hour and a total of 150 miles naturalistic data are collected.

7.3.2 Traffic Context and Vehicle Dynamic Features

The road facing camera captures the front traffic context, focus on the lane detection. The lane detection system is constructed with an edge detector and a Hough Transform. This method has been successfully used for lane detection in many studies [6]. Then, the detected lanes are tracked using a Kalman filter. After the lane positions are detected, a line style detector based on the lane sampling and voting (LSV) scheme is proposed. Three lane styles known as the solid, double solid, and dashed are recognized. Firstly, an odd number of sampling points are generated on the detected lane position. Then, the line style for the left and right lanes can be recognized by extending the sampling points to short sampling segments. The scanning is proposed on the edge image given by the Sobel edge detector and the lane style is determined by counting the rising edge along each sampling line. Conclusion are made by the voting results of the sampling lines. A detailed description can be found in Chapter IV.

Two vehicular signals known as the vehicle speed and heading angle are collected using the VOBX. Then, total feature vector for the outside traffic context and vehicular dynamics time t can be formed as a four-dimensional vector.

$$O_t = [L_r \; L_l \; V \; H] \tag{7-1}$$

where L_r and L_l are the lane style for right and left lane and $L \in [-1, 0, 1]$ represent the three different lane style.

7.3.3 Driver Behavioural Features

The inside features for driver head motion and eye gaze are detected using an open source system, namely, Openface [50]. The driver head position and facial landmarks are

detected using the Conditional Local Neural Fields (CLNF) approach [51]. The CLNF estimate the 3D head pose angles by projecting the 3D representation of the facial marks to the image plane using orthographic camera projection. The pupil locations for both eyes are detected with the deformable shape registration approach. The gaze directions are estimated according to the pupil locations and the 3D eyeball centre [52]. The inside feature vector at each time constant can be formed as follows.

$$I_t = [G_r\ G_l\ G_a\ H_t\ H_r] \tag{7-2}$$

where G_a, is the 2D gaze angle in x and y coordinate, G_r, G_l, H_t, and H_r are the 3D gaze direction for each eye, head pose translation vector and head pose direction vector, respectively. This gives to a 14-dimensional vector inside feature vector. A comparison of the head yaw angles rotation is illustrated in Figure 7-2.

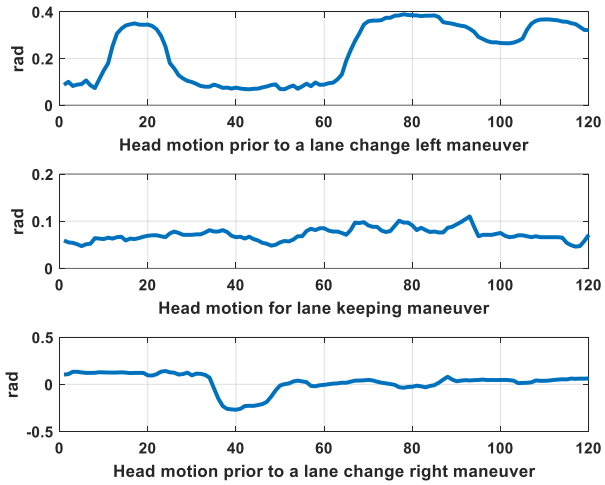

Figure 7-2. Head yaw angle illustration for different manoeuvres.

7.4 Algorithms

In this section, the Recurrent Neural Network basics and the Long Short-Term Memory cell are introduced. The RNN model is used to learn the temporal dependency between the input data, and the LSTM efficiently increase the performance of RNN model by capturing the long-term context dependencies.

7.4.1 Recurrent Neural Network

Since driver intention inference is not an instance detection task, previous driver behavioural data need to be taken into consideration. Therefore, the recurrent neural network is applied in this study to process the sequential inputs. RNN allows to exhibit the dynamic temporal behaviour of a sequence by forming a directed connection between

previous states and current state [53]-[55]. Figure 7-3. illustrates a basic structure of the RNN model, the right-side model in the graph is the un-fold version of the left circuit diagram. Current state s_t can be viewed as the memory of RNN network, it stores the information that happened in all the previous time step.

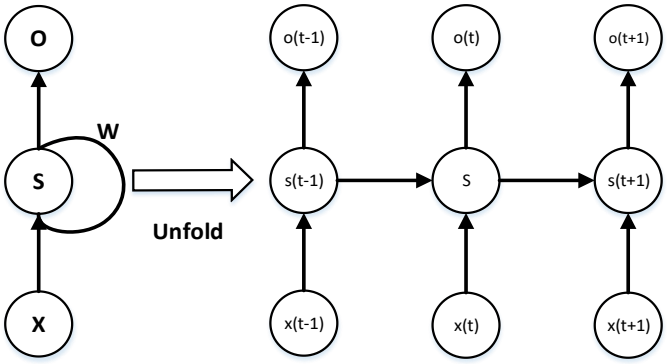

Figure 7-3. A simplified recurrent neural network architecture.

The RNN model can be described as below.

$$s_t = f(\boldsymbol{W}_x x_t + \boldsymbol{H}_s s_{t-1} + b_x) \tag{7-3}$$

$$o_t = softmax(\boldsymbol{W}_o s_t + b_o) \tag{7-4}$$

where f is the activation function of the hidden states, which normally selected as $tanh$ and $sigmoid$ function. X, S, and O represent the input, hidden states, and the output of the RNN, respectively. $\boldsymbol{W}_x$, $\boldsymbol{H}_s$, and $\boldsymbol{W}_o$ are the model parameter matrix for the input, hidden states, and output. b_x and b_o are the bias vector.

To reduce the total number of parameters, the RNN share the same $\boldsymbol{W}_x$, $\boldsymbol{H}_s$, and $\boldsymbol{W}_o$ at each time step. This means that the RNN executes the same task with different input values at each step. Although the number of training parameters are reduced with the parameter sharing scheme, RNN still suffer another serious problem, namely, the gradient vanishing or exploding [56]. Therefore, the simplified RNN structure has limited memory ability to remember the long-term dependency.

7.4.2 Long Short-Term Memory

Hochreiter and Schmidhuber developed the LSTM cell to overcome the drawbacks of RNN [57]. The LSTM-RNN solves the long-term dependency problem by introducing three extra gates, known as the input gate, forget gate, and output gate. The central idea behind LSTM is that the gates in the LSTM cell cooperate with each other to control how much information should be remained and forgot. Figure 7-4. shows the LSTM

architecture. The LSTM RNN still follows the chain-like structure as shown in Figure 7-3[58]. The difference is LSTM RNN replace the hidden unit (normally a $sigmoid$ or $tanh$ activation function) with a LSTM cell.

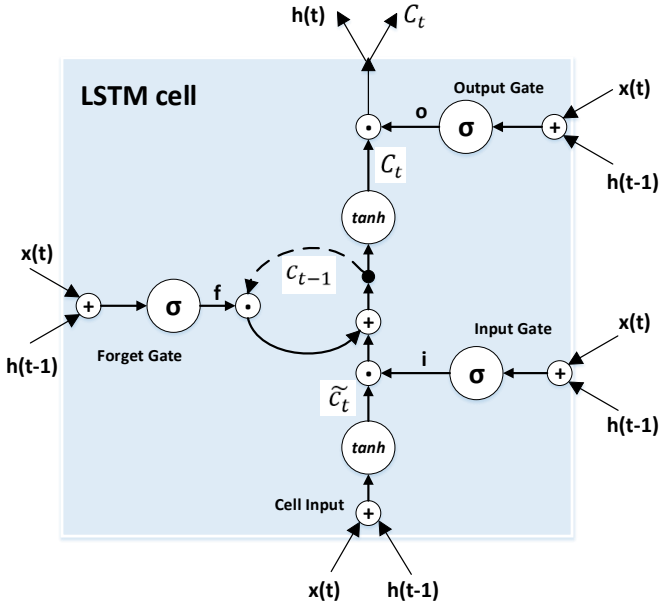

Figure 7-4. Illustration of LSTM cell structure

The LSTM cell has the following mathematic representation. Firstly, the forget gate controls what information to throw away. Then, the input gate chooses what new information to be updated and stored. The output gate controls the candidate layer output.

$$f_t = \sigma\left(\boldsymbol{U}_f x_t + \boldsymbol{W}_f s_{t-1} + b_f\right) \tag{7-5}$$

$$i_t = \sigma(\boldsymbol{U}_i x_t + \boldsymbol{W}_i s_{t-1} + b_i) \tag{7-6}$$

$$o_t = \sigma(\boldsymbol{U}_o x_t + \boldsymbol{W}_o s_{t-1} + b_o) \tag{7-7}$$

The value $\tilde{c}_t$ is the candidate cell state which can be represented as

$$\tilde{c}_t = tanh(\boldsymbol{U}_c x_t + \boldsymbol{W}_c s_{t-1} + b_c) \tag{7-8}$$

The c_t in the central is the internal memory cell state of the LSTM unit, which is the combination of previous c_{t-1} and current candidate states.

$$C_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{7-9}$$

Finally, the layer output is the products of the cell state C_t and the candidate output from the output gate.

$$s_t = o_t * tanh(C_t) \tag{7-10}$$

where σ in the above equations represents the $sigmoid$ function. $*$ is the element-wise production. x_t and s_{t-1}, are current input vector and previous layer output. f, i, o are the forget gate, input gate, and output gate. U, W, b are corresponding model parameters.

The input x_t for the LSTM-RNN at each time step is the concatenation of the inside and outside feature vector $x_t = [I_t, O_t]$, which is an 18-dimensional vector in this study. Hence, each training sample for the LSTM-RNN is an 18-D temporal sequence and the total dataset can be formed as$\{(x_{t-n}, x_{t-n+1}, \cdots, x_t)_j, y_j\}_{j=1}^{N}$, where N is the total number of the training sample. y_j is the intention label for this sequence. In this study, the target values are manually labelled according to the video stream, which has three candidate values as$\{lane\ change\ left, lane\ change\ right,\ lane\ keeping\}$. There are 135 lane change manoeuvres are detected according to the experiment video with 65 lane change left cases, and 70 lane change right cases. Then, 66 normal driving sequences are randomly picked. The LSTM-RNN model are trained with MATLAB Deep Learning Toolbox.

7.5 Experiment

In this section, statistical analysis is proposed based on the critical moment of lane change manoeuvre. Next, lane change intention inference results using different algorithms are described.

7.5.1 Driver Lane Change Manoeuvre Analysis

As shown in Figure 7-5, there are four critical moments for one lane change manoeuvre can be identified, which are the intention occur point (denoted as T1), manoeuvre start point (T2), lane crossing point (T3), and manoeuvre finishing point (T4), respectively. At T1, the driver generates a lane change intention according to the traffic context stimuli. Most of the time, the specific moment when the driver arises an intention is undetectable. Therefore, T1 is replaced by the first mirror checking moment in the following analysis. At T2, driver uses the turn signal to indicate their lane change intention and then turn the steering wheel. Finally, T3 and T4 are moment that the vehicle just cross the lane and finish the lane change manoeuvre, respectively.

In this study, the statistical analysis of the gap between T1 and T2, T2 and T3 for all the lane change manoeuvres are proposed. The time interval between T1 and T2 (denoted as T1-T2) measures time cost for the lane change preparation, and the interval between

T2 and T3 (denoted as T2-T3) measures how long it takes to cross the lane after the driver starts the manoeuvre.

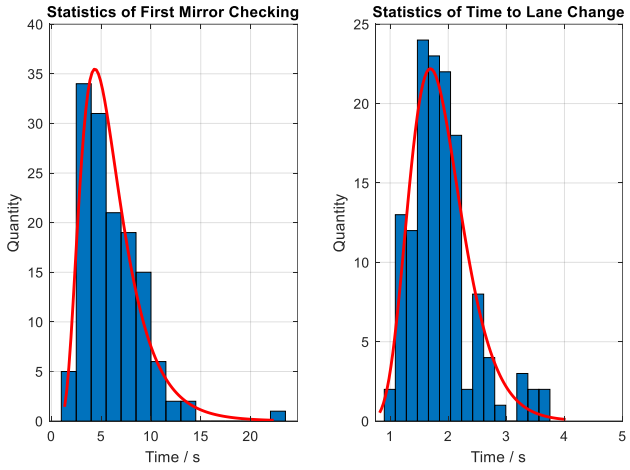

Figure 7-5. Statistic results for critical time intervals of lane change manoeuvre

TABLE 7-1 DESCRIPTIVE STATISTICS OF THE LANE CHANGE PREPARATION AND EXECUTION

Statistics	T1-T2	T2-T3
Mean	6.085 s	1.881 s
SD	3.033 s	0.530 s
Variance	9.200s^2	0.281s^2
Median	5.437 s	1.805 s
Mode	3.060 s	1.890 s
Maximum	22.423 s	3.680 s
Minimum	1.843 s	0.930 s

*SD is short for standard deviation

In Figure 7-6, the time distributions of T1-T2 and T2-T3 are fitted with lognormal distribution function. According to the statistical results shown in Table 7-1, the drivers in the dataset would perform their first mirror checking behaviour about six seconds before they start the lane change manoeuvre. The average time cost between initiate the lane change and just crossing the lane is about 1.88 seconds. Therefore, to predict the lane change intention before driver turn on the signal or the steering wheel, a 6 seconds window is sufficient to cover the important driver behaviour features. After driver starts the lane change manoeuvre, the detector has nearly 2 seconds to recognize an intended lane change. From the dataset it can be found that mirror checking behaviours can occur during the lane keeping manoeuvre, however, the duration for this mirror checking behaviours are much shorter than that during the lane change preparation process. Meanwhile, drivers tend to perform single side mirror checking behaviours multiple times during the lane change preparation step instead of performing both sides checking behaviours as they do during the normal lane keeping process.

7.5.2 Lane Change Intention Inference

In this study, we examine the performance of the intention inference model from two aspects, which are the detection accuracy and the prediction horizon. The prediction horizon measures the different inference results along with the time gap prior to the manoeuvre. In [46], the intention prediction results given by various machine learning models have been richly studied. The authors found that the performance of the RNN model overweighs the other models such as the IOHMM, HMM, and SVM, Therefore, in the following parts, only a FFNN model that trained in a different manner compared to the method in [46] is used for comparison purpose. To predict the lane change intention before the manoeuvre, the temporal sequence which is up to 6 seconds prior to the manoeuvre is selected. Since all the drivers are asked to drive as usual, some lane change manoeuvres are not indicated by the turn signal. Therefore, the start point of the manoeuvre is marked as either turn on the signal or the first moment of turning the steering wheel. To train the LSTM-RNN model, 70% of the data are randomly selected for model training and the rest data are used for testing. Figure 7-6 illustrates the confusion matrix for the intention inference results using LSTM-RNN with complete features.

As shown in Figure 7-6, the general inference accuracy for the three manoeuvres are 96.7%. The left lane change intentions are all correctly predicted before the driver initiate the lane change manoeuvre. For the right lane change and straight driving intention in the testing dataset, each one has one misclassified sample. Unlike [47], which train the non-temporal discriminative model by concatenating the feature vector at each step into a large feature set, in this study, the FFNN is trained with a much smaller feature vector. For each training sample, the 18-D temporal sequence data is transformed to a 72×1 feature vector. The mean, standard deviation, maximum value, minimum value for the 18 channels are computed. The 72-dimensional feature vector is much smaller compared with the 3840-dimensional vector in [47]. The intention prediction results using the FFNN is shown in Figure 7-7. The general performance of the FFNN model is 91.8%. Although the detection rate using the discriminative model is lower than the LSTM-RNN model, the proposed FFNN still shows an acceptable ability on the intention prediction task and achieves a better result compared with the other discriminative models in [46]. In addition, as shown in Figure 7-7, the inference accuracy for the left lane change intention is still

100%, which means for these two different models, the left lane change intention is the easiest to be detected. One of the most important reason is that when prepare the left lane change manoeuvre (right side driving vehicle), the driver has to check the far-end lest-side mirror or the rear-mirror by rotating their head and check multiple times instead of only gaze searching as they did in many right lane change manoeuvre, which makes the system easier to detect the corresponding behaviours and the inner intention.

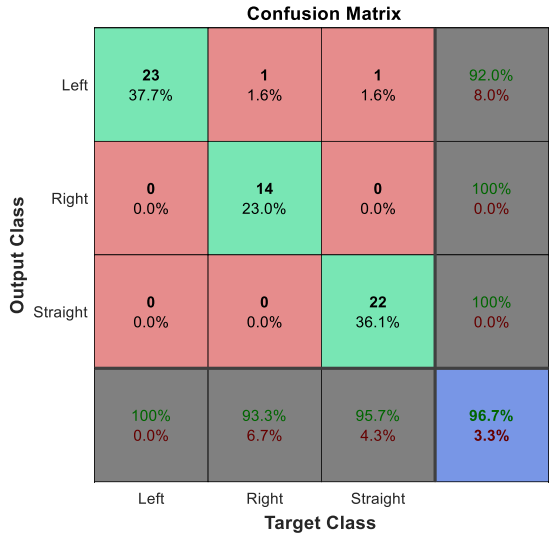

Figure 7-6. Confusion matrix for lane change intention inference using LSTM-RNN.

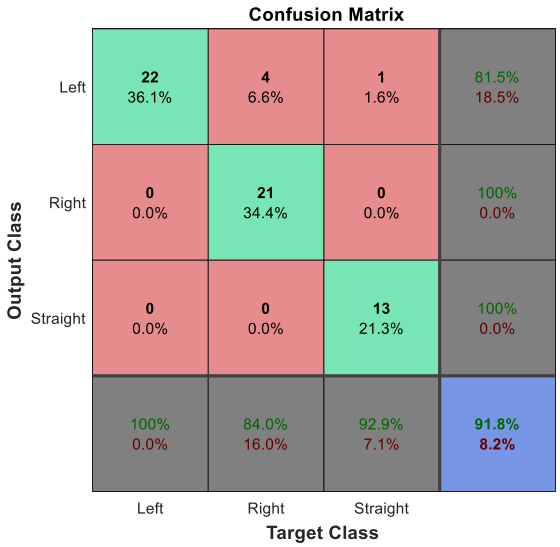

Figure 7-7. Confusion matrix for lane change intention inference using FFNN.

The model prediction performance for the RNN model and FFNN model are evaluated using the sliding window method. Specifically, the testing sequence are shifted back for every 0.5 second, which means the intention is predicted every 0.5 second prior to the manoeuvre. As shown in Figure 7-8 and Figure 7-9, the LSTM-RNN model is more

accurate than the FFNN model when inferring the intention 1 second early. While, the FFNN model gives a more robust detection when the intention is inferred more than 1 second.

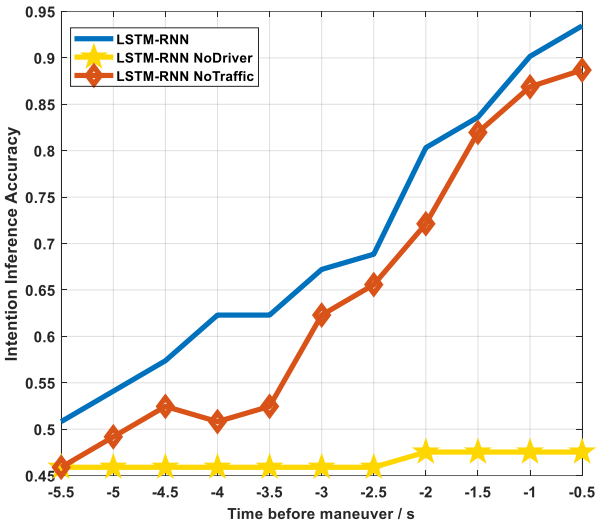

Figure 7-8. Prediction performance vs. time-to-manoeuvre for lane change with LSTM-RNN model.

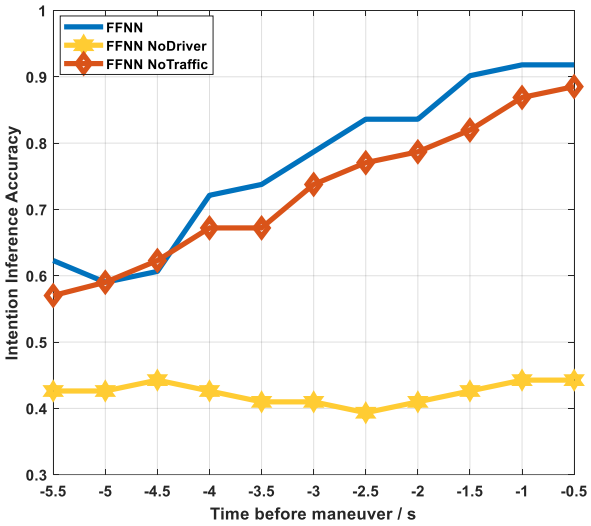

Figure 7-9. Prediction performance vs. time-to-manoeuvre for lane change with FFNN model.

This may dues to the fact that as the testing sequence move earlier, more irrelevant information is involved in the temporal sequence, which would confuse the RNN model on the intention inference tasks. On the other hand, the FFNN model is trained with the statistic features of the sequence. Therefore, the FFNN may still able to capture the significant features within the temporal sequence. The impacts of the driver behavioural

data as well the traffic context data on the intention inference are also evaluated. Based on the results from the two different methods, the traffic context features do not contribute to an accurate detection of the driver intention before the manoeuvre. On the contrary, the model based on the driver behavioural features achieved slightly lower accuracy than the model with full feature vector, which means the driver intention can be roughly detected according to driver's intended checking behaviours.

Figure 7-10 illustrates the intention inference accuracy with respect to the prediction time. In this part, the models are trained with completed sequence data. While at the testing step, the testing data are cropped. The results indicate the intention prediction results by observing partial temporal context. As the testing sequence getting smaller, more important features along the lane change preparation duration will lost. The purple line and the yellow line in Figure 7-10 represent the results by using driver behaviour features only and using traffic context and the vehicle dynamic features only. As shown in Figure 7-10, the driver behavioural features are more important than the traffic context features. The red line indicates the prediction results given by the FFNN model. Unlike the temporal sequence-based methods, the FFNN accuracy drops more rapidly as the testing sample getting smaller. This means the FFNN model is more sensitive to the partially observed dataset.

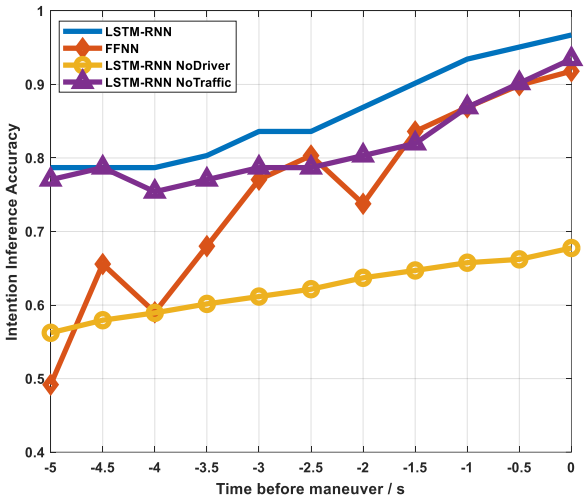

Figure 7-10. Prediction performance vs. time-to-manoeuvre for lane change with incomplete sequence.

7.6 Discussions and Future Works

In this study, the driver intention inference (DII) system is proposed based on the naturalistic driving data on highway. It has been found that it is possible to infer human driver intention before they initiate the lane change manoeuvre. However, as far as we concern, a few works are expected before the driver intention inference can be implemented into real-world system.

Firstly, the intention inference system in this study assumes the human drivers always concentrate on the driving tasks during the experiment. The experiment duration for each participant is about one hour, which can make sure the drivers are not overloaded. However, in real-world, the DII system should not be working as an isolated function. The system needs to cooperate with other driver assistance system to make an accurate prediction result. The most relevant system is the driver workload estimation module. It is found that when the drivers are overloaded, their behaviours change significantly, and the driver workload is highly related with the intention inference system [59]. Therefore, an integrated system which combine the driver workload estimation and intention inference system is needed for real-time driver assistance.

Secondly, as the vehicle intelligence increasing significantly nowadays, more and more driver assistance system will share the control authority with the human driver. What happens if the drivers are performing the secondary tasks and how to correctly infer their intention? The DII system has to clarify the driver status before estimating the true intention. By considering driver distraction and intention has a whole, the control conflicts between driver and the vehicle can be minimized. Meanwhile, for those partial automated vehicles, the estimation of secondary tasks will help the decision system to determine the driver's intention and whether they are capable to take over the control on some emergency tasks [60][61].

Finally, a more challenge work is to have a more comprehensive understanding about the cognitive intention generation process according to the traffic context and human behaviours. By analysing the driver intention with respect to the current traffic context, a human-like intention generation system can be proposed. These knowledges will benefit the design of the decision-making system of the automated vehicles.

7.7 Conclusions

In this study, a driver lane change intention inference is proposed based on the LSTM-RNN model. The experiment data are collected on highway environment in real-world. A general framework for intention inference framework is designed. According to the framework, the outside traffic context is viewed as the stimuli for driver intention, while the driver behaviours and the vehicle dynamics are the corresponding responses to the intention. Based on the statistics of the experiment data, it is found that driver tends to perform the first mirror checking behaviour six seconds prior to the lane change manoeuvre and it takes around two seconds for the vehicle to cross the lane. A LSTM-RNN and FFNN model are used for intention inference. The results indicate that the RNN model achieved an average of 96.7% inference accuracy before the manoeuvre being initiated. The LSTM-RNN model is proved more accurate and robust with partially observed data than the FFNN model. Future works will concentrate on the comprehensive analysis of the driver intention and its integration with other driver status detection system.

7.8 References

[1] Yang, Liuqing, and Fei-Yue Wang. "Driving into intelligent spaces with pervasive communications." *IEEE Intelligent Systems* 22.1 (2007).

[2] Bellis, Elizabeth, and Jim Page. National motor vehicle crash causation survey (NMVCCS) SAS analytical users manual. No. HS-811 053. 2008.

[3] Martinez, Clara Marina, *et al*. "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey." *IEEE Transactions on Intelligent Transportation Systems* (2017).

[4] Lv, Chen, *et al*. "Characterization of Driver Neuromuscular Dynamics for Human-Automation Collaboration Design of Automated Vehicles." *IEEE/ASME Transactions on Mechatronics* (2018).

[5] Ohn-Bar, Eshed, and Mohan Manubhai Trivedi. "Looking at humans in the age of self-driving and highly automated vehicles." *IEEE Transactions on Intelligent Vehicles* 1.1 (2016): 90-104.

[6] Gaikwad, Vijay, and Shashikant Lokhande. "Lane departure identification for advanced driver assistance." *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015): 910-918.

[7] Saito, Yuichi, Makoto Itoh, and Toshiyuki Inagaki. "Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents." *IEEE Transactions on Human-Machine Systems* 46.5 (2016): 660-671.

[8] Milanés, Vicente, *et al*. "Cooperative adaptive cruise control in real traffic situations." *IEEE Transactions on Intelligent Transportation Systems* 15.1 (2014): 296-305.

[9] Mukhtar, Amir, Likun Xia, and Tong Boon Tang. "Vehicle detection techniques for collision avoidance systems: A review." *IEEE Transactions on Intelligent Transportation Systems* 16.5 (2015): 2318-2338.

[10] Tawari, Ashish, *et al*. "Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking." *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014.

[11] Xing, Yang, *et al*. "Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition." *IEEE Transactions on Computational Social Systems* 5.1 (2018): 95-108.

[12] Berndt, Holger, Jorg Emmert, and Klaus Dietmayer. "Continuous driver intention recognition with hidden markov models." *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008.

[13] Kuefler, Alex, *et al*. "Imitating driver behaviourwith generative adversarial networks." *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017.

[14] Morton, Jeremy, and Mykel J. Kochenderfer. "Simultaneous Policy Learning and Latent State Inference for Imitating Driver Behavior." *arXiv preprint arXiv:1704.05566* (2017).

[15] Li, Li, *et al*. "Intelligence testing for autonomous vehicles: a new approach." *IEEE Transactions on Intelligent Vehicles* 1.2 (2016): 158-166.

[16] Liu, Andrew, and Alex Pentland. "Towards real-time recognition of driver intentions." *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*. IEEE, 1997.

[17] Oliver, Nuria, and Alex P. Pentland. "Driver behaviourrecognition and prediction in a SmartCar." *PROC SPIE INT SOC OPT ENG*. Vol. 4023. 2000.

[18] Liebner, Martin, and Felix Klanner. "Driver intent inference and risk assessment." Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort (2014): 1-20.

[19] Gwon, Gi-Poong, *et al*. "Generation of a precise and efficient lane-level road map for intelligent vehicle systems." *IEEE Transactions on Vehicular Technology* 66.6 (2017): 4517-4533.
[20] Balid, Walid, Hasan Tafish, and Hazem H. Refai. "Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance." *IEEE Transactions on Intelligent Transportation Systems* (2017).
[21] Wang, Fei-Yue. "Artificial intelligence and intelligent transportation: driving into the 3rd axial age with its." *IEEE Intelligent Transportation Systems Magazine* 9.4 (2017): 6-9.
[22] Gou, Chao, *et al*. "A joint cascaded framework for simultaneous eye detection and eye state estimation." *Pattern Recognition* 67 (2017): 23-31.
[23] Murphy-Chutorian, Erik, and Mohan Manubhai Trivedi. "Head pose estimation in computer vision: A survey." *IEEE transactions on pattern analysis and machine intelligence* 31.4 (2009): 607-626.
[24] Chen, Jixu, and Qiang Ji. "A probabilistic approach to online eye gaze tracking without explicit personal calibration." *IEEE Transactions on Image Processing* 24.3 (2015): 1076-1086.
[25] Haufe, Stefan, *et al*. "EEG potentials predict upcoming emergency brakings during simulated driving." *Journal of neural engineering* 8.5 (2011): 056001.
[26] Doshi, Anup, and Mohan M. Trivedi. "Tactical driver behaviourprediction and intent inference: A review." *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011.
[27] Salvucci, Dario D., *et al*. "Lane-change detection using a computational driver model." *Human factors* 49.3 (2007): 532-542.
[28] Toledo-Moreo, Rafael, and Miguel A. Zamora-Izquierdo. "IMM-based lane-change prediction in highways with low-cost GPS/INS." *IEEE Transactions on Intelligent Transportation Systems* 10.1 (2009): 180-185.
[29] Salvucci, Dario D., and Andrew Liu. "The time course of a lane change: Driver control and eye-movement behavior." *Transportation research part F: traffic psychology and behaviour* 5.2 (2002): 123-132.
[30] Schmidt, Kim, *et al*. "A mathematical model for predicting lane changes using the steering wheel angle." *Journal of safety research* 49 (2014): 85-e1.
[31] Driggs-Campbell, Katherine, and Ruzena Bajcsy. "Identifying modes of intent from driver behaviors in dynamic environments." *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015.
[32] Henning, Matthias J., *et al*. "Modelling driver behaviour in order to infer the intention to change lanes." *Proceedings of European Conference on Human Centred Design for Intelligent Transport Systems*. Vol. 113. 2008.
[33] Beauchemin, Steven S., *et al*. "Portable and scalable vision-based vehicular instrumentation for the analysis of driver intentionality." *IEEE Transactions on Instrumentation and Measurement* 61.2 (2012): 391-401.
[34] Doshi, Anup, and Mohan Manubhai Trivedi. "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes." *IEEE Transactions on Intelligent Transportation Systems* 10.3 (2009): 453-462.
[35] Lethaus, Firas, *et al*. "A comparison of selected simple supervised learning algorithms to predict driver intent based on gaze data." *Neurocomputing* 121 (2013): 108-130.
[36] Murphy-Chutorian, Erik, and Mohan Manubhai Trivedi. "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness." *IEEE Transactions on intelligent transportation systems* 11.2 (2010): 300-311.
[37] Jang, Young-Min, Rammohan Mallipeddi, and Minho Lee. "Identification of human implicit visual search intention based on eye movement and pupillary analysis." *User Modeling and User-Adapted Interaction* 24.4 (2014): 315-344.
[38] Pech, Timo, Philipp Lindner, and Gerd Wanielik. "Head tracking based glance area estimation for driver behaviour modelling during lane change execution." *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014.
[39] Zhou, Huiping, Makoto Itoh, and Toshiyuki Inagaki. "Influence of cognitively distracting activity on driver's eye movement during preparation of changing lanes." *SICE Annual Conference, 2008*. IEEE, 2008.
[40] Bi, Luzheng, *et al*. "Detecting driver normal and emergency lane-changing intentions with queuing network-based driver models." *International Journal of Human-Computer Interaction* 31.2 (2015): 139-145.
[41] Ding, Jieyun, *et al*. "Driver intention recognition method based on comprehensive lane-change environment assessment." *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014.
[42] Li, Keqiang, *et al*. "Lane changing intention recognition based on speech recognition models." *Transportation research part C: emerging technologies* 69 (2016): 497-514.
[43] Kasper, Dietmar, *et al*. "Object-oriented Bayesian networks for detection of lane change manoeuvres." *IEEE Intelligent Transportation Systems Magazine* 4.3 (2012): 19-31.
[44] McCall, Joel C., *et al*. "Lane change intent analysis using robust operators and sparse bayesian learning." *IEEE Transactions on Intelligent Transportation Systems* 8.3 (2007): 431-440.
[45] Doshi, Anup, Brendan Morris, and Mohan Trivedi. "On-road prediction of driver's intent with multimodal sensory cues." *IEEE Pervasive Computing* 10.3 (2011): 22-34.
[46] Jain, Ashesh, *et al*. "Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture." *arXiv preprint arXiv:1601.00740* (2016).
[47] Jain, Ashesh, *et al*. "Car that knows before you do: Anticipating manoeuvres via learning temporal driving models." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
[48] Michon, John A. "A critical view of driver behaviourmodels: what do we know, what should we do?." *Human behaviourand traffic safety*. Springer, Boston, MA, 1985. 485-524.
[49] Youn, So-Jeong, and Kyung-Whan Oh. "Intention recognition using a graph representation." *World Academy of Science, Engineering and Technology* 25 (2007): 13-18.
[50] Baltrušaitis, Tadas, Peter Robinson, and Louis-Philippe Morency. "Openface: an open source facial behaviouranalysis toolkit." *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016.
[51] Baltrusaitis, Tadas, Peter Robinson, and Louis-Philippe Morency. "Constrained local neural fields for robust facial landmark detection in the wild." *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013.
[52] Wood, Erroll, *et al*. "Rendering of eyes for eye-shape registration and gaze estimation." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
[53] Zyner, Alex, Stewart Worrall, and Eduardo Nebot. "A Recurrent Neural Network Solution for Predicting Driver Intention at Unsignalized Intersections." *IEEE Robotics and Automation Letters* (2018).
[54] Morton, Jeremy, Tim A. Wheeler, and Mykel J. Kochenderfer. "Analysis of recurrent neural networks for probabilistic modeling of driver behavior." *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2017): 1289-1298.

[55] Olabiyi, Oluwatobi, *et al*. "Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network." *arXiv preprint arXiv:1706.02257* (2017).
[56] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." *IEEE transactions on neural networks* 5.2 (1994): 157-166.
[57] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
[58] Greff, Klaus, *et al*. "LSTM: A search space odyssey." *IEEE transactions on neural networks and learning systems* 28.10 (2017): 2222-2232.
[59] Xing, Yang, *et al*. "Driver workload estimation using a novel hybrid method of error reduction ratio causality and support vector machine." *Measurement* 114 (2018): 390-397.
[60] Chen, Huei-Yen Winnie, *et al*. "Self-reported engagement in driver distraction: An application of the Theory of Planned Behaviour." *Transportation research part F: traffic psychology and behaviour* 38 (2016): 151-163.
[61] Eriksson, Alexander, and Neville A. Stanton. "Takeover time in highly automated vehicles: noncritical transitions to and from manual control." *Human factors* 59.4 (2017): 689-705.

PART VI:

CONCLUSION AND FINAL REMARKS

8

CONCLUSIONS, DISCUSSIONS AND DIRECTIONS FOR FUTURE WORK

In the course of this thesis, the main contribution can be summarized as: integrated lane detection algorithm and evaluation system that robust to various situations, the recognition of driver actions and secondary tasks towards a comprehensive driver understanding, driver lane change manoeuvre and intention analysis based on the time-scale, the proposed lane change intention system is based on naturalistic data and able to predict the lane change intention a few seconds earlier before the manoeuvre happens. Furthermore, an extension for the driver mental status study in the appendix is proposed through the estimation of driver workload based on multi-modal signals. These contributions are elaborated in the following so as to discuss the limitations as well as the future development and work directions.

8.1 Integrated Lane Detection towards Robust Traffic Context Perception

Lane detection is one of the most important traffic context awareness way to assist the driving tasks. Meanwhile, understanding ego-lane position and the style play a critical role in the driver intention estimation since a large amount of lane change manoeuvres are generated according to the ego-lane status. In this study, to overcome the robustness issue of the lane detection system, an integrated lane detection framework is proposed with comprehensive literature survey as well as the algorithm design and evaluation. The main contribution of this part can be summarised as follow. Firstly, a state-of-art literature review on lane detection, integration, and evaluation is proposed. In previous studies, there are no such comprehensive analysis for the integration and evaluation methods of the lane detection system. Secondly, by analysing the limitation of previous studies, an algorithm level integrated lane detection system is designed. The system is able to detect lane positions and evaluate the detection performance. The proposed integrated system satisfies the real-time application requirement and robust to various road challenge (lighting, shadows, occlusion, etc.). Finally, a general parallel lane detection system framework is designed. By mapping the real-world problem into the virtual parallel world, unlimited experimental data can be collected. The parallel lane detection system is a powerful way for the lane detection system construction and evaluation.

8.1.1 Algorithm Limitation

Currently, the proposed lane detection system is solely camera-based system. As analysed in this project, camera-based system has its inevitable limitations especially

when facing the atrocious weather condition. Therefore, a more robust system should rely on the sensory fusion technique. Meanwhile, the proposed lane detection algorithm relies on a pre-determined ROI, which can cause inaccurate detection results when transfer the system to other platform without manual calibration. Therefore, the road area detection and automatic ROI selection will be an efficient way to increase the system robustness and transportability.

8.1.2 Directions for Future Work

The proposed lane detection system is tested with both public dataset and local Cranfield dataset collected on highways. However, the testing is still not enough to prove the more general performance. The evaluation issue is a challenge task to all of the vision-based systems. Therefore, to evaluate the system and deign robust driver assistance system, the parallel driving framework is one of the future research direction. The parallel framework will provide more testing dataset to evaluate the performance of the lane detection system. Meanwhile, it enables the online learning and updating of the lane detection system, which makes the system more robust to unseen traffic context. Next, the integration of the lane detection system with other vision-based assistance system is other research direction. By efficiently integrates the lane detection with road detection or vehicle detection, the vision system will be more intelligent to deal with the complex traffic context.

8.2 Driving Activities recognition and Secondary Tasks Detection

In this part, the normal driving behaviours like mirror checking action and multiple secondary tasks are recognised using machine learning methods. The mirror checking behaviour is the most important clues for lane change intention inference. For the secondary tasks, it reflects the distraction level of the driver. It is critical important to identify the driver distraction from the normal driving cases since when the drivers are distracted, their behaviours will have limited contribution to the intention inference. The main contribution of this part includes the following aspects. Firstly, a driving activities and secondary tasks recognition system is proposed based on the feature evaluation and selection. At this stage, driver head and body joints positions are collected. Few researches in the past consider the impact of driver body joints to the distraction detection. The algorithm evaluates the head pose, body pose, driver arm position, and the depth

information separately. It is found that driver body pose features contribute to a more accurate driver status classification. Secondly, to overcome the complex feature extraction and selection process, an end-to-end deep learning-based classification model is proposed. An unsupervised learning method for driver detection is applied. The deep CNN model takes the colour image as the input directly and outputs the driver status. This method will benefit the design of low-cost driver behaviour detection system. Finally, the driver behaviour identification algorithms contribute to a more accuracy driver intention inference, which enables the inference system to recognise the drivers' intention only when they are concentrate on the driving task

8.2.1 Algorithm Limitation

Currently, the driver behaviour detection dataset contains only five drivers. The fiver drivers are asked to perform seven tasks, which contains three different secondary tasks. As we all know, driver can be distracted by different reasons, while the three distraction tasks only contain physical distraction and ignore the psychological distraction. Therefore, the algorithm cannot identify the true status of the driver if the driver is psychological distracted. Unfortunately, it is not able to collect more naturalist data in current stage. Secondly, the deep learning algorithms do not achieve a better result compared with the conventional machine learning methods. The reason for this can be multi-folds. Firstly, it may due to the amount of the dataset, which may need to be enriched. Secondly, the algorithm does not use a precise driver detection algorithm, which makes the segmented image still contain many outliers.

8.2.2 Directions for Future Work

Therefore, future works should concentrate on the collection of more naturalist data from more drivers. Meanwhile, more driver distraction status can be involved such as to estimation the cognitive distraction. This requires a comprehensive driver understanding, which need to take the eye gaze information into consideration. The audio information is another important feature, which is not used in this project. Sometimes the audio information carries more distinct features than the images, especially when the driver speaking with other passengers but looking straight ahead.

Another work is to study the driver behaviours on partially automated vehicles. As driver will have different behaviours when driving the automated vehicles, how the driver

behaviour reasoning system interact with the automated vehicles is a challenging task. Moreover, when using automated vehicles, driver will have more freedom to perform secondary tasks, which is impossible on the convention vehicles. Therefore, the collected dataset would be more valuable for the study of driver behaviours under different situation.

8.3 Driver Intention Inference based on Traffic Context and Driver Behaviour Recognition

Driver intention is different from the previous driver behaviour detection, which is an instance classification task. Driver intention inference requires to capture the long-term dependency of the temporal sequences. In the final part of this thesis, driver lane change intention is inferred based on the studies in previous sections. As one of the most notable contribution of this thesis, driver intention is firstly analysed according to the human intention studies. The generation procedure of lane change intention is analysed based on the time-scale property. The relationship between tactical intention and operation intention is clarified. It is found that traffic context plays the role of intention stimuli, and driver behaviours are the response to the intention. Accordingly, to predict the lane change intention before the manoeuvre happing, the traffic context and driver behaviours are the two most important features. The deep learning-based intention inference algorithm is able to learn the long-term dependence along the training data. Based on the statistical analysis of the lane change manoeuvres from the naturalistic driving dataset, the intention inference algorithm achieved precise prediction results before the manoeuvre is initiated.

8.3.1 Algorithm Limitation

Similar to the previous studies, the dataset in this part is expected to be enriched. The data quantity issue is the common challenge to all the machine learning tasks. To meet the industrial requirement, a mature drive assistance system which can be used in the real-world need hundreds of drivers to be involved. The system still need a lot of tests before publishing to the public. Moreover, the lane change cases are manually extracted from the videos, which is time consuming.

8.3.2 Directions for Future Work

Based on the above discussion, future works for lane change inference system design can be summarised as follows. Firstly, make the system robust to various drivers by

enriching the dataset volume. Next, the lane change cases are expected to be detected automatically based on the lane detection system. If the timestamp of the critical points for each lane change can be detected, the model training and testing can be more efficient. Finally, as discussed in previous sections, future driver intention inference system should not be working individually. The intention inference system need cooperate with other driver status detection system like driver workload estimation and driver distraction detection. It is believed that when the driver is overloaded or being distracted, their behaviours will be different compared with the normal condition. In the future, the relationship between driver intention and driver workload and the influence to each other is a challenge research point.

8.4 Conclusions and Final Discussions

During the development of this project it has been found that the intelligent driver assistance system is no longer an isolated function. It requires the fusion of sensors, algorithms, and systems. The studies performed on different aspects should be connected organically. Therefore, future works on driver studies should focus on the dynamic integration of different units. Driver intention has been studied on conventional vehicles and it has been proved that the intention can be inferred before driver take actions. However, human intention mechanism inside the human brain is still unclear, whether a human intention model can be described into the mathematical form is one of the most challenge work for future driver intention study. If this question can be answered, future intelligent vehicles will be smart enough to prevent the accidents from happening.

Despite the great potential of these technologies, more efforts are required to transfer the theoretical research to the practical side. Future transportation system will have its own challenges and chances. Future transportation system should adopt the common existence of conventional vehicles and autonomous vehicles, which need a holistic perspective of view to study the cooperation between different vehicles. How the human driver influences the surrounding autonomous vehicles and how the autonomous vehicles response to the human drivers? As far as we can see, the development of intelligent transportation system still need new technology to motivate.

APPENDICES

PAPER VII – Driver Workload Estimation

Driver workload estimation using a novel hybrid method of error reduction ratio causality and support vector machine

Authors:

Yang Xing, Chen Lv, Dongpu Cao, Huaji Wang, Yifan Zhao

This paper has been published by

Measurement (2017)

Abstract

Measuring driver workload is of great signi fi cance for improving the understanding of driver behaviours and supporting the improvement of advanced driver assistance systems technologies. In this paper, a novel hybrid method for measuring driver workload estimation for real-world driving data is proposed. Error reduction ratio causality, a new nonlinear causality detection approach, is being proposed in order to assess the correlation of each measured variable to the variation of workload. A full model describing the relationship between the workload and the selected important measurements is then trained via a support vector regression model. Real driving data of 10 participants, comprising 15 measured physiological and vehicle-state variables are used for the purpose of validation. Test results show that the developed error reduction ratio causality method can e ff ectively identify the important variables that relate to the variation of driver workload, and the support vector regression-based model can successfully and robustly estimate workload.

Index Terms - Driver workload estimation, hybrid methods, causality detection, machine learning, real driving data.

A.1 Introduction

Intelligent vehicles have been gaining increasing attention from both academia and industrial sectors [1]. The field of intelligent vehicles exhibits a multidisciplinary nature, involving transportation system, automotive engineering, information technology, energy, and security [2]–[9]. Intelligent vehicles have increased their capabilities in highly, and even fully, automated driving. However, unresolved problems do arise due to strong uncertainties surrounding driving experience and complex driver-vehicle interactions. Before transitioning to fully autonomous driving, driver behaviour should be better understood and integrated to enhance vehicle performance and traffic efficiency [10] [11].

Measuring driver workload is of great significance for improving the understanding of driver behaviours, and worthwhile investigating for the purposes of enhancing driver-vehicle interactions [12] [13]. The workload indicates the proportion of an operator's limited capacity that is needed to conduct a specific task [14]. Driving tasks also require drivers to allocate certain amounts of physical and cognitive workload. A driver's workload is dynamically varied with their different driving behaviours, including straight-line driving, cornering, U-turns, rapid acceleration and deceleration, shifting gears, and changing lanes. Furthermore, the level and the variation of drivers' workload that are affected by the above behaviours could be also influenced by many subjective and objective factors, including driving skills, driving styles, trip objectives, personal tendencies, gender, road conditions, traffic conditions, and so on.

A lot of studies focusing on the measurement and estimation of drivers' workloads have been conducted via different methodologies in recent years. In [15], a method of quantifying driver's workload with five discrete levels was proposed by subjective measurement of vehicle data. In [16], the correlation between distraction condition and drivers' mental load was investigated. The results showed that three variables, namely driver's left-pupil size, skin conductance and pulse-to-pulse interval, could be used for efficiently identifying a driver's distraction. In [17], the driver's subjective mental workload and the multiple task performance were modelled through a proposed queuing network method. In [18], drivers' workloads under lane change manoeuvres were investigated through driving simulations. During simulations, the drivers were required to verbally rate the level of their workloads. In [18], the drivers' physiological information, including the electrocardiogram (ECG) signals, eye blinking, pupil diameters and head

rotational angles, was measured and used to estimate the workload in a driving simulation scenario. Although the above studies provided multiple potential means to estimate drivers' workload quantitatively by using various physiological signals of drivers, all these studies were performed in the simulation environment, which could not reflect the real driving situations and therefore replicate and measure the impact of the potential uncertainties.

Outside of the simulation environment, driver workload estimation with data from real driving environment has also been studied. Analysis of drivers' workload was conducted using the driving data of a real vehicle [19]. EEG data was collected by a sensor mounted on the driver's head during actual driving conditions. The experimental data showed that the EEG signals increased when the vehicle speed went over a threshold limit. Moreover, with respect to the driving scenarios, the EEG signals tended to rise with left cornering and downhill. However, the EEG measurement is very sensitive to external disturbances [20] [21]. In this analysis, the original EEG data was used without filtering noise signals caused by vehicle vibrations and other factors, which may affect the reliability of results. Nevertheless, the existing research in driver workload estimation is mainly in the stage of driving simulations, and methodology of measuring workload with real vehicle data is still very challenging, it is still worthwhile improving this.

To further enhance the algorithm of drivers' workload measurement in real world driving situations, in this study, a novel hybrid method of Error Reduction Ratio Causality and Support Vector Machine (ERRC-SVM) is being proposed. To evaluate the performance of the proposed method, this paper uses a real-world driving data set, including driver's physiological states and vehicle states.

A.2 The Hybrid Methods

A.2.1 Methodology

This paper proposes a novel method, called ERRC-SVM that combines the knowledge in nonlinear system identification and machine learning to effectively estimate the driver's workload.

There are a variety of sensors available to provide signals that may help to estimate workload, where the sampling rate of each measure could be different. Most data-driven modelling techniques require that the observed measurements have the same sampling

rate. A data pre-process procedure is usually required, such as noise filtering, removing mean or normalization, to prepare the data as required by the methods being used.

There are different methods for developing predictive models and it is very challenging to maintain balance between including too many variables (and therefore loss of precision) and omitting important variables (and therefore risk biased prediction) [22]. Including too many types of measurements will a) increase the cost of system; b) cause the overfitting problem if the number of observed data is limited. It is, therefore, important to remove the irrelevant or less relevant variables before modelling. This paper proposes to use a novel causality detection method, named Error Reduction Ratio Causality (ERRC), to assess the importance of each available measure to the variation of workload using a Nonlinear Finite Impulse Response (NFIR) model.

A full model to describe the relationship between the workload and the selected important input variables can then be trained using a support vector regression (SVR) model, where the training data can be either randomly selected if no temporal lag is considered, or a sequence of data. The model can then be applied to the testing data to evaluate the performance. The routine of the introduced method can be summarised by Table A - 1.

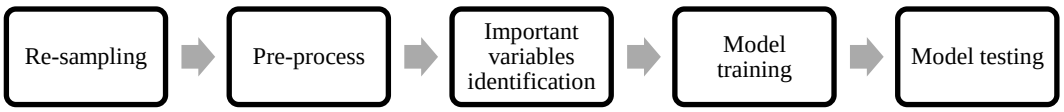

Figure A - 1. Routine of the proposed ERRC-SVM method.

A.2.2 Identification of Important Variables Using EERC

The NFIR model, also known as the Volterra Non-linear Regressive with eXogenous (VNRX) Inputs model, can be used to represent a multi-inputs and single-output system. The model can be expressed as:

$$y(k) = f\left(u_1{}^{[k-1]}, u_2{}^{[k-1]}, \dots, u_R{}^{[k-1]}\right) + \varepsilon(k) \qquad \textbf{(A-1)}$$

where $k(k = 1,2,..)$ is a time index, R is the number of the system inputs, f is some unknown linear or non-linear mapping which links the system output y to the system inputs $u_1, u_2, \dots, u_R$; $\varepsilon(k)$ denotes the model residual. The symbol $u_i{}^{[k-1]}(i = 1,2, \dots, R)$ denotes the past information of the input u_i, which can be expanded as

$$u_i{}^{[k-1]} = \cup_{j=0}^{n_i} u_i(k-j) \qquad \textbf{(A-2)}$$

where n_i is the maximum temporal lag to be considered for the input u_i.

A commonly employed model type to specify the function f in Equation. (1) is a polynomial function [23, 24], which can be expressed as

$$y = \theta_0 + \sum_{m=1}^{N} \theta_m \emptyset_m + \varepsilon \quad \textbf{(A-3)}$$

where $\emptyset_m$ is the m^{th} model term generated from all input vectors; θ_m is the corresponding unknown parameters; N is the total number of potential model terms. Note that $\emptyset_m$ is, in general, non-linear.

If the inputs and output of a system are observable, the model (3) can then be identified. In this paper, the orthogonal least squares (OLS) algorithm [25], is used to determine the model structure from the observations but without estimating the unknown parameters. The OLS algorithm is a popular approach that has been widely used in non-linear system identification where it searches through all the possible candidate model terms to select the most significant model terms which are then included to build model term by term. The significance of each selected model term is measured by an index, called the Error Reduction Ratio (ERR), which indicates how much of the variance change in the system response, in percentage terms, can be accounted for by including the relevant model terms.

Consider a function in linear-in-the-parameters form:

$$y(k) = \sum_{i=0}^{N} \theta_i p_i(k), k = 1,2, \dots, M \quad \textbf{(A-4)}$$

where $y(k)$ is the dependent variable or the term to regress upon, $p_i(k)$ are regressors, θ_i are unknown parameters to be estimated, M denotes the number of data points in the data set, and N denotes the number of terms in the model that is yet determined. Equation (4) can be written as

$$Y = P\Theta \quad \textbf{(A-5)}$$

where

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(M) \end{bmatrix}, P = \begin{bmatrix} P^T(1) \\ P^T(2) \\ \vdots \\ P^T(M) \end{bmatrix}, \Theta = \begin{bmatrix} \theta(1) \\ \theta(2) \\ \vdots \\ \theta(M) \end{bmatrix} \quad \textbf{(A-6)}$$

where

$$P^T(k) = \left(p_1(k), p_2(k), \dots, p_N(k)\right) \quad \textbf{(A-7)}$$

Matrix P can be decomposed as $P = W \times A$ where

$$W = \begin{bmatrix} w_1(1) & w_2(1) & \dots & w_N(1) \\ w_1(2) & w_2(2) & \dots & w_N(2) \\ \vdots & \ddots & \ddots & \vdots \\ w_1(M) & w_2(M) & \dots & w_N(M) \end{bmatrix} \quad \textbf{(A-8)}$$

and A is an upper triangular matrix with unity diagonal elements

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} & \dots & a_{1N} \\ & 1 & a_{23} & \dots & a_{2N} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & a_{N-1N} \\ & & & & 1 \end{bmatrix} \quad \textbf{(A-9)}$$

Therefore, Equation (5) can be rewritten as

$$Y = WG \quad \textbf{(A-10)}$$

where $G = A\Theta = [g_1 \quad g_2 \quad \dots \quad g_N]^T$

Now we introduce how to estimate the importance of each input variable to the variation of the system output. Initially, set values $a_{11} = 1, w_1(k) = p_1(k)$, and calculate

$$\hat{g}_1 = \frac{\sum_{k=1}^{M} w_1(k)y(k)}{\sum_{k=1}^{M} {w_1}^2(k)} \quad \textbf{(A-11)}$$

For $j = 2,3, \dots, M$, set $a_{jj} = 1$ and then calculate

$$a_{ij} = \frac{\sum_{k=1}^{M} w_i(k)p_j(k)}{\sum_{k=1}^{M} {w_i}^2(k)} \quad \textbf{(A-12)}$$

where $i = 1,2, \dots, j-1$. Next calculate

$$w_j(k) = p_j(k) - \sum_{i=1}^{j-1} a_{ij} w_i(k) \quad \textbf{(A-13)}$$

and

$$\hat{g}_j = \frac{\sum_{k=1}^{M} w_j(k)y(k)}{\sum_{k=1}^{M} {w_j}^2(k)} \quad \textbf{(A-14)}$$

The ERR value for each team p_i, as a criterion to select the model structure, is defined as

$$ERR_i = \frac{{\hat{g}_i}^2 \sum_{k=1}^{M} {w_i}^2(k)}{\sum_{k=1}^{M} y^2(k)} \quad \textbf{(A-15)}$$

Values of ERR range always from 0% to 100%. The larger ERR is, the higher dependence is between this term and the output. It is, therefore, a very important index to indicate the importance of each term to the output. To calculate the contribution of each input variable to the output, the sum of ERR values of all selected terms, denoted by $SERR$, is calculated by

$$SERR = \sum_{i=1}^{N} ERR_i \quad \textbf{(A-16)}$$

Note N is the number of the selected terms, not the number of total candidate terms. The value of $SERR$ describes the percentage explained by the identified model to the system output. If the considered inputs can fully explain the variation of the system output, the value of $SERR$ is equal to 100%. It is an indicator of model performance and uncertainty. The contribution of the i^{th} input variable, u_i, to the variation of the system output, denoted as $ERRC_i$, is defined as the sum of ERR values of the selected terms that include this input variable. Because some selected terms may involve more than one input variable due to nonlinearity, the sum of $ERRC_i$ for all input variables can be greater than$SERR$. To over this problem, the value of $ERRC_i$ is normalised and is written as:

$$ERRC_i = \frac{\sum_{j=1}^{N}(ERR_j | u_i \in \emptyset_j)}{\sum_{p=1}^{R}\sum_{j=1}^{N}(ERR_j | u_p \in \emptyset_j)} \times SERR \qquad \textbf{(A-17)}$$

The value of $ERRC_i$ should be always between 0% and 100%. This paper proposes to use this value to represent the importance of each measures to estimate the driver workload.

A.2.3 Model Estimation Using SVM

After the identification of important measures, this paper proposes to use a support vector machine (SVM) to identify the driver workload model. Being a popular machine learning method, SVM was firstly designed to solve binary classification problems by maximizing the margin between different data classes [26]. After that, it was developed to be suitable for both multiple classes classification and regression problems [27]. By utilizing the kernel technique, SVM can map linearly inseparable low dimensional data into separable high dimensional data and construct classification hyperplanes efficiently.

In this paper, a support vector regression (SVR) model is used. Considering a system with a single input and single output, given data set $D = \{x_1, y_1), (x_2, y_2), \ldots (x_M, y_M)\}$, where $x_1, x_2, \ldots x_M$ and $y_1, y_2, \ldots y_M$ are observed input and output respectively. The goal of the SVR model is to find a function $f(x)$ that has at most ε deviation from actual target y_i. Meanwhile, the linear decision boundary $f(x) = wx + b$ should be as flat as possible. Therefore, by applying a soft margin, the loss function for SVR can be represented as:

$$\min \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \qquad \textbf{(A-18)}$$

subject to

$$\begin{cases} \boldsymbol{y_i - \langle w, x_i \rangle - b} \leq \boldsymbol{\varepsilon + \xi_i} \\ \boldsymbol{\langle w, x_i \rangle + b - y_i} \leq \boldsymbol{\varepsilon + \xi_i^*} \\ \boldsymbol{\xi_i, \xi_i^*} \geq \boldsymbol{0} \end{cases} \quad \textbf{(A-19)}$$

where $\xi_i$and ξ_i^* is the slack variables in the soft margin loss function that allows error cases larger than ε. $C > 0$ controls the trade-off between the flatness of function $f(x)$ and the tolerate ratio for the cases that has larger deviations than ε. To efficiently solve the optimization problem, the linear decision function can be further nonlinearized with kernel function, which has the following form

$$K(x_i, x_j) = \emptyset(x_i)^T \emptyset(x_j) \quad \textbf{(A-20)}$$

where $\emptyset(x)$ is the mapping function that maps the raw data points into higher dimensions. In this study, a specific kernel function named radial basis function (RBF) is adopted and represented as below.

$$K(x_i, x_j) = \left(-\gamma \left\| x_i - x_j \right\|^2\right), \ \gamma > 0 \quad \textbf{(A-21)}$$

According to the aforementioned concept, the penalty term C and the kernel function parameter γ are the two most important parameters that determine the model of the SVM. Therefore, in this paper, the Genetic Algorithm (GA) is used to iteratively search the optimal C and γ that can maximize the classification margin or minimize the fitting error. Genetic algorithm is a specific form of evolutionary algorithm inspired by the process of natural population selection. The major parts of GA contain encoding, selection, crossover, and mutation. Specifically, encoding is a process of genetic representation that describes the SVM parameters using multi-bits binary values like genes. Each individual in the population group represents one possible value of the SVM parameter pair. The selection process chooses the individuals with good fitness and pass their genes to the next generation. To increase the gene diversity, crossover aims to make the genes better by randomly combining two well-fitted individuals and the mutation prevents the optimization results from being blocked in the local minimum according to the mutation probability. Since the fitting function is difficult to describe using a single equation in this study, the training error is adopted to measure the quality of fitness. Specifically, the candidate C and γ are decoded and fed into the SVM, the training error, which is the average difference between the predicted workload value and the ground truth values. The detailed implementation of this method can be found in [28]. The procedure of model estimation can be summarized by Table A-1.

TABLE A - 1 PROCEDURE OF COMBINING SVR AND GA

1.	**Input**: train data with label
2.	Initialize GA parameters and generate first population
3.	**for** i = 1,2,...,Max generation
4.	Decoding chromosomes
5.	Computing fitness using SVR for all population
6.	Select Population according to individual fitness
7.	Crossover and mutation to create offspring
8.	**end for**
9.	Find best model parameters
10.	Train SVR with optimized Parameters
11.	Test SVR regression model
12.	**Output**: performance Index and optimized parameters for SVR

TABLE A - 2THE CORRESPONDENCE BETWEEN THE VARIABLES AND THE SYMBOLS USED IN THE NFIR MODEL

Symbol	Variable	Symbol	Variable
u1	ECG	u9	Lightning
u2	SCR	u10	Latitude GPS
u3	Temperature	u11	Longitude GPS
u4	Heart Rate (HR)	u12	Accuracy GPS
u5	Heart rate variability (HRV_LF)	u13	Altitude GPS
u6	Acceleration X	u14	Speed GPS
u7	Acceleration Y	u15	Bearing GPS
u8	Acceleration Z		Video Rating

A.3 Dataset and Pre-Processing

Assessing the drivers' workload in a simulator study is hardly possible because drivers always know that they are navigating through a virtual world. Using the proposed method, this paper analyzed a public dataset [29] collected through a real world driving study. The physiological states of the participants, including Skin Conductance Response (SCR), hand temperature and heart rate using ECG, were recorded. The GPS position, brightness level and acceleration were also recorded. Two cameras were used to record the driver's view onto the road, and the participants were asked to rate the workload offline based on these videos to provide the baseline. The video rating of workload is in the range of 0–1000. A value of 1000 indicates a maximum workload. Ten participants (3 females, 7 males) aged between 23 and 57 years took part in these experiments.

A data pre-process step was undertaken before applying the developed method. Since the sampling frequency of each measure is different, the data have been re-sampled at the frequency of 1 Hz. There are 1515 (25 min and 15 sec) data points for each variable. An example of the selected measures and the video rating for the participant 1 is shown in

Figure A - 2. To save space to present and discuss the results, the corresponding mapping between the name of variables and the symbols in Equation. (3) is shown in

Table A - 2. The mean of each input and output was removed as suggested in Section 2.

A.4 Results and Discussions

A.4.1 Identification of Important Variables

Based on Equation. (3), the NFIR model with quadratic terms to establish the relationship between the 15 input variables and the output is proposed to solve the problem and it can be written as

$$y = \theta_0 + \sum_{i=1}^{15} \theta_i u_i + \sum_{i=1}^{15} \sum_{j=i}^{15} \theta_{ij} u_i u_j + \varepsilon \qquad \textbf{(A-22)}$$

To simplify the model, initially, there is no temporal lag of each input is considered. This model includes 136 candidate terms consisting of 16 linear terms $\{\theta_0, \bigcup_{i=1}^{15} \theta_i u_i\}$ and 120 nonlinear terms $\{\bigcup_{i=1}^{15} \bigcup_{j=i}^{15} \theta_{ij} u_i u_j\}$. The proposed method was then applied to calculate the values of ERRC of input variables and the corresponding SERR for all 10 participants. The results can be illustrated by Figure A - 3.

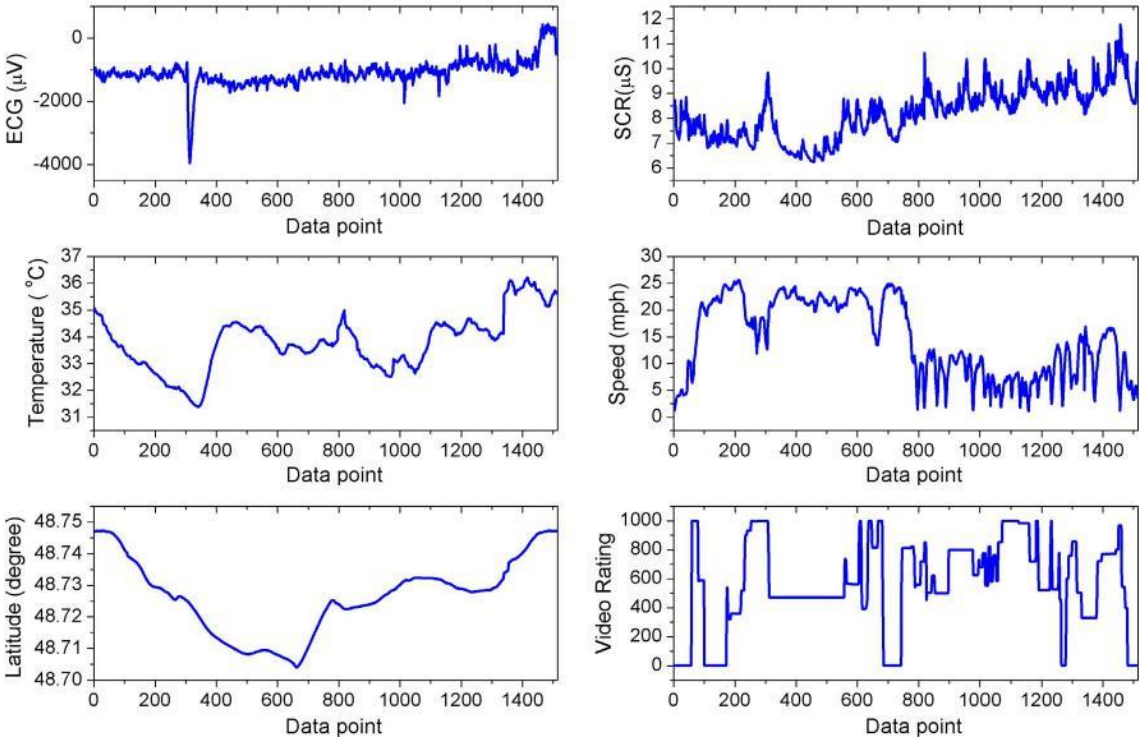

Figure A - 2 Selected measures and the video rating representing the workload for the participant 1.

The detailed value was shown in Table A - 7. The results for all 10 participants exhibit a consistent pattern of contribution of each input. It is observed that some common parameters have a low ERRC values and are therefore less relevant to the driver workload prediction. For example, the values of heart rate, acceleration along different directions, and the accurate GPS signals for all the participants show little contribution to the rating values. However, some input parameters such as ECG, vehicle speed, and latitude GPS signals of most participants show high correlation to the video rating.

To statistically evaluate the importance for each input, the inputs were ranked based on the significance of ERRC values. The second column of Table 3 shows the percentage for each input when the ERRC value is non-zero. The third and fourth column shows the percentage when the ERRC value of the considered input is within the top 3 and the top 5 of all 15 inputs respectively. It can be clearly seen that u1, u2, u3, u10, u11, u13, u14 and u15 have more than 80% probability that the ERRC values are non-zero, and furthermore, they all have appeared at least once in the top 3. However, u15 has only 30% probability that the ERRC values are in the top 5 while others have more than 50% probability. In this study, seven measures (u1, u2, u3, u10, u11, u13 and u14), highlighted in Table 3, are therefore considered as the important measures and will be used in the next section to estimate the driver's workload.

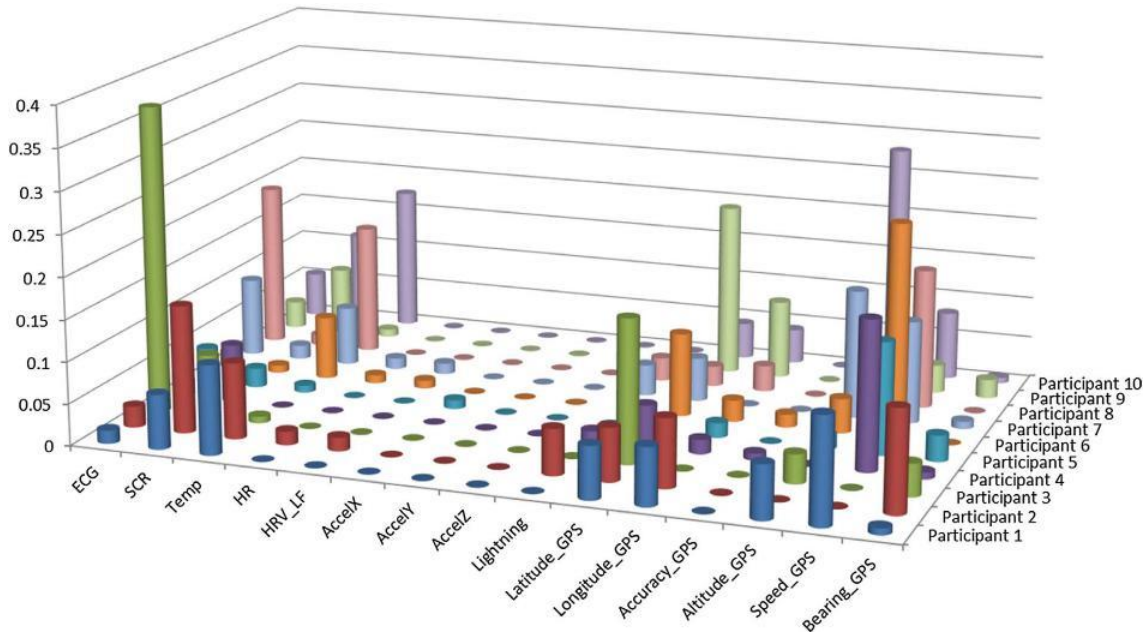

Figure A - 3 Illustration of the calculated ERRC indicating contribution of each input to the variation of video rating for all participants.

The above analysis has not considered the temporal lag of each input. If the past information is considered, the model complexity will be significantly increased. For example, if the past 5 s information (n_I= 5 in Eq. (2)) is considered, the total number of model terms will be increased from 136 to 2926 and increased to 11476 if the past 10 s information is considered. Unless it can significantly improve the performance, including past information should be avoided to save computational time and avoid the model overfitting problem. Table A - 4 shows the comparison of SERR values without and with considering past information of 7 selected inputs. It is clearly shown that there is no significant improvement for the value of SERR except for the participant 2 and 5. Considering the number of sampling data of 1515, this paper only considers the input at time t to predict the output at time t in the next section.

TABLE A - 3 SIGNIFICANT ANALYSIS FOR ALL VARIABLES BASED ON THE ERRC VALUES ACROSS DIFFERENT PARTICIPANTS

Symbol	Non-zero	Top 3	Top 5
u1	100%	40%	60%
u2	100%	50%	70%
u3	90%	50%	60%
u4	30%	0%	0%
u5	30%	0%	0%
u6	10%	0%	0%
u7	0%	0%	0%
u8	0%	0%	0%
u9	40%	0%	10%
u10	100%	40%	50%
u11	80%	20%	70%
u12	20%	0%	0%
u13	70%	20%	50%
u14	80%	60%	80%
u15	90%	20%	30%

A.4.2 Driver Workload Estimation

SVR models for 10 participants were trained based on the selected 7 indicators which are Latitude GPS, Longitude GPS, Altitude GPS, Speed GPS, ECG, SCR, and Temperature. The boundary of the parameter C was in the range of [0.1 100] and γ was in the range of [0.01,1000]. The maximum generation was set as 100, size of population was set as 30, crossover probability was set as 0.4, and mutation probability was set as 0.01. The notion of these parameters can be seen in [28]. For each group of data, 600 data points were randomly selected for the testing purpose and the remaining points were utilized for training. By using the genetic algorithm, the optimal C and γ for 10 participants were estimated and are given in Table A - 5

TABLE A - 4 CALCULATED SERR VALUES INDICATING THE PERCENTAGE OF VARIATION OF THE OUTPUT EXPLAINED BY INPUTS BOTH WITHOUT AND WITH TAKING INTO CONSIDERATION PAST INFORMATION

Participant	**SERR**		
	0 second	**5 seconds**	**10 seconds**
1	0.51	0.51	0.53
2	0.47	0.55	0.58
3	0.68	0.71	0.72
4	0.37	0.37	0.37
5	0.28	0.28	0.39
6	0.56	0.56	0.56
7	0.53	0.54	0.62
8	0.57	0.57	0.58
9	0.50	0.52	0.52
10	0.79	0.79	0.80

TABLE A - 5 THE IDENTIFIED PARAMETERS FOR THE SVR MODELS.

Participant	C	γ
1	97.18	48.75
2	84.60	64.54
3	76.22	73.57
4	69.59	130.36
5	93.94	73.03
6	90.09	133.23
7	85.79	74.56
8	85.82	145.90
9	24.80	70.63
10	84.42	65.78

TABLE A - 6 COMPARISON OF MODEL PERFORMANCE WITH DIFFERENT INPUT COMBINATIONS.

Participant	R^2		
	Human Body Features	GPS Features	GPS and Human Body Features
1	0.68	0.85	0.91
2	0.87	0.67	0.92
3	0.81	0.91	0.97
4	0.54	0.86	0.88
5	0.38	0.69	0.74
6	0.72	0.89	0.94
7	0.72	0.83	0.94
8	0.75	0.86	0.90
9	0.62	0.77	0.83
10	0.95	0.96	0.97
Average	0.70	0.83	0.90

Figure A - 4 illustrates the estimation of workload based on SVR for all the ten participants. Note 600 randomly selected data points (about 1/3 of the total data) for each

participant were used to test the trained SVR model. As shown in the figure, SVR generates precise estimations of workload for all participants, demonstrating that SVR is a robust and reliable estimator for observing workload of different subjects.

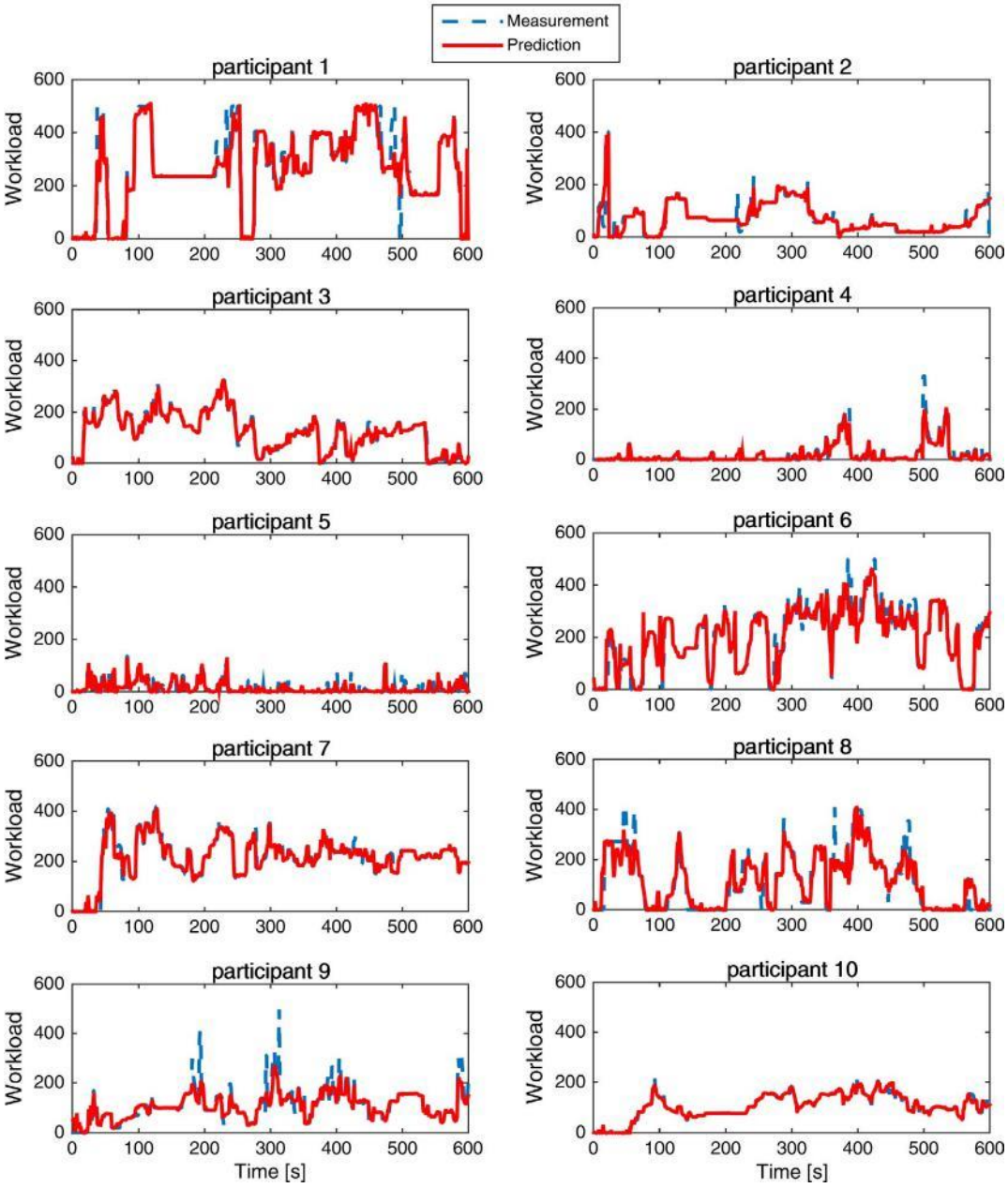

Figure A - 4. The comparison between the predicted workload using the trained SVR models and the recorded video rating for 10 participants.

To evaluate the performance of different input combinations on the workload estimation, three different scenarios were tested and compared: a) human body features only (ECG, SCR, and temperature); b) GPS signals only (latitude, longitude, altitude, and speed) and c) GPS and human body features. The model performance of each participant,

represented by the value of R2 (Pearson correlation coefficient) between the model prediction and recorded data, is shown in Table A - 6.

According to the results in Table A - 6, the average value of R^2 with the workload estimation based on the selected three human body features only is 0.70, while the average value of R^2under the estimation based on the four GPS signals reaches 0.83. This leads to an interesting conclusion that vehicle state measurements are more relevant to driver workload estimation than those physiological signals. It should be noted that some GPS features (e.g. latitude, longitude, altitude) describe the position of the vehicle. In other words, they reflect the road condition, which is subjective. The participants rate the workload based on the video, which captures the road condition and traffic condition. It is therefore not surprising to observe that GPS features perform well in prediction. The final target of this research is to use human body features to estimate the workload. Therefore, this paper is interested in the difference of R^2 between Human Body Features and GPS & Human Body Features. It is observed that the combination of human body features and vehicle's GPS information would construct more relevant feature vectors for estimating driver workload, with an overall improvement of R^2 of 0.2. It has also been observed that this difference varies between different participants.

TABLE A - 7 : CALCULATED ERRC INDICATING CONTRIBUTION OF EACH INPUT TO THE VARIATION OF VIDEO RATING FOR ALL PARTICIPANTS

Input	Participant									
	1	2	3	4	5	6	7	8	9	10
u_1	0.02	0.03	**0.37**	0.02	0.04	0.02	0.10	**0.20**	0.03	0.05
u_2	0.07	**0.15**	0.08	0.07	0.02	0.01	0.02	0.01	0.08	0.11
u_3	0.11	0.09	0.01	0	0.01	0.08	0.07	0.16	0.01	0.17
u_4	0	0.02	0	0	0	0.01	0.01	0	0	0
u_5	0	0.02	0	0	0	0.01	0.01	0	0	0
u_6	0	0	0	0	0.01	0	0	0	0	0
u_7	0	0	0	0	0	0	0	0	0	0
u_8	0	0	0	0	0	0	0	0	0	0
u_9	0	0.06	0	0.01	0	0	0.04	0.03	0.01	0
u_{10}	0.06	0.06	0.17	0.05	0.01	0.10	0.05	0.02	**0.21**	0.04
u_{11}	0.07	0.08	0	0.02	0.02	0.03	0	0.03	0.09	0.04
u_{12}	0	0	0	0.01	0	0.02	0	0	0	0
u_{13}	0.06	0	0.03	0	0.02	0.04	0.15	0	0.02	**0.28**
u_{14}	**0.13**	0	0	**0.18**	**0.13**	**0.26**	**0.12**	0.17	0.03	0.08
u_{15}	0.01	0.12	0.04	0.01	0.03	0	0.01	0	0.02	0.01

For the participant 2 and 10, the difference is smaller than 0.05, which indicates that the selected human body features are sufficient to describe the workload. For the participant 4 and 5, the difference is larger than 0.3, which indicates that the selected human body features are not sufficient and more features (e.g. motion, eye gaze) should be considered in order to better estimate the workload.

A.5 Conclusions

In this paper, a novel hybrid method for measuring driver workload estimation with real-world driving data is proposed. Error reduction ratio causality, a new causality detection approach, is synthesized to quantify the correlation of each measured variable to the variation of workload using a nonlinear finite impulse response model. A full model describing the relationship between the workload and the selected measurements is then identified via a support vector regression model. Real driving data of 10 participants with 15 measured driver's physiological and vehicle state variables are used for the algorithm development, model training, testing and verification. Test results show that the developed error reduction ratio causality method can effectively identify the important variables with the variation of driver workload. Furthermore, the support vector regression-based model can successfully and robustly estimate driving workload. The combination of human body features and vehicle's GPS information constructs more relevant feature vectors for estimating driver workload, resulting in an improved performance of the Pearson correlation coefficient at 0.90. The results demonstrate the feasibility and effectiveness of the proposed novel hybrid methodology for driver workload estimation. It is also concluded that more human body features (e.g. motion, eye gaze) should be considered to better estimate the workload if the subjective GPS features are excluded.

Further work will be carried out in the following areas: collecting more data of real world driving tasks under various situations with more human body features, further refinement of the developed hybrid algorithms for driver workload estimation.

A.6 References

[1] J.L. Gabbard, G.M. Fitch, H. Kim, Behind the glass: driver challenges and opportunities for AR automotive applications, Proc. IEEE 102 (2) (2014) 124–136.

[2] W. Hernandez, Optimal estimation of the relevant information coming from a rollover sensor placed in a car under performance tests, Measurement 41 (1) (2008) 20–31.

[3] J.C. Castellanos, F. Fruett, Embedded system to evaluate the passenger comfort in public transportation based on dynamical vehicle behaviourwith user's feedback, Measurement 47 (2014) 442–451.

[4] G. Andria, F. Attivissimo, A. Di Nisio, A.M.L. Lanzolla, A. Pellegrino, Development of an automotive data acquisition platform for analysis of driving behavior, Measurement 93 (2016) 278–287.
[5] Q. Zhang, Q. Wu, Y. Zhou, X. Wu, Y. Ou, H. Zhou, Webcam-based, non-contact, real-time measurement for the physiological parameters of drivers, Measurement 100 (2017) 311–321.
[6] M.A. Sotelo, Electrical, connected, and automated transportation editor's column, IEEE Intell. Transp. Syst. Mag. 8 (2) (2016) 2–2.
[7] V. Faure, R. Lobjois, N. Benguigui, The effects of driving environment complexity and dual tasking on drivers' mental workload and eye blink behavior, Transp. Res. Part F Traffic Psychol. Behav. 40 (2016) 78–90.
[8] C. Marina Martinez, X. Hu, D. Cao, E. Velenis, B. Gao, M. Wellers, Energy management in plug-in hybrid electric vehicles: recent progress and a connected vehicles perspective, 1–1, IEEE Trans. Veh. Technol. (2016).
[9] D. Pecchini, R. Roncella, G. Forlani, F. Giuliani, Measuring driving workload of heavy vehicles at roundabouts, Transp. Res. Part F Traffic Psychol. Behav. 45 (2017) 27–42.
[10] P. Choudhary, N.R. Velaga, Analysis of vehicle-based lateral performance measures during distracted driving due to phone use, Transp. Res. Part F Traffic Psychol. Behav. 44 (2017) 120–133.
[11] M. Niezgoda, A. Tarnowski, M. Kruszewski, T. Kamiński, Towards testing auditor-y–vocal interfaces and detecting distraction while driving: A comparison of eye-movement measures in the assessment of cognitive workload, Transp. Res. Part F Traffic Psychol. Behav. 32 (2015) 23–34.
[12] M. Chan, S. Nyazika, A. Singhal, Effects of a front-seat passenger on driver attention: An electrophysiological approach, Transp. Res. Part F Traffic Psychol. Behav. 43 (2016) 67–79.
[13] B. Okumura, M.R. James, Y. Kanzawa, M. Derry, K. Sakai, T. Nishi, D. Prokhorov, Challenges in perception and decision making for intelligent automotive vehicles: a case study, IEEE Trans. Intell. Veh. 1 (1) (2016) 20–32.
[14] R.D. O'Donnell, F.T. Eggemeie, Workload assessment methodology, in: K.R. Boff, L. Kaufman, J.P. Thomas (Eds.), Cognitive Processes and Performance, John Wiley and Sons, 1986.
[15] S. Sega, H. Iwasaki, H. Hiraishi, F. Mizoguchi, Verification of driving workload using vehicle signal data for distraction-minimized systems on ITS, in: 18th ITS World Congress, 2011.
[16] Y. Liang, M.L. Reyes, J.D. Lee, Real-time detection of driver cognitive distraction using support vector machines, IEEE Trans. Intell. Transp. Syst. 8 (2) (2007) 340–350.
[17] Wu. Changxu, Yili Liu, Queuing network modeling of driver workload and performance, IEEE Trans. Intell. Transp. Syst. 8 (3) (2007) 528–537.
[18] E.T.T. Teh, S. Jamson, O. Carsten, How does a lane change performed by a neighbouring vehicle affect driver workload?, in: 19th ITS World Congress, 2012.
[19] J.-B. Lim, S.-B. Lee, K.-H. Kim, S.-Y. Kim, J.-S. Choi, A study of the relationship between driver's anxiety eeg and driving speed in motorway sections, J. Korean Soc. Saf. 27 (3) (2012) 167–175.
[20] Y. Zhao, S.A. Billings, H.-L.W. Wei, P.G. Sarrigiannis, A parametric method to measure time-varying linear and nonlinear causality with applications to EEG data, IEEE Trans. Biomed. Eng. 60 (11) (Nov. 2013) 3141–3148.
[21] Y. Zhao, S.A. Billings, H. Wei, F. He, P.G. Sarrigiannis, A new NARX-based Granger linear and nonlinear casual influence detection method with applications to EEG data, J. Neurosci. Methods 212 (1) (2013) 79–86.
[22] P.A. Murtaugh, Methods of variable selection in regression modeling, Commun. Stat. - Simul. Comput. 27 (3) (1998) 711–734.
[23] H.L. Wei, S.A. Billings, J. Liu, Term and variable selection for non-linear system identification, Int. J. Control 77 (1) (2004) 86–110.
[24] S. Chen, S. Billings, Representations of non-linear systems: the NARMAX model, Int. J. Control 49 (3) (1989) 1013–1032.
[25] S.A. Billings, Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-temporal Domains, John Wiley & Sons, 2013.
[26] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[27] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (3) (2004) 199–222.
[28] F. Rong-en, P. Chen, C. Lin, Working set selection using second order information for training support vector machines, J. Mach. Learn. Res. 6 (2005) 1889–1918.
[29] S. Schneegass, B. Pfleging, N. Broy, A. Schmidt, F. Heinrich, A data set of real world driving to assess driver workload, in: Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI '13, 2013, pp. 150–157.