Fakultät für Informatik
Technische Universität München

# Enabling Context Prediction Architectures in Intelligent Vehicle Platforms

**Sina Shafaei**

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktors der Naturwissenschaften (Dr. rer. nat.)**

genehmigten Dissertation.

**Vorsitz:**

Prof. Gudrun Klinker, Ph.D.

**Prüfende der Dissertation:**

1. Prof. Dr.-Ing. habil. Alois Christian Knoll
2. Prof. Dr. Guang Chen

Die Dissertation wurde am 26.01.2022 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 09.07.2022 angenommen.

# Abstract

Automated driving is an inevitable future for the automotive domain. It requires a considerable increase in the level of autonomy for different aspects and functions of the cars and, in some cases, redesigning the entire application itself. Furthermore, the latest technologies in the automotive field are mainly data-driven; hence, the reliance of applications on data increases, especially by introducing new data sources to vehicle platforms. This data is not exclusively originated by the internal applications but will also be provided by external sources, particularly by enabling technologies such as C2C and C2X. The management of this massive amount of data requires well-established context prediction architectures to increase the performance of the vehicle platform and its robustness while maintaining the desired context-awareness for the involved applications. The driver is the key element of driving tasks from planning to controlling the vehicle; therefore, it has a significant impact on creating and utilizing the context in design-time and run-time phases. This work aims to investigate two primary domains of safety assurance and emotional awareness, as the eminent fields which consider a principal role for the driver in their development chain, in order to outline the critical related challenges in enabling context prediction architectures and respectively to provide practical solutions for them with the ultimate goal of enhancing context-awareness in an intelligent vehicle. In the design-time phase of automotive applications, the efficient enforcement of safety standards and safe driving rules into AI-based driving agents is crucial yet challenging due to the heterogeneity level of the AI and safety domains. To tackle this issue and facilitate the integration process, we propose a novel safety violation identification framework deployed on top of the CARLA simulator to fill in the gap between the AI application developers and safety engineers. Respectively, for the run-time phase of the applications, we present an adaptive safety monitoring approach to ensure the safe operation of the driving agents. Emotional awareness and its relying applications is a substantial determinant in the context prediction paradigm. We examine the role of in-cabin behavior-based emotional indicators of the driver and their integration into multimodal emotion recognition systems to increase the robustness of the recognition architectures practically and efficiently. In this regard, we pick the steering wheel angular velocity and vehicle acceleration intensity as the primary lateral and longitudinal driving factors of the driver. We model the selected signals to extract the patterns of abnormal changes in behavior and then map them to respective emotional states. Additionally, we utilize the outcome to develop a multimodal feature vector for the recognition pipeline. We demonstrate the performance of the newly designed multimodal recognition architecture on the data collected through a real car simulator in the lab environment. Finally, we conclude our work by representing an API designed

to address the privacy-related issues of the data owners in data sharing and facilitating future works in this domain.

# Zusammenfassung

Automatisiertes Fahren ist unausweichlich für die Zukunft des Automobilbereich. Dies erfordert eine erhebliche Erhöhung des Autonomiegrades in verschiedenen Bereichen und Funktionen des Fahrzeuges und in einigen Fällen sogar eine vollständige Neugestaltung. Die meisten neuen Technologien im Automobilbereich sind größtenteils datenzentriert, was die Abhängigkeit von Daten steigen lässt, insbesondere auch durch immer weitere Datenquellen in den Fahrzeugplattformen. Diese Daten stammen nicht nur aus den Fahrzeuginternen Anwendungen, sondern auch von externen Quellen wie z.B. durch Technologien wie etwa C2C und C2X. Um diese riesigen Datenmengen handhaben zu können, sind Architekturen für die Kontextvorhersage erforderlich, die gleichzeitig die Leistung und Stabilität der Fahrzeugplattform erhöhen und die gewünschte Kontextsensitivität bei den betroffenen Anwendungen gewährleistet. Im Mittelpunkt der Fahraufgaben steht der Fahrer, das umfasst die Planung bis hin zur Steuerung des Fahrzeuges. Daher hat er einen erheblichen Einfluss auf die Erstellung und Nutzung des Kontextes während des Entwurfes und zur Laufzeit. Das Ziel dieser Arbeit ist es die beiden Gebiete, der Sicherheit und des emotionalen Bewusstseins, als die Kernbereiche zu untersuchen, die den Fahrer als Haupttreiber in der Entwicklungskette sehen, um die damit kritischen Herausforderungen bei der Ermöglichung von Architekturen zur Kontextvorhersage zu identifizieren bzw. praktikable Lösungen für diese zu finden, mit dem obersten Ziel der Verbesserung der Kontextsensitivität in einem intelligenten Fahrzeug. In der Entwurfsphase von Automobilanwendungen ist die effiziente Sicherstellung von Sicherheitsstandards und "sicherer" Verkehrsregeln in KI-basierten Fahragenten von entscheidender Bedeutung, stellt aber aufgrund der Heterogenität der KI- und Sicherheitsdomänen eine Herausforderung dar. Um dieses Problem zu lösen und den Integrationsprozess zu erleichtern, stellen wir ein neuartiges Framework zur Identifizierung von Sicherheitsverletzungen, dem "Safety Violation Identification Framework", vor, das auf dem CARLA-Simulator aufbaut und darauf abzielt die Lücke zwischen Entwicklern von KI-Anwendungen und Sicherheitsingenieuren zu schließen. Wir einen adaptiven Sicherheitsüberwachungsansatz vor, der zur Laufzeit der Anwendungen den sicheren Betrieb der Fahragenten gewährleistet. Emotionales Bewusstsein und seine Anwendung ist ein entscheidender Faktor im Paradigma der Kontextvorhersage. Wir untersuchen die Rolle von verhaltensbasierten emotionalen Indikatoren des Fahrers und deren Integration in multimodale Emotionserkennungssysteme, um die Robustheit der Erkennungsarchitekturen einfach und effizient zu erhöhen. In diesem Zusammenhang wählen wir die Lenkradwinkelgeschwindigkeit und die Fahrzeugbeschleunigung als die primären lateralen und longitudinalen Fahrvariablen des Fahrers aus. Wir modellieren diese ausgewählten Variablen, um Muster anormalen Verhaltensänderungen zu identifizieren und ordnen diese den jeweiligen emotionalen Zuständen zu. Außerdem

nutzen wir diese Ergebnisse, um einen multimodalen Merkmalsvektor für die Erkennungspipeline zu entwickeln. Die Leistungsfähigkeit der neu entwickelten multimodalen Erkennungsarchitektur zeigen wir anhand von Daten, die in einem realen Autosimulator unter Laborbedingungen gesammelt wurden. Zum Abschluss unserer Arbeit stellen wir eine Programmierschnittstelle vor, die den Datenschutz beim Teilen von personenbezogenen Daten verbessert und damit zukünftige Arbeiten in diesem Bereich erleichtern soll.

**Dedicated to:**

Farzad

# Acknowledgments

First and foremost, I have to thank my research supervisor, Prof. Alois Knoll. Without his assistance and dedicated involvement in every step throughout the process of my Ph.D. studies and research work at TUM, this dissertation would have never been accomplished. I'm also grateful to Prof. Guang Chen for being my second supervisor and examiner. I would also like to show gratitude to my first project coordinator Prof. Stefan Kugele as well. His support and enthusiasm for the topic made a strong impression on me, and I have always carried positive memories of his unique leadership during the OSBORNE project. He raised many precious points in our discussions in the domain of safety, and I hope that I have managed to address several of them here.

Getting through my dissertation required more than academic support, and I have many, many people to thank for listening to and, at times, having to tolerate me over the past five years. I can not begin to express my gratitude and admiration for their friendship; Dr. Alexander Lenz, Dr. Morteza Hashemi Farzaneh, and Dr. Mohd Hafeez Osman have been unwavering in their personal and professional support during the time I spent at the Technical University of Munich. Furthermore, I must thank everyone, including my dear friend, Emec Ercelik and my partner in crime during our joint project (OSBORNE), Dr. Christoph Segler. Besides, I cannot forget to thank my kanka, Farzad Nobakht, may his gentle soul rest in peace. Without his tremendous understanding and encouragement over the past few years, it would be impossible to complete my study. I also place my sincere sense of appreciation on record to Ute Lomp and Amy Bücherl, who wholeheartedly supported me through this venture.

Most importantly, none of this could have happened without my family; my parents Reza and Maryam, my younger brother Ali and my beloved partner Sahar, who offered their encouragement through phone calls and messages every week - despite my limited devotion to correspondence. It would be an understatement to say that, as a family, we have experienced some ups and downs in the past five years. You did not let me every time I was ready to quit, and I am forever grateful. This dissertation stands as a testament to your unconditional love and encouragement.

# Glossary

| | |
|---|---|
| ACC | Adaptive Cruise Control |
| ADAS | Advanced Driving Assistance System |
| AI | Artificial Intelligence |
| AND | Automatic Nervous System |
| ANN | Artificial Neural Networks |
| API | Application Programming Interface |
| ATTG | Automated Test Trajectory Generation |
| AU | Action Unit |
| AV | Autonomous Vehicle |
| C2C | Car to Car |
| C2X | Car to X |
| CNN | Convolutional Neural Network |
| CPN | Crash Prediction Networks |
| DBN | Dynamic Bayesian Networks |
| DLT | Direct Linear Transform |
| DMAN | Delta Memory Attention Network |
| DNN | Deep Neural Network |
| ECG | Electrocardiographic |
| ECU | Electronic Control Unit |
| EMG | Electromyogram |
| FACS | Facial Action Coding System |
| HMM | Hidden Markov Models |
| HR | Heart Rate |

| | |
|---|---|
| HRV | Heart Rate Variability |
| HVAC | Heating, Ventilation and Air Conditioning |
| IL | Immitation Learning |
| IoT | Internet of Things |
| KNN | K-Nearest Neighbor |
| LSTM | Long Short Term Memory |
| MDP | Markov Decision Processes |
| MFN | Memry Fusion Network |
| MKN | Multiple Kernel Learning |
| NDRT | None Driving Related Tasks |
| PNP | Perspective-n-Point |
| RBF | Radial Basis Function |
| REST | Representational State Transfer Protocol |
| RL | Reinforcement Learning |
| RNN | Recurrent Neural Network |
| ROI | Region of Interest |
| RSS | Responsibility-Sensitive Safety |
| SAE | Society of Automotive Engineers |
| SAM | Self-Assessment Manikin |
| SFF | Safety Force Field |
| SMO | Sequential Minimal Optimization |
| SMOF | Safety Monitoring Framework |
| SOAP | Simple Object Access Protocol |
| SVM | Support Vector Machines |
| TOR | Take Over Request |
| TTB | Time to Brake |
| TTC | Time to Collision |

| | |
|---|---|
| V&V | Verification and Validation |
| VI | Variational Inference |
| VIA | Validity Interval Analysis |

# Contents

# List of Figures

# List of Tables

# 1 Prologue

In the second half of the last century, the newly deployed economic policies raised the demand for more productivity; thus, they reformed our lives' many aspects. Automobiles were no different from other impacted domains. The concept of cars got reshaped from the *signature* of the families to mainly the *means of commuting*. The colorful manual sedans in the streets got replaced by the black and white spectrum of high-tech efficient hatchbacks and limousines. Over a short period, the control of different vehicle modules inside the car was delegated to intelligent applications. Automated controllers substituted the direct human involvement, from rolling windows to the steering wheel itself and other driving components. Even the engine noises in the cabin are muted thanks to improvements in the design and fabric of the new materials. Undoubtedly these *advancements* have enhanced the overall comfort of the driver as well as the passengers. Moreover, statistical shreds of evidence prove the considerable increase in the vehicle's safety as well. However, these claims are valid only within the context of the newly defined paradigms for the general application of *car* in recent years. The shift of definition for cars from *private property* in a personal context towards becoming the *means of transportation* in a social context has enabled much of this progress and indeed, increased the efficiency, but only for the price of fading the *driving experience* itself.

Let us recall the movie *The Matrix*. The story takes place in a future where humans had consumed all the planet's resources by overutilizing their factories and machines and only realize the dark upcoming realities when it became too late. The only solution left for the humans was to shut down the machines and get rid of them. However, on the other hand, machines have gained a high level of intelligence; therefore, they want to survive and thrive even more. The machines' artificial intelligence (AI) allowed them to stand one step ahead of humans in prioritizing their fate above their owners and taking decisions accordingly. Eventually, the story leads to a deadly conflict between the human race and machines. As the last trick, humans decided to cover the planet's atmosphere with dark gases to cut the machines' primary source of power. It is not hard to imagine that AI was not defeated with this trick. Soon the machines found an alternative solution to survive. They returned to human bodies as the *only* leftover source of power. They enslaved the humans in cubes, putting them in an artificial coma from the moment of birth to death. The movie's overall story is about the heroic fight of a group of humans to acquire their freedom from machines and defeat them. However, the indispensable part of the story is the machines' success in understanding a crucial fact about the living conditions of humans. They have realized that humans can not survive long enough without having the *experience* of life. Therefore, they develop *The Matrix* as the virtual platform which gives the encapsulated humans inside the cubes the simulated pleasure known as the *life experience*.

*The Matrix* movie belongs to the end of the 20th century. Some may argue the relevance of referring to a cinematic experience in scientific works. However, when a research relates to the human-machine interaction area, it is inevitable to face the concerns of individuals and society. The new developments in the automotive industry, especially in increasing the autonomy of the vehicles, are not a new thing. The amount of public attention put on these concepts and represented by the accredited movies such as *The Matrix* at the time of these industrial developments demonstrates an essential demand on addressing the critical issues in this regard. Maintaining a pleasant experience while enhancing human lives' comfort should always be an integral part of each scientific work.

## 1.1 Guide to This Thesis

In this work:

**chapter 2** represents a high-level overview and different aspects of the context-awareness and involved domains in the automotive domain. In addition, the principal elements and the consideration in designing and developing the pro-active architectures for the intelligent vehicle are presented in this chapter. Furthermore, two main user-centered enablers of context prediction architectures, namely *safety assurance* and *emotional awareness* are highlighted as the main objectives and focus points of this work. Finally, the research goals are introduced and listed for detailed and further investigation in the following chapters of the thesis.

**chapter 3** is divided into two main parts of *safety assurance* and *emotional awareness*. The first part outlines the practical definitions for safety in design and run-time phases, particularly in the automotive domain. Then, it introduces the *uncertainty* as one of the main challenges of integrating machine learning-based solutions in intelligent vehicle components and safety-critical applications. The second part includes a thorough overview of the definitions for *emotions* in different research domains. Furthermore, the concept of *affect recognition* in the automotive domain and cabin environment is presented. Finally, different modalities with the highest impact on the recognition process and state-of-the-art fusion architectures for multimodal solutions are represented in detail to form a concrete baseline for the objectives of this work related to emotional awareness.

**chapter 4** outlines the methodologies of the designed experiments to tackle the issues identified in the initial research goals. Regarding safety in the design-time phase, a novel framework for safety violation identification is designed to be deployed in the CARLA simulator in order to fill in the existing gaps between safety engineers and AI developers in the automotive application development chain. Among the proposed monitoring mechanisms in chapter 4 that can mitigate the uncertainty issue in the context prediction architectures, the *crash prediction networks* (CPN) are represented as a novel solution based on interactive learning. For the emotional awareness side, following the necessity of performing an empirical study, behavioral-based factors are modeled as the natural target modalities to induce and map the emotional states of the driver, with the main focus on in-cabin driving metrics and the interaction of the driver with the respective in-cabin components. Finally, the prospects of having a centralized and secure API to ease the coordination between the provided data and developed models in this domain are represented.

**chapter 5** lays out the experiments that are performed to validate the methodologies represented in chapter 4. First, multiple stages of the safety violation identification framework are integrated into the CARLA simulator and evaluated according to the

pre-defined safety violation factors. Based on the designed evaluation criteria, it is demonstrated that this framework considerably facilitates the integration of safety rules and standards in the AI development chain for the driving agents. Then, two versions of the CPN concept, namely the *simple* and the *Spatio-temporal*, are implemented and evaluated in the CARLA simulator against static and dynamic obstacles. This set of experiments attempts to depict the potential benefits of the CPN as a novel safety monitoring approach in this domain, tackling the uncertainty issue and demonstrating its performance in this regard as well. On the other hand, for the emotional awareness, after an extensive empirical study to validate the general assumptions regarding different perspectives on emotions and their correlation with in-cabin behavior during driving, a set of experiments have been conducted on the VIRES virtual test drive (VTD) simulator to collect the required data for populating the desired evaluation datasets. Furthermore, for the behavior-based emotional indicators, vehicle acceleration intensity and steering wheel angular velocity are considered in the multimodal system design and evaluated accordingly to reveal the benefits of utilizing behavioral-based factors in maintaining emotional awareness. Finally, a unified API is developed to illustrate the interests of centralizing the developed models and modalities for future developments and cooperation between the data owners and model developers.

**chapter 6** summarizes the experiments designed and evaluated in this work to tackle the challenges of developing context prediction architectures in an intelligent vehicle from the perspective of two main user-centered domains of safety assurance and emotional awareness. This closing chapter also clarifies the matters that remained out of the scope of this work and must be addressed in the following future works.

## 1.2 Scientific Contributions

Parts of this thesis have been previously published at international peer-reviewed conferences, peer-reviewed workshops, or investigated during special workshop sessions. The general idea and the primary motivation of this work regarding context prediction architectures in intelligent vehicles, the impacting domains and the associated challenges of them were discussed thoroughly in:

- Sina Shafaei, Fabian Müller, Tim Salzmann, Morteza Hashemi Farzaneh, Stefan Kugele, Alois Knoll: **Context Prediction Architectures in Next Generation of Intelligent Cars**. 21st IEEE International Conference on Intelligent Transportation Systems, 2018

With the help of a manoeuvre planning system, as a leading use case application in autonomous vehicles, we outlined the safety challenges in both the design-time and the run-time phases. Then, we thoroughly examined the possibility of four potential safety monitoring mechanisms that can be deployed and efficiently operate in different driving conditions, and respectively demonstrated the potential advantages and possible shortcomings of each one of them in detail in:

- Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman and Alois Knoll: **Uncertainty in Machine Learning: A Safety Perspective on Autonomous Driving**. First International Workshop on Artificial Intelligence Safety Engineering, SAFECOMP Conference, 2018

The first focus point of our work in the safety domain was in the design-time phase. We developed a safety violation identification framework that allows the effective enforcement of safety measures and standards into the development chain and facilitates the identification of safety violations in design-time. We were able to demonstrate promising results in detecting and identifying the relationships between the "actions that are taken" by the driving agent, "type of the detected safety violations", and the "recognition of safety-critical situations" in driving scenarios in the CARLA simulator environment based on the set of pre-defined safety measures. The experiments and outcome of the evaluations are published in:

- Lukas Heinzmann, Sina Shafaei, Mohd Hafeez Osman, Christoph Segler and Alois Knoll: **A Framework for Safety Violation Identification and Assessment in Autonomous Driving**. AISafety: The IJCAI-19 Workshop on Artificial Intelligence Safety, 2019

For the run-time phase, we evaluated the chosen proposal for the online safety monitoring mechanism in the CARLA simulator with an exclusive focus on visual sensor data. We demonstrated the advantage of exploiting the expressiveness of neural networks by incorporating contextual data while maintaining the monitor setup's robustness due to its ensemble-like structure. The designed architecture of the monitoring mechanism and the evaluation scenarios, along with the preliminary results on the effectiveness of the newly proposed monitoring concept, are described in detail in:

- Saasha Nair, Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman and Alois Knoll: **Monitoring Safety of Autonomous Vehicles with Crash Prediction Networks**. SafeAI: The AAAI's Workshop on Artificial Intelligence Safety, 2019

- Saasha Nair, Sina Shafaei, Daniel Auge and Alois Knoll: **An Evaluation of "Crash Prediction Networks" (CPN) for Autonomous Driving Scenarios in CARLA Simulator**. SafeAI: The AAAI's Workshop on Artificial Intelligence Safety, 2021

In the emotional awareness sub-domain of this work, we aimed to evaluate the in-cabin behaviour-based emotional indicators, identify the impact of their integration into emotion recognition systems, and enlighten the challenges of developing a multimodal structure that utilizes behaviour-based emotional indicators in its recognition pipeline. For this purpose, we performed an empirical study through a survey to directly incorporate the subjects' inputs; redefined our definition of in-cabin behaviour, and respectively picked the "vehicle acceleration intensity" and "steering wheel angular velocity" as the leading in-cabin behaviour-based emotional indicators for our evaluations. With the help of data collected through the VIRES VTD (virtual test drive) simulator and the designed multimodal architecture, we demonstrated the positive impacts of the fusion of behaviour modalities with the facial expression-based modality for classifying the subjects' emotional states in an in-cabin environment. We also demonstrated the importance of adjusting the length of the observation-prediction window and properly defining the decision-making stage for performing classifications in such systems that must be taken into account due to the dynamically changing environment around the subjects. The following publications cover the evaluation phase of our works in this regard:

- Sina Shafaei, Tahir Hacizade, and Alois Knoll: **Integration of Driver Behavior into Emotion Recognition Systems: A Preliminary Study on Steering Wheel and Vehicle Acceleration**. In: Computer Vision - ACCV 2018 Workshops, 2018

- Mesut Kuscu, Sina Shafaei and Alois Knoll: **Abnormal Driver Behavior Detection for Automated Emotion Recognition (Poster)**. 6th French-German Summer School, Emotion-aware Vehicle Assistants (EVA), 2018

Additionally, the following international workshop has been organized at the 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS 2019), exclusively to represent the main objectives of this work and discuss the open challenges and research directions in this domain with other researchers:

- **International Workshop on Machines with Emotions: Affect Modeling, Evaluation, and Challenges in Intelligent Cars**. Sina Shafaei, Alois Knoll, Radoslaw Niewiadomski, Stefan Kugele, Christoph Segler, Morteza Hashemi Farzaneh. Co-located with IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019

The following publications did not directly contribute to this work but are associated with some of the objectives of this thesis:

- Stefan Kugele, Vadim Cebotari, Mario Gleirscher, Morteza Hashemi Farzaneh, Christoph Segler, Sina Shafaei, Hans-Joerg Voegel, Fridolin Bauer, Alois Knoll, Diego Marmsoler, and Hans-Ulrich Michel: **Research challenges for a future-proof e/e architecture - a project statement**. 15st Workshop Automotive Software Engineering, 2017
- Ana Maria Radut, Sina Shafaei: **A Regression-based Control Approach for Limited Slip Differencial**. TUM (Technical Report), 2017
- Stefan Kugele, David Hettler, Sina Shafaei: **Elastic Service Provision for Intelligent Vehicle Functions**. 21st IEEE International Conference on Intelligent Transportation Systems, 2018
- Christoph Segler, Stefan Kugele, Philipp Obergfell, Mohd Hafeez Osman, Sina Shafaei, Eric Sax, Alois Knoll: **Evaluation of feature selection for anomaly detection in automotive E/E architectures**. 41st International Conference on Software Engineering, Companion Proceedings, 2019
- Mohd Hafeez Osman, Stefan Kugele, and Sina Shafaei: **Run-time Safety Monitoring Framework for AI-based Systems: Automated Driving Cases**. The 26th Asia-Pacific Software Engineering Conference, 2019
- Michael Hammann, Maximilian Kraus, Sina Shafaei, Alois Knoll: **Identity Recognition in Intelligent Cars with Behavioral Data and LSTM-ResNet Classifier**. CoRR abs/2003.00770, 2020

Respectively, the following international workshops have impacted this work:

- **International Workshop on Data Driven Intelligent Vehicle Applications (DDIVA)**. Alois Knoll, Emec Ercelik, Esra Icer, Burcu Karadeniz, Christoph Segler, Sina Shafaei, and Julian Tatsch. Co-located with IEEE Intelligent Vehicles Symposium (IV), 2019
- **2nd International Workshop on Data Driven Intelligent Vehicle Applications (DDIVA)**. Alois Knoll, Emec Ercelik, Esra Icer, Neslihan Kose, Burcu Karadeniz, Christoph Segler, Sina Shafaei, and Julian Tatsch. Co-located with IEEE Intelligent Vehicles Symposium (IV), 2020

Furthermore, the following patent application was also registered aligned with the goals of this work but had no direct influence on the research objectives:

- Christoph Segler, Sina Shafaei: [Patent] **Verfahren, Vorrichtung, Computerprogramm und Computerprogrammprodukt zur Datenbearbeitung für ein Fahrzeug**. Patent Application DE 10 2018 202 348 A1, 2018

# 2 Introduction

## 2.1 Motivation

Machines are usually designed to either fill in the existing *gaps* in our daily activities or replace the humans in performing *boring* or *dangerous* tasks, with the ultimate goal of increasing the level of *comfort* and *safety*. The user acceptance is also an important factor in developing new technologies. All this together indicates the critical role of humans at the epicenter of the development chain. Nowadays, machines can be found at nearly every corner of our surrounding environment while affecting all aspects of our lives, from our bedrooms to offices and workplaces. Establishing a certain level of trust between humans and machines is necessary to maintain sustainability and preserve the high quality of the provided services. However, this *trust* matter and establishing of it is a challenging matter due to the complex nature of most of the involved factors. Having a clear, well-structured perspective that incorporates all the relevant parameters contributing to the overall context is crucial for addressing the existing issues. A subject's actions are usually defined and justified only when the relevant context is thoroughly and adequately addressed. The human user is one of the main elements and, in most of the cases, the root cause in shaping that context and completing the decision-making loop. From a general perspective, context is formed around a subject by identifying the involved factors and parameters and clarifying the existing correlation among them as well as their impact on the subject. Human actions and decisions can be appropriately defined by incorporating the relevant context, such as the time, location, and other factors affecting the decision-making process. Context-aware applications are the ones that rely heavily upon context, and their behavior and decision-making process follow the changes in the context accordingly. Recent technologies and developments aim to quantitate the context and utilize context prediction architectures to enhance the efficiency of the applications and facilitate their functions for the end-users. The automotive domain is also one of the fields that have considered the benefits of context-awareness in its developed applications. The recent trends in this field demonstrate a promising future for this matter; however, it introduces a new set of challenges as well. For example, how we should perceive an *autonomous vehicle*. Is it genuinely an intelligent *agent*, capable of defending its actions and decisions regardless of its user, or is it just a different type of *device* with more options and buttons than a conventional vehicle, under the human user's control? In safety-critical situations, who would take the final decision: the car or the users of the car? There is no consensus on these questions among researchers, developers' communities, and car manufacturers at the moment; however, among all the car makers, Daimler was the first one which recently announced that if it comes to a situation that the car will require to decide to save either the life of its passengers or

the people on the street, the autonomous vehicles built by them, will give priority to its passengers.

This work investigates the involved determinants and hurdles in enabling context prediction architectures and maintaining context-awareness in intelligent vehicles from the perspective of two fundamental and challenging domains. In this respect, as is depicted in Figure 2.1, we outline two areas of *safety assurance* and *emotional awareness* as the focus domains with the highest impact on the development of efficient context prediction architectures in intelligent vehicles. On the one hand, emotional awareness directly influences generating context and contributes to context awareness of the vehicle and its applications from the driver and occupants' side. On the other hand, the safety domain relies extensively on context awareness to enforce efficient safety policies and preserve the required assurances. Excluding the driver and the occupants from the equations of the safety domain makes this field irrelevant. Besides, maintaining emotional awareness inside the cabin environment is essential in determining the acceptance of the vehicle's provided services. A widely utilized application of this phenomenon is the *feedback loop* which is not only used in comfort applications but is utilized in driving dynamics and safety-critical functions as well. Each of these domains contains a sub-set of important elements facing technical issues that we aim to tackle during this work. The considerable complications in developing artificial intelligence-based models and algorithms, such as uncertainty, along with the concerns related to the enforcement of safety standards as well as designing efficient monitoring approaches for the safe operation of the associated applications, are categorized under the *safety assurance* domain with a strong focus on design-time phase. Respectively, the *emotional awareness* side focuses on the in-cabin environment and, more precisely, the human-machine interaction. Investigating this objective requires revising the definitions and the impact of the driver in interaction with vehicle applications inside the cabin and their functions. Besides, modeling the in-cabin emotional behavior is necessary to enable multimodal emotion recognition systems and maintain the occupants' emotional states. Modeling user emotions is a complex challenge by nature. From a technical point of view, it is utilized as the complementary part of the feedback loops for the applications that closely interact with the users, e. g. drivers. The lack of generic models for emotional profiling of the driver and addressing different involved modalities in recognizing emotions inside the cabin enlightens the importance of incorporating different factors and employing multimodality to maintain reliable emotional awareness. In the following, we will investigate each of the areas listed in Figure 2.1 more closely to shed light on the list of open challenges.

### 2.1.1 Context Awareness

The most significant benefit of context-aware applications becomes visible in mobile sensor-rich devices. Smartphones and wearable gadgets have become highly adaptive to the environment they are used in and, therefore, improved the engagement with users. However, the set of sensors and the desired functionality of such devices is limited and well defined. A car, by definition, is also a mobile device. Compared to smartphones, it is equipped with an even richer set of sensors and aims to fulfill many kinds of functionality,

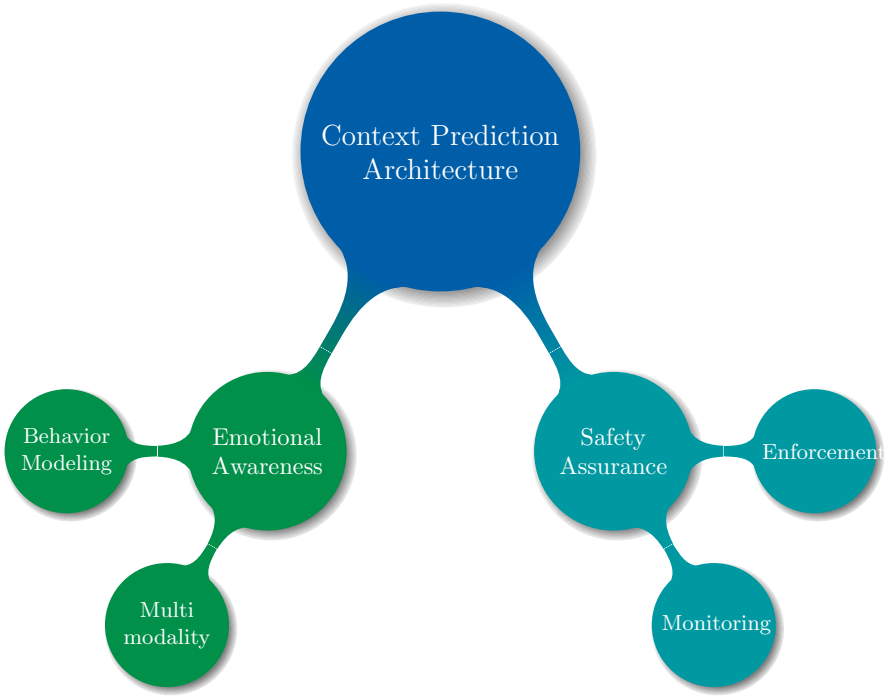



**Figure 2.1:** Enablers of context prediction architectures in intelligent vehicles considered in this work

including highly complex driving tasks. The growing number of intelligent components inside a car leads to a considerable increase in the produced data. The paradigm of *context-awareness* that enables the proactiveness for the applications based on this feature plays a significant role in managing this data while offering numerous prospects and advantages for existing and new applications through the intelligent vehicle.

This leads to developing context prediction architectures that promise reliable solutions in enhancing the comfort of the occupants and vehicle dynamics while maintaining safety standards [13]. The importance of such architectures is also evident in the direction of new advancements in the automotive domain toward increasing the autonomy of vehicles. In referring to automated driving in this work, we consider the level-taxonomy defined by the *SAE* in standard J3016 [14]. It marches from 0 to 5, where level 0 refers to driving entirely in manual mode, and level 5 refers to driving in a fully automated mode without a steering wheel and controlling pedals. In level 1, either the longitudinal or the lateral control is automated. In contrast, in level 2, both degrees of freedom are automated, but the driver has to monitor the vehicle's operation constantly and, thus, is in full response at all times. Cars equipped with level 3 systems can conditionally drive autonomously, while the driver is not required to monitor the system constantly. However, the driver is still the fallback if the conditions are violated and the car issues a *take over request* (TOR). Then, the driver has to regain control over the vehicle within a limited period. This moment is critical, as the driver must understand the current traffic situation within the given time frame, even in situations entirely out of context.

In highly automated driving of level 4, the car can already drive in autonomous mode during specified situations but will continue to operate safely if the TOR is ignored.

Abowd *et al.* define the *context* as *"any information that can be used to characterize the situation of an entity"* [15]. Due to the high amount of sensory data, it is not straightforward to define which sensor input directly forms a/the context; however, the comparable impact of the contributors must be identified. There are already some approaches to enabling context-awareness in automotive software architecture. According to early developments, the context data can be stored on a central server [16] or on a layered architecture that relies on a context provider's services [17]. The complex nature of the produced data set in an intelligent car and the lack of comprehensive modeling and evaluations make it difficult to define a general functional context-based architecture. To achieve this goal, one must acquire considerably broad knowledge and a better understanding of multiple involved domains to evaluate them based on the impact level. More importantly, it is vital to properly recognize and address the challenges to steer in the right direction. Modern context-aware applications can also predict the future context while using the current context to improve the user experience. This feature empowers a huge number of new applications and potential developments accordingly. Furthermore, as the future context is predicted, the application can rely on anticipation by adjusting how it interacts with the user or automatically remodels its functionality.

An automotive context prediction architecture enables a wide range of new driver comfort features and extended driving functionalities. As an intuitive example for comfort applications, the seat and in-cabin temperature could be adjusted before the driver even enters the car. Moreover, a context-aware architecture can considerably increase the energy efficiency of the systems [18]. This feature could even be improved further by enabling context prediction in vehicle platforms, which is a great advantage in modern electric cars with limited energy capacity. Recent developments of machine learning-based applications in intelligent vehicles and the ever-growing utilization of AI make it inevitable to neglect the role of newly developed models in defining and consuming context and respectively enabling proactiveness. Let us consider two typical and basic applications in a vehicle that can be deployed based on a context prediction architecture and benefit from the context-awareness to provide proactive services. These use cases help us to intuitively demonstrate the general concerns and domains of challenges in maintaining context awareness for vehicle applications:

- **Heating, Ventilation and Air Conditioning (HVAC) system:** is a set of functions to improve the comfort of the driver and passengers inside the cabin. While every modern car provides some HVAC system, their context-awareness is limited and does not employ any specific context prediction mechanism. Much of the functionality of an HVAC system is very straightforward and results in simple actions taken by the system; therefore, HVAC is one of the straightforward yet promising use cases that intuitively demonstrates the benefits of context-awareness for providing pro-active comfort services inside the car while outlining the service demands. Our leading application for an HVAC system is the *comfort* functionality, where the inner space temperature of the car is automatically adjusted to ensure

the welfare of the occupants. The effectiveness of this service, and respectively the users' satisfaction, can be detected by the emotion recognition systems designed to monitor the driver and passengers' emotional status and well-being.

- **Adaptive Cruise Control (ACC) system:** describes an essential function for semi and highly automated driving. It enables the car to automatically stay on a lane with a preset velocity or follow another vehicle if it goes slower than the preset velocity. In contrast to HVAC, the ACC only provides a single function to the user. The underlying functionalities of ACC and the generated trajectory make it genuinely more complex than the HVAC system. *Lane detection* is a vital sub-function of ACC, and we consider it the leading application of our ACC use case.

There are different solutions in the automotive domain to maintain the context-awareness and enable the proactiveness in the applications such as those mentioned above. However, the standard set of concerns regarding the potential involved challenges can be grouped and categorized as follow:

**Prediction and Inference Methods** There are widely used methods for prediction and inference in different domains of research related to context prediction. A thorough and detailed comparison of them is represented in [19]. The *sequence prediction* approach is one of the exceptionally researched ones in theoretical computer science. D. Cook *et al.* [20] provides a comprehensive overview of sequence prediction techniques focusing on smart homes. Another approach is based on *Markov chains* which are formal models. Some projects utilized them to address context prediction problems, like the one presented at [21], in which the authors addressed an active device resolution problem, or respectively, the work presented at [22], where the authors used discrete-time Markov chains to predict the driving route. *Dynamic Bayesian Networks* (DBN) is a generalization of Markov models while avoiding some of the Markov model's shortcomings which got used in numerous projects like the user modeling and user goals inference at [23], and respectively predicting the person's indoor movement at [24] by representing the context as DBN where the possibility of visiting the current room depends on several rooms visited previously. Here the duration of staying depends on the current room, exact time of the day, and the exact day of the week. The neural networks in machine learning are gaining more and more success in solving various problems like pattern association, recognition, and function approximation; hence, they are promising enough to be used as a reliable approach here. One of the earliest works represented at [25], where the authors describe an intelligent house that predicts expected occupancy patterns in the house, estimates hot water usage, and the likelihood of entering a zone. Similar use cases based on neural networks for context prediction can be found in works presented at [26, 27, 28]. The majority of these prediction and inference methods are employed for specific use cases and in laboratory environments; hence, it is evident that there is a lack of a general prediction method that performs equally well in all of the use cases and can be utilized in the automotive domain.

In order to select and integrate a method into the context prediction architectures of vehicle platforms, there are a set of concerns that must be taken into account in advance. First, the *knowledge inference* factor must be identified beforehand since some of the state-of-the-art prediction approaches, like neural networks, do not consider the prior knowledge inference in their decision-making process. Moreover, the prediction core must provide a reliability estimation while maintaining the developer's readability to have the option of validating the correctness and verifying the safety of the predicted context or decisions taken. This fact also indicates the importance of the observability factor in decision-making regarding being a white or black box. In most cases, the information loss in pre-processing step is unavoidable. Nevertheless, this loss shall not affect the critical and relevant features in a way that the predicted context ends up in an undesired region. Besides, the other related concern is the mutual dependency between the predicted context and system actions which is still an unsolved issue in the current state of development. The works on *Markov Decision Processes* (MDP) address this issue. MDPs are a plausible and practically effective way to predict the context in situations when Markov models are applicable, and control actions can significantly affect prediction results. However, the prediction of context can also be seen from a relatively different angle:

- **Feature prediction:** For a function, it focuses on finding correlations between the feature subsets (known as the current information) and the output. If such a correlation exists, the information is likely to be part of the function's context. So far, this has not been a focus of context prediction architectures in automotive. However, other fields of study have already started research in this area. For example, finding correlations in big data sets (e. g. buying habits) is of great interest in marketing. Some tools like *association rule learning* or *market basket analysis* aim to find correlations in data. Regarding this issue, evaluating the existing models from other fields and checking the feasibility of integrating them into the automotive domain is undoubtedly beneficial. It is worth mentioning that correlation does not automatically deduce a causal connection. This sentiment has to be accepted and kept in mind at the time of designing the relevant models.

- **Time-series prediction:** This is another alternative to feature prediction. Sigg [29] and Rosa [30] both overview different machine learning models and the corresponding feasibility and integration for context time-series prediction. For a general model, it is imperative to get set or automatically vary the prediction horizon. Recalling the ACC use case, only the past and next seconds are essential and convey helpful information. However, for the HVAC use case, it is required to predict the context for the next minutes or even hours to achieve the desired efficiency. A practical example is when the planned navigation route is over a mountain pass. As the temperature is expected to drop in such locations, the in-cabin heating should be increased. For this aim, it is necessary to inspect the existing models used in traditional prediction architectures and adapt or extend them accordingly.

**Output/Action Inference** Finding a relatively generic model that describes the output/action inference within a context prediction architecture in an intelligent vehicle is challenging. Therein multiple factors play a role. First of all, depending on the functionality, the complexity of the output might be considerably high. For example, in our ACC use case, the output is a structure describing the lanes on the road. Furthermore, the output is much more complex when there is more than one output value, like in the HVAC use case (e. g. desired temperature). Also, the input dimension might vary (as known as dynamic context features), and the data representation in those dimensions can vary too (like normalization, discrete/continuous, etc.). Therefore, it is challenging to find a supervised model which is generic enough to serve all purposes in this regard. Reinforcement learning, however, enables a shift of complexity from the model itself towards finding an *action policy* and a suitable *reward function*. Here the model and learning algorithm is the same for all use cases. The policy and reward function must be defined separately and desirably enforced dynamically for each use case.

**Online Learning** The functionality of the components and applications must be studied regarding the different dimensions of data flow to decide which one could/should be deployed and benefit from online learning. For a plug & play architecture, it is essential to learn the necessary context features online. The other context-related elements have to be highly adaptive and get acquired online since their input dimensions are exposed to potential changes. In this case, most of the desired context-aware functionality must be available in the car at the beginning of the usage or be learned without hesitation. Especially for context prediction, it is essential to learn the desired output of the application quickly. For the use case of HVAC, the deployed algorithm has to acquire knowledge about the desired temperature of an individual after a short period and quickly adapt to changes. Otherwise, it is not an enrichment and will not be accepted by the driver. This procedure is not feasible with pure online learning methods. Hence thorough research into possibilities to combine offline and online information for the training of models is required like Ye *et al.* [31] in representing an online planning approach with regularization included in an autonomous driving system for real-time control of the vehicle. This work itself was originated by the proposals of Gelly *et al.* [32] on *"algorithms that combine the general knowledge accumulated by an offline reinforcement learning algorithm, with the local knowledge found online."*

**End-to-End Learning** Having said that, all context-related components support different aims; they also are highly dependent on each other. If the context features are determined online, as mentioned earlier, the following components must be adaptive. This matter raises the question of whether developing independent models for each component is reasonable. Working towards end-to-end learning could be an alternative in this case. Even a single model for all functionality that receives all information as input and all expected output-actions as output can be plausible. This is a reasonable approach to rely upon for HVAC functionality with its multiple actions of limited complexity. In contrast, this does not seem to be a promising approach for the use cases

like ACC with a highly complex output and underlying structures. On the one hand, finding good models for end-to-end learning is difficult and often a matter of experience. Besides, it is still reasonable to use a reinforcement learning-based approach to shift the focus toward finding a suitable policy instead. On the other hand, an end-to-end learning approach has the advantage that interfaces between components are dynamic by definition.

### 2.1.2 Safety Assurance

In a domain-agnostic model, Varshney *et al.* [33] defines *safety* as the minimization of risk and uncertainty. In other words, it is the absence of failures and dangerous situations. From this perspective, safety is an essential consideration when it comes to automotive software architecture. When we approach an entity, naturally, we investigate its *safeness* from our perspective, but what is *being safe*, and how do we approach it in the automotive domain? The scientific perspective on safety is focused solely on providing high-level definitions and a list of requirements for its adaptations toward the changes in different contexts. Then, the system developers take the definitions and enforce them accordingly into the desired development chains of the applications and systems. However, the abstract definitions of safety are not enough in practice due to the lack of solid development guidelines; therefore, it amplifies the need to have technical mechanisms for the integration purposes and monitor the system's safe functioning, not only in design-time but especially during the run-time phase. These technical mechanisms provide a set of principles to enable safe actions and maintain safety in application functions. Besides, the users of the applications and the systems must *feel* safe in their interactions with the vehicle in order to embrace them. This matter demonstrates the vital role of user preference and its impact on the application's design phase, development chain, and deployment.

The functional safety concerns in automotive software development are addressed by the ISO 26262 [34], an international standard established to ensure that all components have been designed and developed with rigor to ensure safety by minimizing random and systematic failures. Current software methodologies and tools to establish safety assurance in safety-critical systems have been around for a long time and represent quite a mature field [35]. However, the trouble is that these methodologies have been developed for traditional software systems, i. e. the ones that have been explicitly hardcoded to act in a certain way, with a definite set of requirements. However, the conventional verification and validation methods cannot perform as expected in dealing with knowledge inference and machine learning-based solutions. This concern is also a crucial matter in context discovery, inference, and prediction.

Consider the neural networks as one of the promising candidates commonly used to enable the applications' context-based pro-activeness. The main challenges from the safety perspective can be enumerated as *black-box structure*, *implicit specification*, and lack of proper *coverage-based testing* [36]. Splitting the data into *train*, *validation*, and *test* data is one of the popular methods and is extensively used nowadays to ensure that the developed adaptive system works well for the given set of inputs. This method helps

to verify the functioning of a neural network but is not usually extensive enough to be considered a *guaranteed* approach in ensuring context-based safety-critical systems [37]. Being left with only a small set (typically around 20%) of samples to test the model is one of the various arguments for lack of trust in the *train-validation-test* method. This issue will increase the possibility of being ignored for potential cases of high interest and unpredicted context. Test data generation tools can also be seen as a solution to tackle this problem by generating synthetic data points for testing the trained neural networks. According to their correct behavior, this approach is beneficial for the verification procedure of the neural networks by unveiling missing knowledge, as known as context, in fixed neural networks and increasing the confidence in the working of adaptive neural networks [37]. Similarly, rule extraction algorithms can be used to model the knowledge that a neural network has acquired during the training phase. These rules can be generated in a conjunctive or subset selection form. The rules extracted can be manually verified by integrating them into the human-readable format or using a proper model checker tool. This method can be helpful to establish trust in the system, as it augments the explainability of the system. It also aids traceability of the requirements, as one can verify whether the rules depict functional requirements specified for the system. They can also help examine the various functional modes of the system and help to ensure the inclusion of safe operation mode by specific inputs. Considering the advantages of this method, it can be a reliable solution for offline learning systems, wherein in the verification and validation phase, the system designer can extract rules from the network when the training is completed. On the other hand, online feature learning is a *must* for plug & play architectures. Utilizing it creates an added overhead for the system due to the rules needed to be extracted after every iteration to ensure that the learning has been performed as expected. This method can be an expensive and challenging issue in terms of computation and time. However, there are solutions like **online monitoring** which is a technique that uses multiple monitors working together as an oracle to provide information about the functioning of the neural network in order to aid stability and convergence analysis [38]. The goal here is to ensure that the adaptation dynamics do not cause the network to diverge, triggering behavior unpredictably. Data sniffing [39] is an example based on the preceding technique, which studies the data entering and exiting a neural network. If a particular input could pose negative results, the monitors generate an alert and even flag down the data, thereby not allowing it to enter the system. This method is favorable in cases where outliers could degrade the functioning of the system.

### 2.1.3 Emotional Awareness

Emotions have enormous effects on our cognitive performance. Respectively, memory, attention, problem-solving, and decision-making are among the most significant cognitive abilities that are highly engaged with driving tasks. It is known that drivers eliciting anger or excessive happiness are biased in their risk estimation abilities, resulting in a lowered driving performance and safety preservation [40]. Intelligent in-car systems could use in-cabin emotional awareness to adapt their behavior empathically or provide

relief for negative emotions and improve driving safety and comfort level [41]. The comfort and safety here are not entirely independent attributes. For example, it is evident that driving with a low comfort level for an extended period results in increased fatigue, negatively impacting safety due to a loss of focus on the driving tasks. Dealing with this issue requires constant driver state monitoring, which is a considerable challenge in semi-automated driving systems that rely on the driver as a fallback when they cannot handle a situation by themselves. In such cases, the system needs to be aware of whether the driver can take back control in a *safe* period. In fully autonomous driving, comfort will become a decisive market factor from the manufacturer's perspective since the safety of the driving itself will be taken for granted by the customer. The well-being monitoring mechanism in the cabin environment is mainly camera-based and relies upon the driver's facial recordings. The desired emotion recognition can also be achieved by other modalities such as audio-based approaches, while the driver/passengers interact with the intelligent assistant application of the vehicle, or simply through the pop-up questions on the head unit, which requires the direct input of the occupants regarding emotional feeling. On the one hand, asking directly from the user will provide the most accurate answer about his/her emotional status. On the other hand, it is not a practical approach since it will impose too much distraction and frustration upon the subject. Respectively, although the camera-based solutions perform better for *continuous* emotion recognition, they lose their efficiency and functionality by sudden changes in the in-cabin environment. For example, by entering the vehicle into a tunnel which leads to a sudden change of the interior lightning or direct illumination into the camera from external sources, the camera may not be able to feed the expected input correctly to the consumer applications, and this will affect the quality of the context-based proactive services relying upon continues recognition of emotions as their feedback loop. Similar challenges apply to other approaches, such as audio-based solutions that suffer exclusively from insufficient data to predict based on continuously.

### 2.1.4 Situational Awareness

Newly introduced concepts of cloud-based services and Car-to-X (C2X) information sharing are now growing fast in the automotive domain. Their utilization in intelligent vehicles highly impacts situational awareness and context prediction architectures. Thanks to these technologies, new context can become available to cars without being obliged to sense it themselves. This can be the exchange of trajectories with other vehicles (as known as Car-to-Car) for the functions involved in automated driving or information about the current status of the environment around the vehicle, like traffic lights (as known as Car-to-X). A context prediction structure can better understand and predict more accurately due to the quickly changing environment by utilizing such information. In an idealistic scenario, the rich sensory infrastructure of the leading vehicle can provide a high-quality data set for the ego vehicle to be used in forming the knowledge base for the context, hence widening the horizons of pro-activeness for the integrated applications. However, communication remains as the Achilles heel for this

domain. As Wagner *et al.* [42] point out, ensuring the compatibility interfaces for C2X is also a significant challenge.

Despite these challenges, the utilization of cloud services and C2X communication can be considerably beneficial in modeling and predicting the future context for the applications of the vehicle. However, with all the benefits that cloud services, C2C and C2X, bring to the driving, there is a rising concern about ensuring the network's security. Avoiding malicious data injection into the intelligent components by an external attacker, especially on the services that rely on online learning and detecting the safety violations by the provided information (e. g. trajectories) either by the environment or other vehicles, must be addressed. Nonetheless, this feature brings many opportunities to context-aware and context prediction-based applications. It is also an essential element required for enabling the pro-activeness of intelligent applications integrated inside the vehicle.

## 2.2 Research Goals

This work takes a practical approach to investigate the critical factors that enable context prediction architectures in intelligent vehicle platforms and respectively provide pro-activeness for the vehicle functions with the help of context awareness. For this purpose, *safety assurance* and *emotional awareness* domains are chosen to be addressed accordingly as the primary determinants for context prediction architectures that have an indispensable role for human drivers in their development chain. These domains' design-time and run-time phases contain various applications, from driving modules to comfort services. The main designated research objectives and goals of this work are categorized as follow:

- **Safety assurance**: It is a challenging task to integrate the specific safety measures of the existing (as well as newly developed) standards and enforce the safety rules in the development phase of AI-based applications in vehicles. From AI practitioners' perspective, the development of the applications mainly focuses on increasing the performance factors, such as the accuracy of the functions. However, this viewpoint comes with the price of excluding the important inferring concepts like *safe operation* from the technical scope in most of the cases. Hence, it is crucial to provide a reliable use-case-independent platform to promote the enforcement of safety rules and standards into the AI development chain and facilitate the identification of safety violations for the experts during design time while maintaining the overall efficiency of the system **(Research Goal 1)**. Besides, by considering the black-box nature of most of the AI-based applications in intelligent vehicles with regard to the decision-making process, we aim to address the critical considerations in designing a reliable *safety monitor* for applications involved in driving task and respectively develop an application-independent adaptive solution to monitor the safe operation of the driving agents during the run-time **(Research Goal 2)**.

- **Emotional awareness**: The conventional applications that utilize the users' emotions usually rely on camera-based approaches. Despite the general capability of such approaches to predict the subject's emotional state with high accuracy, they are not a robust solution for the in-cabin environment. The emotional awareness itself is a multimodal challenge, and its related concerns can not be addressed from the perspective of one modality alone. The presence of occupants, especially the driver, incorporates multiple emotional indicators that are seen as contextual factors for comfort and safety applications. One of the main objectives of this work is to investigate the utilization of in-cabin behavior modalities as the emotional indicators and identify the impact of their integration into emotion recognition systems **(Research Goal 3)**. This set of objectives requires an in-depth study of the in-cabin behavioral factors and a precise evaluation of state-of-the-art machine learning-based methods on different modalities. The ultimate goal is to identify and address the challenges in developing a behavior-based multimodal emotion recognition structure that utilizes a reliable prediction pipeline and can be deployed inside the cabin environment **(Research Goal 4)**. Besides, an important application of the emotional indicators of the driver is the complementary part of feedback loops in decision-making for comfort and safety applications inside the cabin environment. It demonstrates their influential role in enabling context-based pro-activeness in desired services.

# 3 Background and Related Work

Numerous studies have been performed in the scientific domain regarding the development of intelligent cars and the involved technologies in the automotive sector, especially focusing on safety and emotional awareness. Nowadays, most intelligent cars are equipped with drowsiness and fatigue detection systems that use the driver's emotions as feedback and rely on the driver's emotional status and awareness level for driving tasks. In addition, various intuitive use cases exist that demonstrate the potential applications for safety and emotional awareness domains in the in-cabin environment. However, the development of such applications requires a considerable effort to address new issues and challenges presented by the integration of artificial intelligence. For example, an intelligent application such as path planning that engages with driving tasks largely increases the decision-making process's overall complexity. In order to clarify the perspectives and demonstrate the importance of the motives for this work, in the following sections, we take a gradual look into the related works in the fields of safety and emotional awareness, from definitions to the current state of research and development in the automotive domain.

## 3.1 Safety

With recent efforts to make vehicles more intelligent, artificial intelligence-based solutions using machine learning techniques have been absorbed by the ecosystem. They are seen as the golden ticket for the development of intelligent applications in future systems. These systems in the automotive domain are growing fast, speeding up the promising future of highly and fully automated driving while fostering new challenges regarding the safety assurance of the applications. The majority of the works in the safety domain are focused on enforcing the standards such as ISO2626/2 on implementation of conventional applications and verification of their functions. On the one hand, most integrated intelligent services dealing with the car's driving dynamics, such as *Advanced Driving Assistance Systems* (ADAS), rely on hard-coded solutions or merely employ basic AI-related techniques. On the other hand, the comfort domain and the applications, which are focused on the well-being of the occupants, have made considerable progress in utilizing AI and machine learning-based solutions to increase the quality of their services. These advancements are driven by the fact that their functionality does not require extensive verification, and they are not obliged to follow strict safety standardization procedures.

### 3.1.1 Design-time Vs. Run-time

In recent years, the safety aspect of artificial intelligence has been placed at the center of attention for researchers in different domains, especially the applications deployed by machine learning-based methods such as neural networks and deep learning methods [43,44,45,46,47]. This issue has been investigated mainly from two different perspectives of (i) *Run-time* [48] and (ii) *Design-time* [49]. However, there is still a considerable lack of concrete solutions to address the related challenges practically. The very nature of this concern is due to the diverse range of applications and, as stated before, dynamically changing contexts for them. However, one may narrow down the problem to some specific common grounds that would considerably target the status quo, expose the most significant concerns and widen the scope of the potential solutions. Neural networks are the core of machine learning and most of the respective developments in the artificial intelligence domain. However, utilizing them is not always straightforward in the development phases. For example, the automotive domain is based on concretely defined standards for maintaining the functional safety of the developed applications. The main challenges associated with applying traditional safety assurance methodologies to neural networks, as was explained thoroughly by Cheng *et al.* [50], can be categorized as follow:

- **Implicit specification:** formal verification and validation (V&V) methods (as suggested in ISO 26262 V model) emphasize ensuring that the functional requirements specified at the design time of the system are met. However, neural network-based systems depend solely on the training data for inferring the model's specifications and do not depend on any definitive list of requirements, which can be problematic while applying traditional V&V methods. Furthermore, this issue is

highly use-case-dependent and can become challenging due to various applications deployed in a vehicle.

- **Black-Box structure:** unlike traditional software development approaches, the control flow is not explicitly hardcoded in neural networks. It is the reason for referring to them as black-box structures. Traditional white-box testing techniques such as code and decision coverage cannot be directly applied to neural networks; thus, there is a need to construct paradigms for adaptive software systems by the progress of this field.

To enlighten the importance of the challenges mentioned above and investigate the existing solutions, we can examine the issue from two angles of the training phase (as known as design-time), as the approaches that are exclusively used during the training phase of the neural networks, and the operational phase (as known as run-time), as the ones that are used in the execution environment of the neural networks to ensure proper functioning. Certainly, the training phase bears more importance and has the highest level of impact in this regard; therefore, more studies have been carried out with this focus. We can divide the evaluated solutions accordingly, under the following categories:

- **Train/validation/test split**: This method is the most typical approach to ensure that the developed adaptive system works satisfactorily for a given set of inputs. The method involves splitting the available data to obtain three subsets in a way that the largest of the sets is used exclusively for training. Of the remaining two sets, one is used for fine-tuning the network's hyper-parameters, and the last one is used to test the working neural network to study how well it reacts to previously unseen data points. Though this method helps to verify the overall functioning of the neural network, it is not comprehensive enough to be considered as an ultimate guarantee to ensure the enforcement of safety standards in high criticality systems [37].

- **Automated test data generation**: The lack of trust in the train-validation-test split method originates from the fact that one is left with very few data samples to test against, wherein the chances are that cases of high interest might even get missed in the testing phase. An approach to overcome this problem is to use test data generation tools to generate synthetic data points to test the trained neural networks. Tools such as *Automated Test Trajectory Generation* (ATTG) [51] and the more recent approach of generating scenes that an autonomous vehicle (AV) might encounter using ontologies [52] also are beneficial in this regard. This approach can help the V&V procedure for neural networks by unveiling missing knowledge in fixed and increasing confidence in the working of adaptive neural networks [37].

- **Formal methods**: Formal verification refers to the use of mathematical specifications in order to model and analyze a system and its behavior [53]. Though these methods work well with traditional software development processes, they have not

shown much success in adaptive software systems. It is due to challenges in modeling the non-deterministic nature of the environment, difficulty in establishing a formal specification set to encode the desired and undesired behavior of the system, and the need to account for adaptive behavior of the system [54]. Formal verification techniques for neural networks deal with proving convergence and stability of the system [55], by using methods such as Lyapunov analysis [56].

- **Rule extraction**: Rules are viewed as a descriptive representation of the inner workings of a neural network [57]. Rule extraction algorithms, such as KT [58], Validity Interval Analysis (VIA) [59], and DeepRed [60], can be used to model the knowledge that a neural network has acquired during the training phase. These rules can be expressed as easy-to-understand *if-then* statements, which itself can be manually verified due to its human-readable format or with a third-party model checker. This method can be helpful in establishing trust, as it augments the explainability of the system [61]. It also aids requirements traceability, as one can verify whether the rules depict functional requirements specified for the system. They can also help examine the system's various functional modes and ensure that a safe operation mode is induced by specific inputs while respecting the expected safety limits. Though this method brings enormous advantages, it is more applicable for offline learning systems, wherein the V&V practitioner can extract rules from the network after complete training.

Respectively, the solutions related to the operational phase, namely *monitoring techniques*, seem to be more concrete, reliable, and practical in real-life scenarios. This group of solutions involves utilizing one or more *monitors* working as an oracle to ensure the continued and proper functioning of the neural network over time [62]. The goal is to ensure that the adaptation dynamics do not cause the network to diverge, triggering unpredictable behavior. Data sniffing is an example of the preceding technique, which studies the data entering and exiting a neural network [39]. If a particular input could pose negative results, the monitors generate an alert and can flag down the data, thereby not allowing it to enter the system. This method is advantageous in cases where outliers could degrade the system's functioning. In the following, we will take a closer look into the monitoring concept.

## 3.2 Safety Monitors

Deep learning pipelines for autonomous vehicles can get quite complex, making it more likely for errors to creep in. It is thus imperative to have a system that can monitor the run-time performance of various system modules.

As depicted in Figure 3.1, a safety monitor can be thought of as an oracle that envelopes the system of interest [63]. It observes the run-time environment and behavior of the system to ensure that the functions do not deviate from previously agreed-upon safe states. If a violation is detected, the safety monitor can trigger the appropriate intervention, which could be preventive or corrective based on the design and objectives

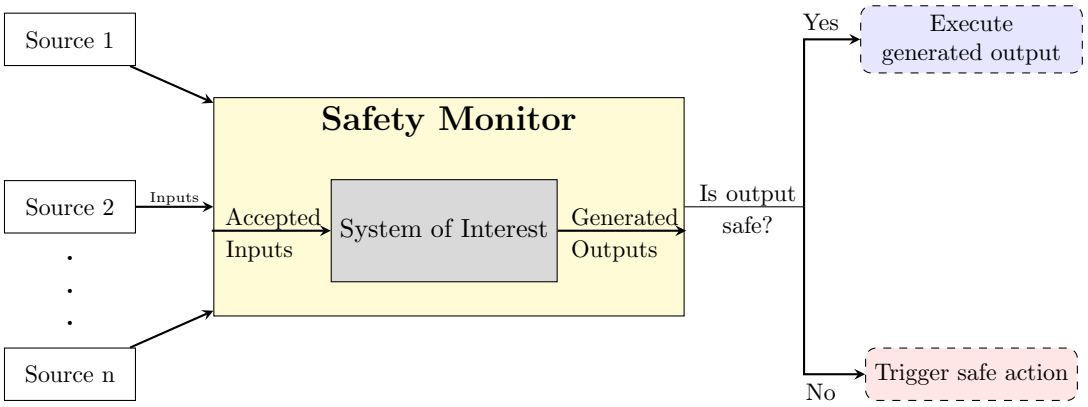



**Figure 3.1:** The general concept of a *safety monitor*

of the system. However, a prerequisite for such a setup is that the monitor trusts the sensors and the actuators in properly providing the expected feed under observation.

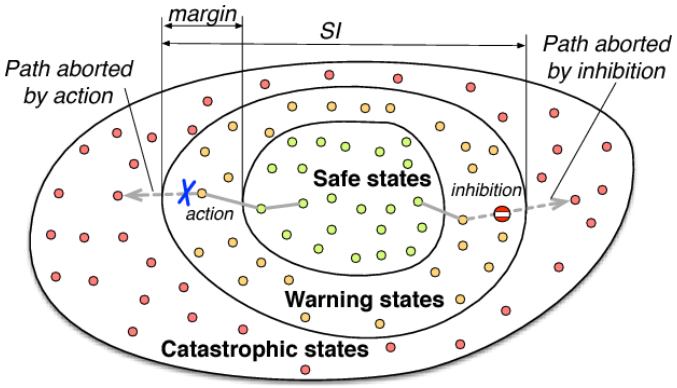



**Figure 3.2:** The *safe*, *warning* and *catastrophic* states fo an autonomous system [1]

Safety monitors usually differentiate between three primary states that the system of interest can be placed in during the operations. The *catastrophic* state is one where the damage has already been done and from which the system cannot be recovered immediately. The remaining two non-catastrophic states are the *safe* state, where the system behaves as expected, without any constraints, and the *warning* state, wherein the system is close to being involved in a catastrophe; hence an intervention is applied accordingly [1]. As depicted in Figure 3.2, for a system to transit from a safe to a catastrophic state, it must always pass through the warning state; thus, the warning state can be thought of the margin, wherein if the autonomous system executes the correct action, it still has the possibility of being brought back to the safe state. For this purpose, a safety strategy must be defined thoroughly, incorporating a set of safety rules that determine how the system responds in different warning states. A popular framework for setting up such monitors for autonomous systems is called the *safety monitoring*

*framework*, SMOF [1]. This framework defines a five-step process for outlining safety monitors for a target system, as listed below:

1. **HAZOP-UML Hazard Analysis:** utilizes UML use case and sequence diagrams to model the autonomous system. The diagrams are then used to detect operational hazards by assessing potential deviations in the run-time behavior of the system. Causes, consequences, and severity are then determined for the potential hazards. The output of this step is a list of conditions which, when they are violated, the system is forced into a catastrophic state, i. e. safety invariants, as described in natural language.

2. **Safety Invariant Formalization:** The safety invariants from the previous step are converted to a form that can be mathematically represented. For this purpose, the safety invariants are mapped to variables that the monitor can observe, such that the conditions are expressed as predicates on these variables. The framework only focuses on variables that can be compared to fixed threshold values to allow formal verification.

3. **Safety Invariant Modeling:** The SMOF modeling template is used to build state-based models to represent the formalized conditions generated in the previous step. Each safety invariant is modeled separately to ensure that the models can be easily validated. Furthermore, the non-catastrophic states identified in the state-based models are partitioned into safe and warning states by splitting the intervals that lie within the previously determined thresholds.

4. **Strategy Synthesis:** Considering the state diagrams with safe, warning, and catastrophic states are determined per safety invariant, interventions are established. This helps to ensure the system is always in (or can be easily reverted to) a safe state.

5. **Consistency Analysis and Implementation:** By the safety invariants being modeled and handled separately, the chances of developing conflicting safety strategies for a set of conditions are extremely high. Thus, this step ensures global consistency of the established safety strategies, followed by implementing the safety strategies onto real-time safety monitors.

### 3.2.1 Uncertainty

As stated before, the integration of machine learning methods in automotive applications provides enormous opportunities to enhance and automate the existing applications and increase the overall performance by utilizing AI-based solutions to the end-user of the vehicle. Most of the in-cabin applications built on top of machine learning-based methods are related to the occupant's comfort and do not require concrete safety measures. Nevertheless, the applications involved in driving tasks or acting as a bridge between the user and the vehicle dynamics are required to be formally authorized as *safe* before being deployed. There are various methods to achieve an acceptable level

of *safety* for the developed applications in the automotive domain. The model-based formal verification is the widely accepted solution in this regard [64]. However, in the presence of machine learning-based models, formal verification becomes a very tough and challenging task due to the arising complexities directly from the models and their different nature with regard to mathematical foundations [65]. For instance, uncertainty in data and machine learning methods is a pivotal point to investigate one of the main origins of safety-related concerns, especially when it comes to automotive and safety assurance mechanisms in its applications. In order to clarify the issues which are either directly originated or got impacted from uncertainty in intelligent cars and automated driving applications, consider an ordinary maneuver planning system and the respective safety-critical situations that are intuitively explained in the following scenarios:

**Case 1.** The system has been trained and tested on the data from roads in a country with well-behaved traffic patterns and law enforcement, but for operation is deployed to be driven on roads in another country with chaotic driving conditions. Another similar case is when the vehicle has been trained and tested on roads with four wide lane driving options, but during the operation is placed on a 2-way narrow lane road. In such situations, the outputs of the intelligent vehicle cannot be relied upon, as there is no guarantee that the system would behave as it is expected.

**Case 2.** The vehicle which employs this system decides to overtake another vehicle in front of it. The country's driving rules state that one must overtake only from one side (either from left or right). Though this is imbibed in humans as the user while learning to drive, there is no guarantee that the system in an autonomous vehicle has indeed learned this fundamental rule and always follows it.

**Case 3.** The vehicle needs to execute a lane change operation to reach its goal state. However, there happens to be a vehicle on the left side in such an alignment that increases the possibility of a collision. Since standard deep learning techniques generate only rigid classifications as output, there is a chance that such low probability gets ignored and leads to costly collisions/accidents.

**Case 4.** Humans are designed to be innately optimistic, which might be reflected in neural networks' training data. Such networks in autonomous vehicles are usually trained to exhibit the positive outputs we expect from them. Those benefits could be reaped by getting trained to generate both positive and negative outputs. However, excluding the negative cases has become a common practice among the developers and researchers in the race of achieving *promising results* first, which puts the reliability of the outcomes at risk.

According to [66], the *uncertainty* in machine learning algorithms from a high-level perspective can be categorized into two types:

- *aleatoric* or data dependent; where the noise in the data is captured by the model, resulting in the ambiguity of training input.

- *epistemic* or model-dependent; which is seen as a measure of familiarity, representing the model's ambiguity when dealing with operational inputs.

More precisely, the significant causes of concern while dealing with machine learning-based applications are as follows:

**Incompleteness of Training Data** Traditional software systems are developed with a predefined set of functional requirements. However, in neural networks, and more generally in machine learning algorithms, the system's functional requirements are implicitly encoded in the data that it is trained on, expecting that the training data represents the operational environment. The setback, however, is that training data is by definition incomplete [67], as it represents a subset of all possible inputs that the system could encounter during operation. Insufficiency thus arises when the operational environment is not wholly represented in the training set. In autonomous vehicles, critical and ambiguous conditions usually tend to be problematic, where the vehicle is expected to act predictably. Due to their extremely rare or hazardous nature, such situations tend to be underrepresented in the training set [44, 47].

**Distributional Shift** In the case of an autonomous vehicle, the operational environment is highly unpredictable [44] as it is constantly changing in response to the actors within the system. Therefore, even with an excellent and near-perfect training set, the operational inputs may not be similar to the training set. In other words, there could be a considerable shift in the distribution of operational data compared to the original training data, hence resulting in the unpredictive behavior of the system.

**Differences Between Training and Operational Environments** Subtle changes in the operational environment can lead to a state of unpredictable behavior in neural networks [44]. A fine-tuned neural network for a specific setting provides no guarantee of functioning in the same way when the settings are changed at the run-time.

**Uncertainty of Prediction** Every neural network has an error rate associated with it [67], the training phase aims to reduce this error rate as much as possible. In the operational environment, this error rate can be interpreted as the uncertainty associated with the output produced by the model. This uncertainty can convey helpful information on how well the system models the environment; however, it is not accounted for very much in today's cyber-physical systems [66]. Standard deep learning models use point estimates of the predictions rendering them incapable of dealing well with uncertainty [68]. On the other hand, Bayesian deep learning infers distributions over the model parameters, allowing one to adopt a more probabilistic approach to modeling the situation [69, 70]. In addition to generating uncertainty estimates, Bayesian deep learning also helps to reduce over-fitting. However, these models have not yet become standard due to difficulties in optimizing the objective function. In this regard, the MC-Dropout technique [71] helps to avoid these challenges while giving the added benefit of making only minimal changes to the standard deep learning models. MC-Dropout works

by applying dropout at not just training time but also during the making of predictions in test time [69, 71]. Dropout causes a certain number of hidden units to be randomly *switched off*, essentially has the effect of being a different model every time a data point is passed through for an inference. MC-Dropout thus builds on this idea by making T stochastic forward passes through the model and then calculating the mean and variance of the T passes, as equivalent to applying an ensemble of neural network models, which approximate a Bayesian function.

## 3.3 Definition of Emotions

The earliest study on emotions can be traced back to 1872 when Charles Darwin wrote his book "Expression of the Emotions". He tried to challenge the reason for the existence of emotions in humans and other animals and the main motives to express them. Afterward, there have been numerous studies on emotions in different scientific domains. However, the only evident fact among them is the lack of consensus on the definition of emotions itself. Different methodologies and models in numerous domains lead to different definitions of emotions. Most research domains agree that emotions are more personal experiences, making it tough to develop a general definition that suits all the use cases. Moreover, emotions by nature are multimodal; therefore, unimodal proposals can not be representative enough. It is true to say that the statement from Augustine of Hippo, in his book *Confessions XI, written between the 4th and 5th centuries,* is the most valid explanation in this regard: *"What then is an emotion? If no one asks me, I know what it is. If I wish to explain it to him who asks, I do not know".*

Emotions can be seen as modes of functioning, shaped by natural selection, that coordinate physiological, cognitive, motivational, behavioral, and subjective responses in patterns that increase the ability to meet the adaptive challenges of situations [72]. In dealing with Affective phenomena, we must address some definitions first. If we consider personal experience and perspective, we may refer to **emotions** as *the intense feeling from one's circumstances, mood, or interactions with others.* We can also refer to the **mood** as a temporary state of mind or feeling. Respectively the **feeling** itself would be an emotional state or reaction to external stimulus. Feelings generate emotions, and emotions form feelings in parallel. That is why we name the experience of feeling as **affect**. However, when we examine the scientific perspective, we can come up with slightly a bit different yet more concrete set of definitions as follow:

- **Emotions**: multi-situated body mechanism to give semantic meaning and coordination to internal and external data in order to create action states,
- **Mood**: pervasive emotion over a more extended period which is user/personality-dependent,
- **Feeling**: the self-perception of an emotional event,
- **Affect**: outward, representative, physical signs of emotions.

The emotional process itself includes certain phases and operations that are required to be clarified in advance. Let us consider a driver in a typical driving car in level 3 of autonomy as an intuitive case study. The **appraisal** refers to the estimation of the driver from the situation. The driver continuously evaluates its danger and safety level by changing the environments and situations resulting in different driving contexts.

The natural consequence of any appraisal that could be either physical or behavioral (in general, known as subjectivity) is called **arousal**. It can have **representations** such as gestures, tone, or facial expressions. When the driver detects a critical situation,

he conscientiously determines the severity of it. This measure can be spread along with positive, negative, and [neutral] levels, called **valence**. In other words, valence represents the pleasantness of the experience. Valence in emotions correlates to reaction; hence a proper action needs to be taken accordingly (e. g. breaking in case of being in a hazardous situation). This is called **elicitation**. By observing multiple situations, evaluating the severity, and taking the respected actions, the driver learns the general goal to pursue (e. g., "from now on, try to avoid ending up in such situations"). We call this **drive**; the lesson learned by experience.

Several categories can classify emotions since they run at different levels of complexity and performance. Philosophically the categorization of emotions can be outlined in the paradigm of pain-happiness. The eastern perspective believes that *"humans want to be happy"*, while the western perspective approaches this issue as *"humans want to avoid pain"*; However, eventually, all categorizations can be mapped to one of the following groups:

- **Primary groups**: high-level categorization into positive, negative, and neutral,
- **Basic**: as listed in table 3.1,
- **Secondary (& tertiary)**: the objective ones with no consensus on the origin of them. A brief number of them are listed in table 3.2,

## 3.4 Emotions in Neuroscience

Various distributed structures in the brain are identified to be correlated with emotions, such as the amygdala [73], hypothalamus [74], and insula cortex [75]. Historically, studies on cognitive processes, more specifically on attention, memory, and perception, have excluded the role of emotion in modulating cognition [76]. However, the distinction between cognitive and affective processes is increasingly blurred as the processes largely overlap in neural mechanisms and structures [77]. Due to this inter-dependency, emotion recognition is crucial in relaying information about the user's stress or comfort level and cognitive faculties. With this information, we can infer that emotional state is vital for maintaining adaptive comfort of the occupants in a vehicle and is crucial in various cognitive processes like decision-making. While behavioral characteristics can be elicited in response to emotional states (e. g., facial expressions or gestures), they can also be measured more directly. This process typically involves physiological measures, such as galvanic skin response, heart rate, blood volume pulse, facial electromyography, and electroencephalogram.

In the automotive domain, to have a comprehensive picture of a driver's emotional state, capturing physiological or neural data has a significant impact on inferring the brain's current emotional response(s). For instance, it has been demonstrated that by utilizing EEG signals in a car, driver emotion can be successfully modeled and predicted [78, 79]. Furthermore, it has been evidenced by Shimojo *et al.* that the human brain uses multimodal information in order to infer decision-making processes [80].

| Theorist | Basic Emotions |
|---|---|
| Plutchik | Acceptance, anger, anticipation, disgust, joy, fear, sadness, surprise |
| Arnold | Anger, aversion, courage, dejection, desire, despair, fear, hate, hope, love, sadness |
| Ekman | Anger, disgust, fear, joy, sadness, surprise, wonder, sorrow |
| Frijda | Desire, happiness, interest, surprise, wonder, sorrow |
| Gray | Rage and terror, anexiety, joy |
| Izard | Anger, contempt, disgust, distress, fear, guilt, interest, joy, shame, surprise |
| James | Fear, grief, love, rage |
| McDougall | Anger, disgust, elation, fear, subjection, tender-emotion, wonder |
| Mowrer | Pain, pleasure |
| Oatley | Anger, disgust, anxiety, happiness, sadness |
| Panksepp | Expectancy, fear, rage, panic |
| Tomkins | Anger, interest, contempt, disgust, distress, fear, joy, shame, surprise |
| Watson | Fear, love rage |
| Weiner | Happiness, sadness |

**Table 3.1:** Grouping of basic emotions [11]

## 3.5 Emotions in Automotive

In the automotive context, emotional stimuli and the current emotional state of the driver can significantly influence the cognitive aspects [81,82]. For instance, as shown by Zhang *et al.* [81], Assari *et al.* [83], and the national sleep foundation [84], a bored driver may be more prone to drowsiness and consequently lose control during driving. This will result in slower reaction speeds that can lead to an unexpected incident. Emotions also can influence a driver in both positive and negative ways, as described by Lotz *et al.* [85]. For example, anger tends to be associated with riskier driving behavior, such as aggressive driving. Drivers in this state are more prone to be involved in an accident, as shown by Zhang *et al.* [81] and Lu *et al.* [86]. Furthermore, emotional stimuli in the driving environment can also increase driver distractions [82]. Fear is another emotion that can impact driving behavior. This impact can be positive since a fearful driver can perceive a situation as a potential risk, enforcing defensive driving. However, fear can also induce anxiety caused by either the driving environment or the act of driving itself, especially by the increased level of autonomy [86]. Therefore, identifying the emotions and addressing the emotional states is essential to determine the driver's stress level and

| **Secondary** | **Tertiary** |
|---|---|
| Suffering | Agony, suffering, hurt, anguish |
| Sadness | Depression, despair, hopelessness, gloom, glumness, sadness, unhappiness, grief, sorrow, woe, misery, melancholy |
| Disappointment | Dismay, disappointment, displeasure |
| Shame | Guilt, shame, regret, remorse |
| Neglect | Alienation, isolation, neglect, loneliness, rejection, homesickness, defeat, dejection, insecurity, embarrassment, humiliation, insult |
| Sympathy | Pity, sympathy |
| Horror | Alarm, shock, fear, fright, horror, terror, panic, hysteria, mortification |
| Nervousness | Anxiety, nervousness, tenseness, uneasiness, apprehension, worry, distress, dread |

**Table 3.2:** Secondary and tertiary emotions [12]

current driving competency. It is also notable for differentiating the importance of the emotions in higher levels of autonomy by shifting the driver's role from the human user to the vehicle. In order to clarify this, we can divide the current status of the research and developments into two main categories as follow:

### 3.5.1 Conditional Automation

The main focus of conditional driving scenarios is on human-machine handover cases. In situations that the intelligent vehicle can not continue to drive in its autonomous mode, it will issue a TOR that informs the driver to regain control of the vehicle and stop with all non-driving-related-tasks (NDRT) that was the focus of attention during the autonomous driving period [87]. According to *takeover time* and *takeover quality*, the recent studies on takeover situations reveal that emotions with positive valence and high arousal achieve better takeover performance [88].

The outcomes demonstrate that tracking and understanding the emotional state can significantly impact safety during the takeover. It is challenging, especially when the vehicle gets in charge of maintaining the driver's emotional state at all times and therefore selects appropriate measures in advance to prepare the driver for a safe takeover without triggering the risk of a crash or unsafe driving. The emotional awareness of the in-cabin interfaces can also be beneficial in the presence of negative emotional states, in simplifying the information and executing suitable actions to ease the tension to prevent unsafe driving behaviors (e. g. by providing helpful information to comfort and calm down the passengers) [89]. Meshram *et al.* [90] relies on an old-fashioned visual approach and propose using an ECU camera to constantly detect the emotional status (i. e. calmness,

happiness, sadness, or anger) of the driver and use it to evaluate the readiness of the driver for a manual takeover. The autonomous vehicle enables manual driving in neutral emotional states (known as calmness) and positive valence (e. g. happiness). Suppose negative emotions like sadness and anger are detected. In that case, the vehicle stays in automation mode with reduced driving speed, and the driver can not engage in manual mode to prevent the risk of traffic violations.

### 3.5.2 High/Full Automation

Egger *et al.* [91] demand that *"computers should respond to their users humanely"* in order to build a trustworthy relationship between humans and the autonomous vehicle. Other studies also revealed that humans need emotions and emotional feedback to interact with machines more comfortably. Thus, the autonomous vehicle must react depending exclusively on the individual passenger sitting inside the vehicle. Paiva *et al.* in Figure 3.3 present an *affective loop of emotional robots* which illustrates the embedding of emotional awareness in systems of autonomous robots in a cycle of three phases: emotion detection, emotional behavior generation, and emotion elicitation. As a result, human-machine relationships appear closer when machines can examine and use information about emotional awareness. Hence, incorporating features to enable emotional awareness will considerably impact user acceptance of vehicle applications by achieving a personal relationship between humans and machines.

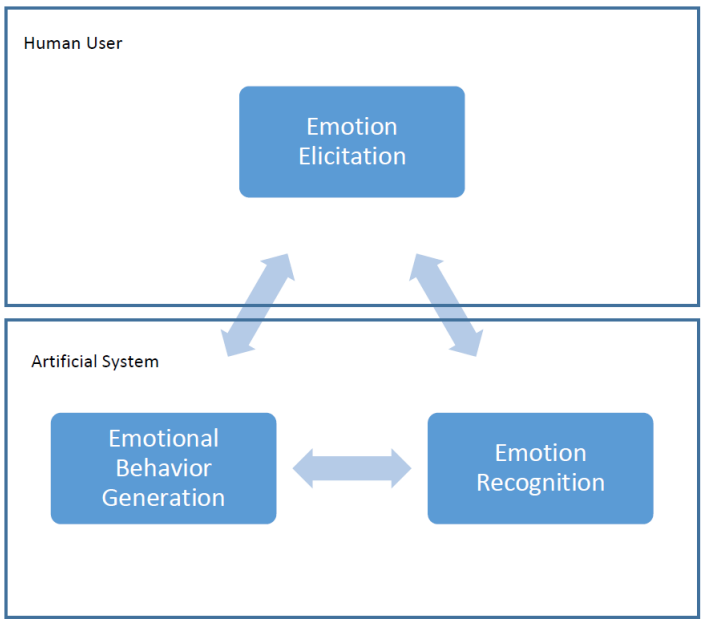



**Figure 3.3:** Affective loop of emotional robots [2]

Sini *et al.* [92] also highlight the importance of building a trustful relationship between humans and self-driving vehicles. The authors stress the integration of information about

the emotional state of passengers into the calibration process of driving styles. In their work, emotional awareness is utilized in driving features of the vehicle in order to make the decisions taken by the vehicle closer to the passenger's expectations as much as possible. In their subsequent work [93], they indicate the influence of emotional awareness on safety-relevant applications like the driving style in three individual scenarios:

- **Negative feelings like sadness and scariness**: caring driving style is adopted where the speed is reduced, and curves create less lateral accelerations,
- **Neutral feelings like calmness**: typical driving style is adopted,
- **Positive feelings like happiness and pleasure**: sportive driving style is adopted with steeper acceleration and breaking,

A survey conducted by Braun *et al.* [94] outlines a variety of future demands and requirements about the application of emotional awareness in autonomous driving. As a result, drivers propose using emotional awareness measures to evaluate the best fitting driving style of all types of passengers (e. g. mother, grandmother, children) inside the vehicle so that everyone feels safe and comfortable. Maurer *et al.* also reveal similar results in a conducted survey at [95]. Besides the general public argument on increasing road safety by introducing autonomous vehicles, the participants highlight benefits like stress reduction and more convenience by utilizing autonomous vehicles, which can be achieved when an autonomous vehicle senses emotions and reacts to the present emotional state of the users accordingly. In this regard, one of the early works is represented at [96], where the authors investigated the impact of the empathetic voice assistants deployed inside the cabin on safety. The result demonstrates that drivers with a voice assistant that reflects their emotional state had, on average, less than half as many accidents as drivers without matching voice assistants. By matching the voice assistant, the connection level with the driver is increased, a characteristic that is usually taken over by the co-driver [41].

In the higher two levels of autonomy, the intelligent vehicle takes over (almost) complete control, and the driver becomes an ordinary passenger by delegating the whole driving task to the car. Therefore, the vehicle must show a high level of transparency on its actions and why it is performing them with an additional constraint to act humanly to maintain trust and acceptance. Whereas emotional awareness in level 3 has a beneficial role for safety applications, the integration of emotional awareness features is seen as a mandatory step to ensure the introduction of level 4 and 5 vehicles [89, 91]. Human trust and human acceptance are seen as crucial concerns when thinking about highly and fully autonomous vehicles. They only are overcome if the autonomous vehicle can build a decent human-machine relation that considers all essential characteristics of a human-human relation. Therefore, as discussed before in Section 2.1.2, considerable integration of emotional awareness in all safety-relevant applications are inevitable so that the vehicle can choose the best fitting driving style depending on the state of the passengers, to make humans feel safe and comfortable and to establish trust accordingly [91, 95, 97].

## 3.6 Emotion Recognition

Humans experience and process emotions differently. There exist two main theories providing a thorough overview on this phenomenon, namely *categorical* and the *continuous* approach. In the categorical approach, each emotion is defined using a qualitative measure, where each measure itself is comprised of a specific emotional category like *happiness*. Paul Ekman can be seen as a pioneer in this perspective, and his development of facial expressions that correlate to specific emotional categories represents one of the preliminary works in this regard [98]. Ekman's continuous model that describes emotions in a feature vector is generally based on valence and arousal measures. The origin of this theory goes back to a psychologist named James Russell, and it is famous as the *circumplex model* [99]. In this model, emotions may have a range of intensities, and different intensity levels may lead to different emotions collectively. There exist also hybrid approaches that combine the dimensionality and categorical information of both fundamental theories. As represented in Figure 3.4 in Plutchnik's work [3], there are essentially categorical emotions that have various intensities where the farther out from the center of the wheel, the more complex the emotion becomes.

The recognition of emotions has wide implications in different applications and recently got the attention of many researchers in the field of automotive, especially for the applications of driver fatigue detection [100, 101, 102, 103], human-car interaction [100, 104], and respectively the highly and fully autonomous driving scenarios [105, 106]. According to the *7-38-55* rule, 93% of human communication is performed through nonverbal means, including *facial expressions*, *body language*, and *voice tone* [107]. Therefore, a system that automatically analyzes the emotions of humans should exclusively focus on these non-verbal channels. This research field is called *affective computing*, an emerging research field in enabling intelligent systems to recognize human emotions. The main challenges in automated camera-based affect recognition are *head-pose* variations, *illumination variations*, *registration errors*, *occlusions*, *identity bias* and in general, *subject-independent affect recognition* [108]. The most common and effective approach in the field of affective classification is the utilization of multimodal methods. The general aim of multimodal fusion is to increase the accuracy and reliability of the estimates while maintaining the robustness of the systems. Based on empirical studies and statistical measures, multimodal systems are consistently more accurate than their unimodal counterparts, with an average improvement of 9.83% (median of 6.60%) [109,110]. However, the fusion of multiple modalities into one single final output is a challenging task. The right fusion method highly depends on the underlying data types and streams. Common fusion techniques in the field of affective computing are *feature-level fusion*, *kernel-based fusion*, *model-level fusion*, *score-level fusion*, *decision-level fusion*, and *hybrid* approaches [109,110,111,112]. The most common fusion techniques are feature-level fusion and decision-level fusion. In feature-level fusion, the data from separate modalities are first aggregated and then used as a single input into one model. Each modality has its trained model in decision-level fusion, and the predictions are then combined to a single output as the identified emotional state.

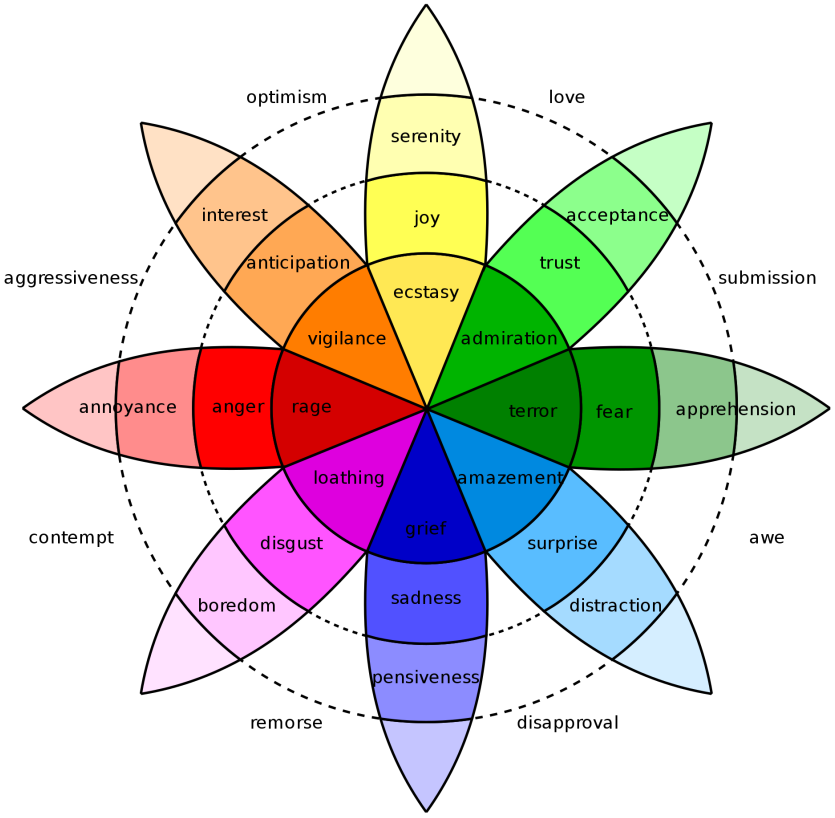



**Figure 3.4:** Plutchnik model of emotions [3]

## 3.7 Emotional Measures

Emotions can be defined in various ways; hence, various measures exist to detect and identify them. Ideally, for a perfect measurement of emotions, we would need to measure the changes of appraisal processes during the central nervous system processing, the responses of neuroendocrine, autonomic, and somatic nervous systems. It also includes action tendencies resulting from the appraisal processes, facial, vocal, and body expressions, and the subjectively experienced feeling [113]. However, using all these factors in real-life applications is not practical and plausible due to a considerably massive amount of generated data, intrusive nature and increased complexity of medical measures listed, and the users' unwillingness to constantly provide feedback for the applications. This is notable, especially in an in-cabin environment where only a subset of these measures can be utilized due to resource limitations and safety concerns.

### 3.7.1 Human Observation

An intuitive way of detecting emotions is to entrust human judgment. This method is used mainly for creating and verifying the emotion labels in an affect recognition dataset. A standard procedure is to record the representations of the participants during a study. Afterward, participants watch the recorded video and annotate their emotional states accordingly in each different frame. Therefore, psychologists have considered the questionnaire systems extensively as the primary tool of indicating the affect and emotional state. An established measure is the positive and negative affect schedule, where affects can be rated on a 1-5 scale [114]. Since words for describing emotions are used in different contexts by each individual, ratings with pictograms such as the self-assessment manikin (SAM) are developed [115]. This approach avoids predetermined emotion categories but is hard to transfer to a computational affect recognition system.

One disadvantage of using self-reports is the distortion of the data because of the potential risks. For example, participants may be provided with answer possibilities that they would not have chosen when freely describing their emotions, or the answer set does not include the matching emotion description that they thought of; hence the participants are forced to use a similar alternative, a residual category, or omit the question [113,116]. Another drawback is that respondents may be unable to identify their emotional state or refrain from giving socially uncommon answers that would lower their social image [117], and the use of language distracts the participants in processing their feelings and the emotional effect diminishes accordingly [118]. However, Barrett [119] still holds the opinion that verbal report is the best method for efficiently measuring emotions at the moment.

A second method involves the behavioral observation and labeling of emotions by others, conventionally done by experts specialized in emotion psychology. In order to ensure consistency in this method, the same instance is analyzed by multiple scientists. They rely on the body and facial movements and vocal uttering. This way, the disadvantages of the self-report can be overcome since the experts often detect emotional patterns that individuals may not recognize or demonstrate directly [117]. Nevertheless, scientists mostly label with a limited number of emotion categories predefined in advance [113] which may distort the truth of the experienced emotion. Human observation is necessary to collect training and test data with the outlook on an automatic affect recognition system. Nevertheless, it is not applicable for a real-time automated emotion recognition system inside a vehicle.

Humans often rely on facial expressions to identify the emotional state of others; therefore, we continue with how this way of detection is transferable to a machine as well.

### 3.7.2 Facial Expressions

Currently, the most frequently used measure for emotion recognition in the automotive domain is facial expressions. A facial expression is a contraction of specific facial muscles and usually lasts between 250ms and five seconds [120]. An automatic facial

expression-based recognition system typically consists of a pipeline of face detection and face registration, then feature extraction, and finally recognition of expression [121].

Initially, the face is recorded with a camera; afterward, it is algorithmically detected and tracked, often using an algorithm like Viola-Jones [122], which is mainly chosen due to its fast performance, based on a cascade of weak classifiers [121]. The main facial features are localized on the tracked face, and optionally, more specific landmarks are detected. In detail, eyes are the most noteworthy facial feature showing affective states of the individuals and focus of attention [123]. For the feature extraction, either pre-designed or learned features are used. Humans specify Pre-designed features beforehand to extract relevant information. Then, learned features are picked up by a machine from a set of training data [121]. In the following Section 5.5.1, we represent a similar pipeline in more detail, designed and developed exclusively for our multimodal recognition architecture. A widely used system for describing facial expressions is the *Facial Action Coding System* (FACS), which was originally developed by Ekman and Friesen [124] and subdivided facial expressions into one or more action units (AU). One AU has a name and a number code for a facial muscular action, head movement, or behavior, e. g. *1 - inner brow raiser*, *51 - head turn left*. Depending on the emotion, different action units are activated, e. g. a distracted state can be detected with the action units *Lip Tightener* (AU23), *Jaw Drop* (AU26), *Lip Suck* (AU28), and *Blink* (AU45) [123]. Notably, FACS records all visually distinguishable facial movements; however, those action units postulated as *emotion-relevant* are usually encoded as EFACS/EMFACS.

By utilizing action units as features and the advancements in machine learning methods, emotions have been recognized by a high accuracy with the help of K-nearest neighbor (KNN), Bayesian networks, hidden Markov models (HMM), and artificial neural networks (ANN) [111]. From the applicability point of view, vision-based models are well-suited for the in-cabin environment. Since drivers naturally face forward 95% of the time, keeping a frontal head position and cameras can easily be installed in the dashboard area. Nevertheless, the detection performs notably worse in dark or visually noisy surroundings (e. g. significant head movement, driver wearing sunglasses or a hat). Furthermore, change in the environmental conditions (e. g., in-cabin lightening) may have a highly negative impact on the prediction outcome of such models along with the subject-related issues (e. g. ethnicity). In order to increase the robustness of the camera-based system that captures different head poses, multiple cameras pointing toward the driver's face can be used [123]. For sparsely lighted situations, an infrared/thermal camera can detect the basic features, as seen in the eye detection model by [125] and [126]. Therefore, achieving a more robust solution in this modality requires an extensive utilization of extra hardware.

### 3.7.3 Body Movements and Behavior

Even though facial expressions are the primary indicator for sentiments, the rest of the body is not different from representing emotional status. Previously, it had been assumed that body behavior such as gestures or postures only shows the intensity of emotions rather than specific emotional status [127]. In [128], Dael *et al.* show that the body

obtains a diverse repertoire of emotional body postures, movements, and gestures. They conclude that all emotions are differentiable by body behavior, even subtle emotions. Since not many studies focus on the whole body's movement, there is a lack of corpora that includes body posture, movement, or gestures in this regard. For in-cabin environments, the movement possibilities are limited due to space and posture. However, the head, shoulders, and arms can be moved freely. In [129], Mota *et al.* use body postures of children sitting on a chair to detect their emotional state with a considerable accuracy of 87.6%. They use a chair with pressure sensors, which could be incorporated into a car seat as well.

### 3.7.4 Audio Signals

In highly illuminated situations and dark environments, the face and body of the subjects are hardly visible; hence, audio signals become relevant in extracting emotional status. For example, in situations of anger or road rage, a driver's annoyed exclamations are characteristic. Speech transmits emotions via linguistic content, yet paralinguistic features of utterances like pitch, voice, intensity, and intonation convey emotion. Vocal bursts like shrieks, groans, and grunts, as well as breathing and laughter, additionally transmit information about affect [127]. While driving, stress, behavior, distraction, and mental workload influence the human voice. Different stress levels of the drivers can be detected by using their voice waveform [130]. Recording speech is a non-intrusive and straightforward method to obtain emotional data, yet drivers are often alone in the car, and therefore uttering is not always present. Audio-based affect recognition can also improve speech-based interactive car systems by transferring the speech patterns of the driver onto the automated voice [104]. Vocal signals can give information about the affective state of the driver, especially the level of arousal. Some researchers assume that each emotional state has a constant vocal pattern, and some argue that vocal uttering can originate in various unpredictable states. However, it is mostly agreed upon that some parts of a voice pattern are mappable to emotions, e. g. some voice signals during arousal are distinguishable from those during other emotional states [131].

### 3.7.5 Physiological Measures

we, as humans, intuitively use visual and vocal cues for conveying emotions. However, sometimes body signals like trembling or irregular respiration provide reliable information about the inner emotional processes [132]. The body uses various mechanisms as a response or trigger for emotions. The autonomic nervous system (ANS), consisting of respiratory functioning indicators, cardiac functioning like blood pressure and heart rate, as well as electrodermal measures (skin conductance), are the most common physiological measures [117]. Emotion-related changes of physiological variables have been investigated closely over the past years and, according to [133], are best understood among existing emotional measures. Even if a person does not represent intensive emotions, physiological patterns give away the subject's emotional state [134].

Physiological measures are controlled by the sympathetic nervous system responsible for triggering arousal and the parasympathetic nervous system that reduces arousal, hence forming the autonomic nervous system. In most research, it is assumed that changes in the autonomic system result from variance in affect. It is not to be neglected that various non-emotional states like mental effort or attention also result in autonomic changes [135]. Physiological measures are suited well for experiments in the lab environments, however they are inconvenient for the applications in a real-life emotion detection systems in cars since most of the respective approaches rely extensively on human assistance and require comparably a complex sensory infrastructure to be integrated inside the cabin in automotive domain.

**Electrocardiogram** The electrocardiogram is the most occurring physiological measure in studies with a focus on detecting aroused emotions. The average action potential (neuronal impulse) is measured with electrocardiographic (ECG) sensors on the skin. Heart rate (HR), interbeat interval, heart rate variability (HRV), and skin temperature are collected through these sensors, which differentiate positive and negative emotional activity, indicating e. g. mental effort and stress [136]. Heart rate describes the number of heartbeats per minute, HRV defines the time differences between heartbeat sequences. During emotional driving situations, a change in HR, and HRV signals can be noticed due to the fluctuations in ANS activities, e. g. stress is directly related to heart rate. Another relation of interest is that drivers' HR and HRV signals are affected by contextual parameters such as vehicle maneuvering, traffic volume, alerting events, road direction, driving across various routes, driving tasks, and drivers' emotions and fatigue. HRV is shown to be increased by driving task initiation and decreasing after passing alerting events [130].

**Electromyogram** An electromyogram indicates muscle activity by measuring the voltage of a contracting muscle [136]. Muscular reactions while driving can occur unconsciously, even in the4 absence of significant physical movements. Muscle activity can be effectively recorded with sensors on the shoulders or the facial muscles [130].

**Skin Conductivity** Related to the electrocardiogram, skin conductivity is when the skin changes its conductivity depending on the emotional experience, especially arousal differences. It is also known as galvanic skin response. The SC sensor applies a small voltage to the skin and measures the transient conduction or resistance. The conductance is mainly changed by sweat and pore size, so the sensors are usually applied in the palms of the hands, fingers, or feet where eccrine sweat glands are located. The skin conductivity is divided into two components: the slow-moving tonic component that indicates the general activity of the perspiratory glands (e. g. due to temperature) and a faster phasic component that is influenced by emotions. During an arousing emotional experience, the skin conductance increases fast due to a higher sweat release. Generally speaking, skin conductivity is a successful measure for driver arousal; nevertheless, intersubject variability between drivers can cause fluctuations. This is because the number of

sweat glands and the different positioning of the sensors varies for each person. Therefore, a safe option is to combine skin conductivity with HR or HRV to increase the overall robustness [130, 136].

**Respiration** Emotional states trigger a change in respiration activity. The respiration system is a metabolic and homeostatic regulator of the speed and depth of breathing. The respiration response in an in-cabin environment can be collected with belt or thorax sensors which are flexible so that the amount of stretch caused by breathing can be measured. Alternative options are using a thermistor in the nose and mouth or a flow meter [130]. More specifically, the rate of respiration and depth of breath are commonly tracked. Respiration rate shows relatively low values during relaxation and rises during startling events or tense situations; it demonstrates irregularity when experiencing negative emotions [136].

**Pupillary Dilation** Pupillary dilation stands as an intersection between facial expressions and physiological measures. Besides surrounding illumination, the pupil diameter indicates sympathetic activation; more precisely, the ANS controls two iris muscles (sphincter and dilator) which determine the size of the pupil. Pupil constriction is often driven by parasympathetic activity, while pupil dilation is regulated by the sympathetic pathway [137]. As an emotional measure, the pupillary response is a way of detecting expressive emotional states such as stress [130]. Recent studies demonstrate that arousal-induced pupil dilation is mainly mediated by sympathetic activation, while pupil dilation related to saccade preparation is primarily mediated by parasympathetic inhibition [138]. This demonstrates the role of affective processing in evoking pupil dilation.

## 3.8 Multimodal Architectures and Fusion Approaches

Multimodality, as stated before, refers to utilizing more than one modality to increase the accuracy and reliability of the predictions. In order to achieve this goal, there exist different fusion techniques, such as *feature-level*, *kernel-based*, *model-level*, *score-level*, and *decision-level* fusion [139]. However, feature decision-level fusion approaches are the widely used techniques in the field of multimodal emotion recognition. In *feature-level* fusion, the data from separate modalities are aggregated by using machine learning techniques like *Support Vector Machine* (SVM) [140], decision trees, or HMMs and then is used as an input to an emotion classifier [141]. In contrast, the idea behind decision-level fusion is to combine multiple weaker base classifiers to reduce the uncertainty of the predictions made by the emotional classifier itself. Therefore, each modality has its trained model, and the predictions of those models are combined to one single output [142].

One of the considerations for every multimodal solution in emotion recognition systems is the capability of being trained in an end-to-end fashion. This feature enables facial-based approaches, such as the RBG camera recordings and thermal imaging and provides a baseline for other modalities, namely behavioral factors or physiological signals. There

is a wide range of fusion methods that can be utilized for a multimodal emotion recognition system. One fusion approach, for instance, is a rule-based fusion, such as a voting mechanism that assigns weights to the decisions of the classifiers of each modality [143]. Such an architecture is comparably and computationally efficient and easy to optimize. Custom rules can enhance the architecture concerning the given modalities. However, this system is not flexible enough to add or remove modalities. Furthermore, such an architecture would be more sensitive to outliers since the rules are made specifically based on the data of the given modalities. Research on classification-based approaches such as SVM indicates low performance on large and noisy datasets [144]. Therefore, joint training is a more reliable alternative. In order to build a joint model, a concatenation of the feature vectors must be built. Alternative operations such as *tensor fusion* [4], where the joint model is created from the cartesian product, are computationally expensive approaches; hence, they are not a preferred choice for the applications of this domain. More importantly, incorporating more modalities would be infeasible. The concatenated feature vector can then be fed into a convolutional neural network(CNN)/recurrent neural network(RNN) network. More involved multi-view systems like memory fusion network (MFN) [5], which uses a system of RNNs, have the potential of performing better due to the attention mechanism. However, such a system incorporates significantly more parameters than a single RNN or even a double-stacked RNN network. Furthermore, due to the modeling of the inter-modality dynamics, it will be tuned and optimized more specifically to the given set of modalities. Therefore, it will not be robust enough as expected by adding new or removing existing modalities.

From an internal perspective, incorporating different features to increase machine learning methods' performance poses the challenge of handling the heterogeneity gap of different modalities. *Joint representation* is a reliable candidate to project unimodal representations into a common semantic subspace, where multimodal features can be fused [145]. The most straightforward approach is to concatenate the feature vectors directly. However, another direction is to implement the subspace by a distinct hidden layer, in which modality-specific vectors will be added, combining the semantics from different modalities. The following equation describes this property, where z is the activation of output nodes in the shared layer, v is the output of the modality-specific encoding network, and w is the weights connecting the modality-specific encoding layer to the shared layer [146]:

$$z = f(w_1^T v_1 + w_2^T v_2) \tag{3.1}$$

Modeling the dynamics within a modality and across modalities is a serious challenge in multimodal learning architectures. A commonly used fusion method is early fusion, which simply concatenates the feature vectors from each modality and does not consider inter-modality dynamics. Zadeh *et al.* at [4] present a tensor fusion network for multimodal sentiment analysis, which aims to model inter-modality and intra-modality dynamics. To this end, they use a 3-fold cartesian product from the modality feature vectors as an early fusion method, which models the unimodal, bimodal, and trimodal interactions as depicted in Figure 3.5.

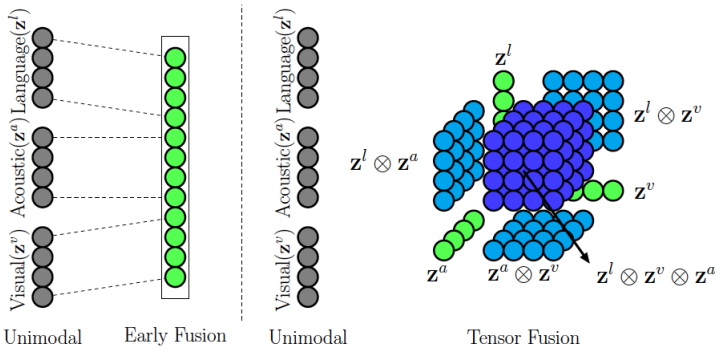



**Figure 3.5:** Tensor fusion architecture [4]

In their following work at [5], the same authors present a memory fusion network that outperforms the previous approach. This architecture consists of three components: a system of long-short term memories (LSTM), delta-memory attention network (DMAN), and multi-view gated memory. The system of LSTMs aims to model the dynamics of each modality independently as depicted over Figure 3.6.

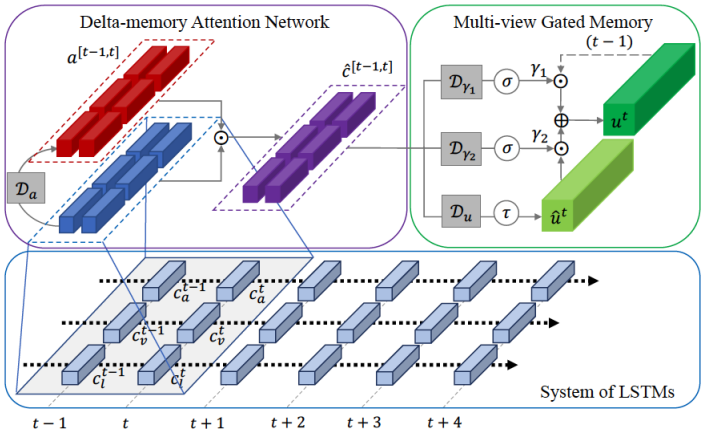



**Figure 3.6:** Memory fusion network (MFN) [5]

Therefore, each modality is assigned to an LSTM function. The DMAN is constructed to discover the dynamics across modalities and temporal interactions. In particular, the DMAN assigns a relevance score to the memory dimensions of each LSTM. The multi-view gated memory is a dynamic memory module that stores the cross-view interactions over time. The authors replace the DMAN module in the MFN architecture with a dynamic fusion graph, improving performance and interpretability.

Incorporating deep neural networks for building a multimodal end-to-end recognition system is beneficial; however, it is imperative to note that other learners which are not based on deep neural networks can perform better in certain domains with considerably lesser requirements regarding computational resources. For example, *multiple kernel*

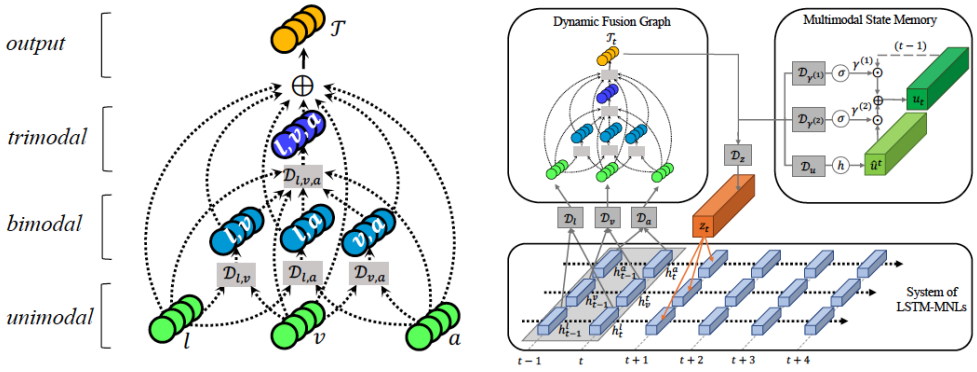



**Figure 3.7:** Graph-MFN [6]

*learning (MKL)*, which extends a kernel SVM by using different kernels for different modalities of the data, is one of those none deep neural network (DNN)-based solutions. MKL is more suitable for fusion concerning multimodal heterogeneous datasets than a conventional SVM since kernels can be regarded as similarity functions between data points [145]. They were often used for object detection before the advancements of deep learning architectures surpassed them. Furthermore, they were also used previously for multimodal emotion and affect recognition and demonstrated acceptable performance [147, 148, 149]. Even though nowadays deep learning methods are preferred, MKLs still have significant advantages over them. MKLs have been used in various domains, whereas many deep learning methods for multimodal fusion are developed for specific domains. Deep fusion systems are developed for audio, visual, and textual modalities in multimodal sentiment and emotion recognition. In the automotive domain, however, as stated before, audio and textual modalities are not very viable options. The limited availability of computational resources puts restrictions on the utilization of such computationally expensive approaches.

It is notable that by the ever-growing focus on machine learning-based solutions, new architectures are being introduced by the researchers in emotion recognition irrespective of use case's domain; hence, keeping a consistent track of them is also a challenging task. Nevertheless, we have performed a comparative analysis of the state-of-the-art machine learning-based architectures by incorporating fusion technique, modality-specific features, evaluation metric, detected emotions, overall performance, and the utilized datasets during our preliminary evaluations. The top 20 of them are represented in table 3.3 and table 3.4 to provide a high-level overview regarding the current status of the research and development in this domain.

## 3.9 Shortcomings of Machine Learning Approaches

Machine learning methods are the core of emotion recognition systems in most cases; however, they are not a perfect tool in the development toolbox and, in some cases, suffer from serious shortcomings. A broad range of factors can cause weaknesses of

machine learning models in emotion recognition. Some of them are due to the quality or the distribution of data, some of them arise because of the model architecture, some occur during the deployment. The consequences for misclassifying an emotional state can be very high. For example, if the model does not recognize an unsafe emotional state, the risk of an accident increases or in case of misclassifying a safe emotional state, the vehicle initiates some driving limitations unnecessarily which will increase the dissatisfaction level with the vehicle. We can group the shortcomings into the following categories:

**Lack of diversity in data:** The performance of machine learning models highly depends on the data they are trained upon them. Additionally, the training phase requires large amounts of data on the subject matter. Moreover, the data must be diverse to encompass all the different cases occurring and create a precise and universal machine learning model. The model will provide unsatisfactory results if the training data has a different distribution from the real-world data. By training a machine learning model for emotion recognition in the automotive domain, one must ensure this criterion is met accordingly. This complicates the data gathering because people vary in how they look, sound, and act when expressing emotions. The skin color, facial features, language, voice tone, accents, gender, age, and culturally accepted emotional expression come into play when gathering a well-balanced dataset for training emotion recognition models [169].

**Reliability of the data:** Several factors are coming into play when it boils down to data reliability. Labeling the driver's emotions during data collection is crucial for training supervised machine learning models. Before each experiment, one has to induce the drivers' emotions artificially, which can deviate from emerging natural emotions in driving scenarios. In general, two methodologies are used for annotating training data, namely internal and external annotation [142]. The more common method of internal annotation requires the driver to assess his affective state by himself. This is a subjective measure since each individual assesses his internal emotions differently, leading to discrepancies in the dataset. The second approach is external labeling which is executed by a third-party supervisor assessing the driver's emotional state by observing their physiological responses. This process is highly error-prone because humans express their emotions very differently depending on personality and culture. Hence, the data labeling for an emotion recognition dataset with mentioned sub-optimal measures negatively affects the dataset's quality and can be a significant factor for low accuracy values at the model inference. In addition to that, there are certain limitations to the level of accuracy that one can reach depending on the data type. For example, audio-visual data is easily collected; however, it can be compromised. People can intentionally and unintentionally fake emotional expressions [170], or in the case of some neurodiverse individuals, not even be able to express them adequately [171]. Also, the expression of an emotional state heavily depends on the culturally accepted norms [169].

**Context-sensitive data:** In a fundamental perspective, the context describes where a vehicle and its neighboring vehicles are situated and what is happening inside and outside the vehicle. Some data types, such as driving behavior data, are context-sensitive. Using them outside of the context can be very misleading. Boonmee *et al.* [172] developed a model based on driving behaviors for a bus: take off, brake, turn left, and right. For the model to function as desired, it had to rely on the GPS receiver input to locate the vehicle precisely and integrate the involved factors in shaping the desired context. The probability for false positives was very high in mountainous areas and roads with sharp turns. The driver would have to perform a lot of quick steering and braking associated with aggression, which can falsify the outcome relying on those factors.

**Lack of in-cabin data:** The specific lighting, noise, cabin design, and interaction with the in-cabin interfaces create an environment where a general emotion recognition model would suffer from inaccuracy because it was not trained upon data capturing similar situations. Li *et al.* [173] analyzes how the activation in parts of the face (as known as action units) differs between dynamic driving scenarios (collected in the DEFE dataset) and static life scenarios (collected in the JAFFE dataset). It was shown that people are less likely to enable the action units while driving, meaning the emotional expression is suppressed. Such a difference in expression style can be described due to the additional cognitive tasks that come with the driving task.

**Implementation and real-time capability:** By deploying machine learning models for real-time applications in intelligent vehicles, a fundamental requirement is a short processing time at inference to ensure the presence of predictions within the boundaries of timing limitations. In some cases, providing predictions with a delay neglects the purpose of notifying the driver on time. For instance, warning a driver in a drowsy state is highly time-critical. A machine learning-based system processing input from multiple sensors (e. g. images from multiple cameras) increases inference time [174]. Especially multimodal models with separate pipelines to process each input result in comparably longer processing times and more memory demand. Moreover, complex deep learning models often come with the downside of high computational costs and loss of real-time capability. To overcome this limitation of high computational effort, feature extraction or dimensionality reduction methods are considered for some applications in real-time implementation [175].

**Model sensitivity and invariance:** Training machine learning models is a challenging task, specifically when generalizability is the primary concern, meaning the ability of the developed model to adapt to unseen scenarios. For example, speech-based methods record a loss in recognition rates when receiving audio input with background noise included [175]. Different types of noises can occur in a driving situation, such as engine noise, noise due to road conditions, or passengers' creation. Often researchers try to overcome this issue by artificially adding noise to the training data. However, even in that case, it is impossible to cover all potential noise types and sources. This results

in a lack of generalizability of the model and hence, affecting the accuracy negatively. Similar problems arise for predictions of models based on bio-signal methods since a change in humidity or ambient temperature highly affects sensor measurements. For example, Leone *et al.* in ablation studies [176] investigate the performance loss of their developed approach as the result of missing invariance to yaw angle, pitch angle, and lighting condition of the model. With the rotation of the driver's yaw angle by 40 degrees or the pitch angle by 20 degrees, a drop in prediction accuracy by roughly 10 percent can be reported. In addition, an increase in luminous emittance from 100 lx to 500 lx lowered the accuracy of classification results by an average of 5 percent.

| | Architecture | Fusion | Modalities & Features | Emotions | Performance | Train | Test | Reference |
|---|---|---|---|---|---|---|---|---|
| 1 | SVM | decision | face data, steering wheel angular velocity, acceleration | valence, arousal | acc. 93% | own database: 15 subjects | own database: 8 subjects, CK+ | [139] |
| 2 | SVM | decision | sitting posture, blinking | drowsiness | F1: 0.536<br>RMSE: 0.620 | own database: 50 subjects | subject-wise cross-validation | [150] |
| 3 | SVM | decision | 3D data from face, head, hand gestures, body movement | 6 emotions | MSRC-12: avg. acc. 45.7 %<br>UCFKINECT: avg. acc. 23.4 %<br>MSRACTION: avg. acc. 46.3 % | own database: 15 subjects | MSRC-12 | [151] |
| 4 | SVM + ANN | decision | face data, EEG | 4 emotions,<br>3 levels | online: acc. 81.25 %<br>offline: acc. 82.75 % | own database: 20 subjects | subset of own database | [152] |
| 5 | SVM + ANN | feature | vehicle data, physiological data, face data, eye tracking | stress | KNN: acc. 56.06%<br>SVM: acc. 61.59%<br>ANN: acc. 65.09%<br>STACK: acc. 63.22% | own database: 68 subjects | one-subject-out cross-validation | [153] |
| 6 | SVM | feature, decision | EEG, eye tracking | 3 emotions | feature : acc. 73.59%<br>decision : acc. 72.98% | own database: 5 subjects | subset of own database | [154] |
| 7 | BDAE | feature, decision, BDAE | EEG, eye movements | 4 emotions | feature-fusion: acc. : 75.88%<br>DNN: acc. : 85.11% | own database: 44 subjects | subset of own database | [155] |
| 8 | BDAE | feature, BDAE | eye movement, EEG | 5 emotions | avg. acc. 74.6% | SEED V | cross-validation | [156] |
| 9 | CDBN | feature | facial features, body gesture, vocal expressions, physiological data | 23 emotions | emoFBVP: acc. 83.2%<br>CohnKanade: acc. 97.3%<br>mind reading: 93.4%<br>DEAP: acc. 79.5%<br>MAHNOB-HCI: acc. 58.5% | emoFBVP | Cohn Kanade, MindReading, DEAP, MAHNOB-HCI | [157] |
| 10 | BDAE | feature | EEG, eye tracking | 3 emotions,<br>valence, arousal | SEED: acc. 93.97%<br>DEAP: avg. acc. 83.53% | SEED , DEAP | subset of train set | [158] |

**Table 3.3:** Part1: Comparative analysis of state-of-the-art ML-based architectures

| | Architecture | Fusion Technique | Modalities & Features | Emotions | Performance | Train | Test | Reference |
|---|---|---|---|---|---|---|---|---|
| 11 | BDAE | BDAE | EEG, forehead EOG | fatigue | COR: 0.852<br>RMSE: 0.094 | own database: 21 subjects | cross-validation | [159] |
| 12 | CNN + LSTM | feature | auditory and visual modalities | valence, arousal | avg. $\rho_c$ 0.76 | RECOLA | subset of RECOLA | [160] |
| 13 | CNN | feature | face data, audio, text | 4 emotions | MOUD: acc. 96.55%<br>IEMOCAP: avg. acc. 76.85 % | MOUD | MOUD, IEMOCAP | [161] |
| 14 | CNN + LSTM | model fusion | ECG, vehicle and environmental data | stress | avg. acc. 92.8 % | own database: 27 subjects | subset of own database | [162] |
| 15 | LSTM | feature | heart rate, mouth and eye features | drowsiness | acc. 90.5% | own database: 5 subjects | na | [163] |
| 16 | RBM | RBM fusion | galvanic skin response,<br>EEG, skin temperature | valence, arousal | DEAP: acc. 60.7%<br>MAHNOB-HCI: acc. 59.1% | na | DEAP, MAHNOB-HCI | [164] |
| 17 | CNN + LSTM | decision | EEG, gyroscope data,<br>image processing data | fatigue | acc. 93.91% | own database: 5 subjects | cross-validation | [165] |
| 18 | OKL-RBF NC | score | face data, audio | 6 emotions | ORL: acc. 98.5%<br>YALE: acc. 99.5%<br>CK+ : acc. 96.11%<br>ENTERface'05 : acc. 86.67%<br>RML: acc. 90.83% | ORL, YALE, CK+, ENTERface'05, RML | ORL, YALE, CK+,<br>5 ENTERface'05, RML | [166] |
| 19 | ANN, RF | feature | face data, EEG | 9 emotions | NN: acc. 88.16%<br>RFC: acc. 97.7% | MAHNOB-HCI | subset of MAHNOB-HCI | [167] |
| 20 | RF + SVM | feature | heart rate,<br>galvanic skin response | 3 emotions | DEAP (subset): avg. acc. 97% | own database: 37 subjects | DEAP (subset) | [168] |

**Table 3.4:** Part2: Comparative analysis of state-of-the-art ML-based architectures

# 4 Methodology

## 4.1 Safety Violation Identification Framework

The formal concept of safety is not easy to grasp from a development perspective of AI-based technologies, especially when safety can be seen as a feeling based on the individual's own experience. For example, the important metrics used for current self-driving car implementations are the accident-free driven kilometers, the count on necessary takeovers by the safety driver, and the general well-being of the occupants [177].

From safety engineering perspective, establishing a valid safety framework for evaluating the safety-critical applications of intelligent vehicle platforms is challenging due to various regulations of different countries, the complex and often unpredictable outcomes of the deployed methods, and the lack of proper well-defined standards. As stated in the previous sections, the utilization of machine learning-based approaches to increase the level of autonomy brings along several sources of uncertainty as well. Reinforcement learning is the most complex black box in this regard, considering that developer experts can provide the *right* and *wrong* actions for the driving agent only during the initialization phase. In this context, the argument that agent continuously learns safe actions is doubtful and can often not be generalized because encoding the complete knowledge into a single numerical function is highly error-prone. A good example is reward hacking, in which the reinforcement learning algorithm collects many rewards without reaching the actual goal by exploiting a bug in the reward function [47]. From the automotive functional safety point of view, the V-shaped development model is a reliable solution in product development [34]. The V-shaped model carries a solid requirement that will be the primary input of the product's safety validation. However, gathering a complete set of requirements for a machine learning-based application is complex due to the level of uncertainty associated with these models. By entering the era of autonomous driving, the responsibility will shift from the human driver to the car itself for driving tasks. At the same time, behavioral safety is still a fundamental part of this development chain. This fact demonstrates the significance of an evaluation phase for the deployed agents to avoid incorrect behaviors that lead to severe accidents. Following the research goals of this work, we aim to facilitate the integration of safety rules and concerns in the development stage of AI-based applications, particularly the automated driving agents; therefore, we propose a framework for an intuitive setup of safety measures and self-driving car agents with an exclusive focus on reinforcement learning-based scenarios in the CARLA simulator. To validate our approach, we consider the several executions of the RL agent to gather the required information indicating safety violations that are exposed to the agent. Those safety violations are then mapped and visualized in the end. Developers and safety engineers can use them to analyze the performance of the

deployed agent from a safety perspective. It is essential to differentiate between the ideas of conventional *statistical* approaches, and the *runtime* approaches here. Statistical approaches use existing data such as reported accidents and visualize them accordingly. While they are currently only relevant for safety from the perspective of planning and defusing dangerous road segments, they still could play a major role in automated vehicles and the autonomous driving era. Traffic accident maps like *Unfallatlas*/(Germany) [178] or *CrashMap*/(Great Britain) [179] can be seen as the most famous uses cases of such approaches. These maps display the accidents based on their location and further information such as severity, affected means of transport, and the incident's date. The *Unfallatlas* also represents the accident frequency for a given stretch of road. On the other hand, Runtime approaches evaluate the safety during driving since some locations, and respective situations are *labeled* as safer than others. *Time to Collision* (TTC) or *Time to Brake* (TTB) are also among the metrics that the researchers employ to define the safety level of different situations [180, 181, 182, 183]. One example of a runtime approach is the *Responsibility-Sensitive Safety* (RSS) proposed by Mobileye [184]. This approach is based on *safe distances* to define a *dangerous situation*, for which *proper responses* are defined accordingly. NVIDIA proposes a similar approach with the *Safety Force Field* (SFF) [185], which predicts the environment and mitigates harmful scenarios. Other approaches observe autonomous driving safety by reading sensors or data buses and evaluating them based on predefined rules [48].

### 4.1.1 Framework Architecture

The proposed framework utilizes the concept of safety measures, which are activities, precautions, or behavioral codes to avoid unnecessary risks and maintain safety. Moreover, it enables safety measures based on predefined rules, proven practices, and accepted guidelines in a real-world simulation environment. The original concept of safety measures is not new and already well established in the behavioral safety domain, with famous examples like traffic regulations or rules for defensive driving. Being quantifiable is the most important advantage of the safety measures. For instance, it is possible to determine whether drivers are violating the speed limit or are tailgating. In our proposed framework, safety measures are based on integrating expert knowledge on top of simulated situations that statistically may have higher risks for injuries. A severity level is assigned to each safety measure to quantify the negative impact on safety. The respective measures are named as *safety constraints* in our development, and by violating a constraint, a *safety violation* is recorded accordingly. The overall architecture of the framework is separated into three stages: **Initiation**, **Execution**, and **Analysis**, as outlined in Figure 4.1.

The *Initiation* phase consists of two different sections, one for application developers (🧪) and the other for safety engineers (🔍). The *Agent* interface provides a platform for application developers to integrate the developed approach as an RL-based agent in our case. *Safety Constraint* interface also contains a set of safety restrictions respectively. In the *Execution* phase, the agent has to drive in the predefined environment and is evaluated by the given safety constraints. This phase is completed after a stop criterion

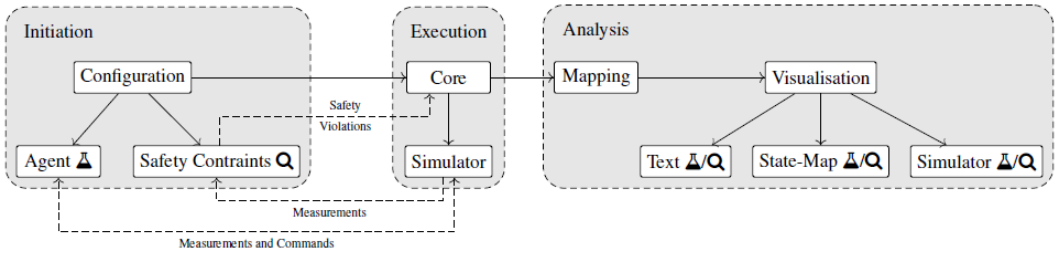



**Figure 4.1:** Architecture of the proposed framework – Roles: Developer, Safety-Engineer

is matched. Then, the safety constraints are evaluated against the current situation and, if required, trigger a safety violation that contains relevant information about the current situation among other agents, types, and locations. At the end of this stage, the framework perseveres the given events. In the last stage of *Analysis*, the safety violations are filtered, mapped, and visualized. The location and the type are the primary parameter for the grouping but could vary in extended developments of the framework in the future. The framework calculates different safety indicators for each group to make the groups comparable. The generated groups are visualized more intuitively concerning the calculated safety indicators and allow the developers and safety engineers to understand the system better.

## 4.2 Safe Operation Monitoring and Enforcement

From a technical point of view, identifying and covering *all* of the potential safety-critical situations is not possible in practice. However, we assume that the required reaction to being taken in the presence of safety violation and respectively *fail-safe* mode is known beforehand and could include slowing down the car, bringing the car to a halt, or even handing over the control to the human driver. Moreover, since we are mainly focusing on safety assurance for AI-based software, we keep the risk assessment part out of the scope of this work. Respectively, as it is thoroughly addressed in our work at [186], we identify four main perspectives that can be utilized in order to ensure the safe operation of a neural-network-based system, as follow:

### 4.2.1 Filtering Anomalous Operational Inputs

This method targets the problems that originate from differences in training and operational conditions and builds on the idea of online data monitoring.

The fundamental intuition of this approach, as depicted in Figure 4.2, is to calculate the distance and identify how *far away* is the input from the data the system was trained on. In other words, the aim is to detect whether the input is an *anomaly*, i. e. a data point that is significantly different from the original data. If that is the case, the system is expected to enter a fail-safe mode; otherwise, the system's regular operation continues.

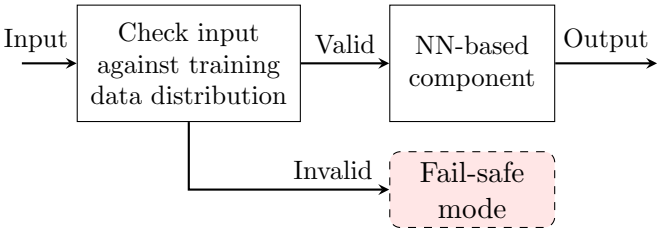



**Figure 4.2:** Control flow of the anomaly detection approach

Given some data $X$, variational inference (VI) [187] aims to find a distribution $\mathrm{Q}(Z)$ which is as similar to the actual posterior $\Pr(Z \mid X)$ as possible, where the distance between the distributions can be calculated using the *Kullback-Liebler Divergence* as known as relative entropy. Applying variational inference is proposed for this type of online detection of anomalies initially in [188]. The advantage of this approach is that the characteristics of expected input are learned from the data; hence, no further specific feature engineering efforts are required. It also demonstrates that this approach is highly generalizable and is not limited, and is use case-independent. Simply exposing the system to data for modeling the environment can help the system draw the required inferences.

### 4.2.2 Ensuring Coverage of Positive and Negative Cases

In the example of maneuver planning system previously mentioned in Section 3.2.1, the component under evaluation should be able to predict lateral and longitudinal actions and the outputs that may lead to adverse outcomes such as driving off the road, or a crash, or in general ending up in hazardous situations.

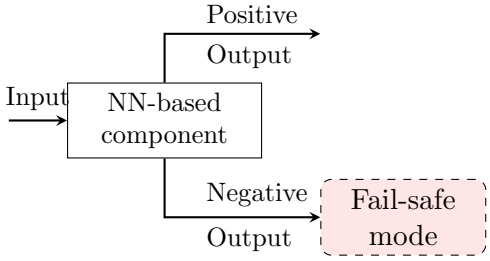



**Figure 4.3:** The control flow of the approach based on predicting possible positive and negative outputs

In such a system, if the output of a workflow falls in a negative class, the system would enter a fail-safe mode; otherwise, it would continue generally functioning as before, as it is visualized in Figure 4.3. This setup benefits from a higher assurance of the system being trained on under-represented or rare situations/inputs; hence, leading to a better response to safety-critical situations. Since the system learns expected desired and undesired outputs from the data directly, it would generalize well to other use cases

without explicit specifications. It is also comparably easier to implement and more intuitive than other approaches.

### 4.2.3 Defining Environmental Constraints

Ontologies are practical tools to model the entities and relations in a system and the constraints [189]. To apply this approach in *design-time*, the automotive safety engineer needs to create a safety ontology structure based on specific software-system functions and context. The main ontology topics (for functional safety) can be derived from ISO26262 (Part 1 - Vocabulary) [34]. The concepts stored in ontologies will be internally translated into machine-readable first-order logic (e. g. Prolog code), making it more straightforward to describe constraints that the system must obey in the environment. Thus, ontologies can be seen as a *safety blanket* around each machine learning-based component, as depicted in Figure 4.4. Inputs to the component and outputs generated will be tested against a set of environmental constraints to ensure that they fulfill the requirements for a safe operation; otherwise, the system enters a fail-safe mode.

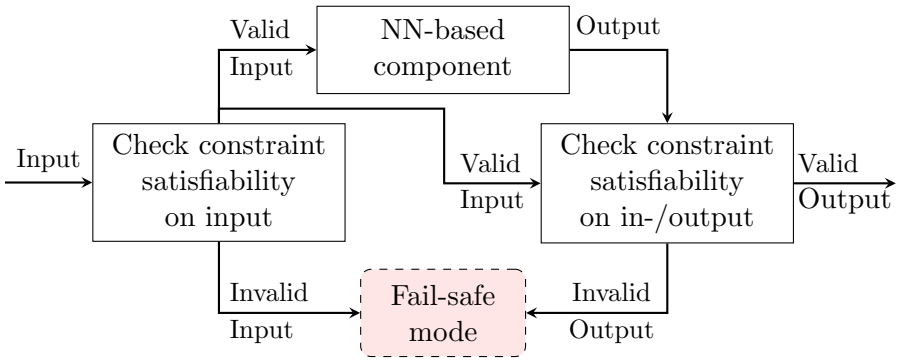



**Figure 4.4:** Control flow of ontology-based constraint satisfaction approach

This ontology-based solution improves the system's reliability and follows the principles of formal verification and validation methods, enforcing the developed system to abide by the intuition of human actors. Moreover, it can improve the traceability of issues and help to magnify the bottlenecks of the system.

### 4.2.4 Pre-exploration Using Reinforcement Learning

This approach aims to augment *learning* itself with two trainable components. As depicted in Figure 4.5, initially, a reinforcement learning agent is responsible for exploring the environment. Following that, it describes the online neural network implemented in the standard manner for the component in question. The reinforcement learning agent is expected to learn by exploring and interacting with its environment; therefore, it would be trained via simulations. Since simulation-based testing does not pose a real threat to human lives, it is also possible to explore negative outcomes. The agent would learn and thereby generate a map of situations, actions, and associated reward values. This

mapping can then be used to categorize situations that lead to different levels of risk, namely high, medium, and low, based on the reward values of each state. This approach can be seen as an extension to the monitoring techniques, wherein, rather than manually labeling the state space as being safe or not, the output of reinforcement learning agent is used to generate such a mapping, with the reward function determining the severity of the hazard for each state-action pair. Thus, every input being passed to the neural network-based component would first be checked against the safety invariance mapping to enter a fail-safe mode when the input is in a catastrophic zone. In generalizing to other use cases, this approach could do quite well with the limiting factor of additional hyperparameter tuning required for the agent. The advantage of such an approach is that rewards and objective functions can also be set up to be more aligned with human intuition, thus making the system more compliant with human expectations.

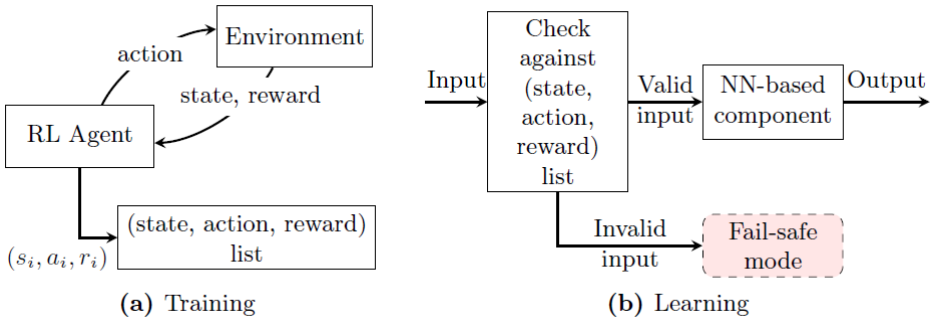



**Figure 4.5:** Control flow of RL-based pre-exploration approach

Our newly introduced concept of *Crash Prediction Networks* (CPN) [7] is solely based on this approach, that is represented in the following sections:

## 4.3 Crash Prediction Networks (CPN)

The majority of the modern approaches relate to testing a developed model before being deployed in the operational environment. Machine learning-based components, however, suffer from serious problems, which were stated earlier at 3.2.1 and it leaves such components vulnerable to errors. Thus, it is necessary to focus on monitoring-based approaches, which are gaining more interest recently, to alleviate the safety concerns associated with such systems [190]. Therefore, an end-to-end deep learning model for lane change maneuvers has been chosen to elaborate on the specifics of the proposed solution. Such a model uses a deep neural network that takes input data from sensors representing the environment around the ego vehicle and generates one of three actions, either allowing the ego vehicle to continue driving in the current lane or to switch to the left or right lane depending on the presence of obstacles. The *Crash Prediction Networks* concept, in brief, involves a neural network model responsible for determining the likelihood and severity of a crash at any given time step. The model takes into consideration multiple features such

as the output of the perception module of the vehicle, planned trajectory/action of the ego vehicle, predicted (or intended, if available through C2C communication) trajectory of the obstacles, and possibly also historical information such as number and severity of previous crashes that the ego vehicle and obstacles were involved in. Specifics of the system can be understood by distinguishing between the training and, after deployment, the operational phases of the model. The training phase relies on the model receiving the required input values for the previously described feature set and knowing whether a crash incident has been recorded. Hence, the model requires an architecture that involves a reinforcement learning environment to allow the model to know the outcome at every step for a given set of feature values. It would also allow the vehicle to crash often, as it is one of the notable characteristics of RL agents, especially at the start of the training phase. Therefore, we propose to train the model by allowing it to spar with an RL agent such that the ego vehicle closely imitates a real-world vehicle that can perform tasks, similar to the lane change maneuver use-case described earlier. At each step, the RL agent and the crash prediction network will access information about the vehicle's environment. Thus, the crash prediction network will predict whether a crash will occur, while simultaneously, the RL agent would interact with the environment to determine whether a crash has occurred. Based on the differences in the output of the two networks, the crash prediction network would be updated to eventually acquire the ability to predict crashes with a high level of accuracy. The operational stage of this model is designed such that the inputs are fed as usual to the machine-learning-based component, which is responsible for determining the lane change maneuver in the ego vehicle. The vehicle, however, does not act directly on the generated lane change action command. Instead, the action command and the environmental inputs, in the form of sensor data, are directed to the crash prediction network, which performs the task of predicting the likelihood of a crash. If the determined likelihood is low, then the vehicle is allowed to perform the desired actions; otherwise, the vehicle is pushed into a fail-safe mode which can be multiple options according to the predicted severity of the crash. It is important to note that it needs to learn and improve even in the operational stage of the model to stay relevant to the environment. Thus, the difference between the actual and predicted output is utilized to update the model, similar to the training stage. The concept of CPN does not just focus on driving agents of autonomous vehicles. It can even be utilized in the current status of ADAS developments as well, thereby allowing a smoother transition toward full autonomy of the vehicles in the near future. Respectively, the model can be seen as an intuitive, informed decision-making entity by incorporating data from multiple sources. Additionally, such a system would also generalize and scale appropriately to different scenarios that the vehicle might encounter during the operation. However, one of the significant challenges that would be encountered during the development of the model is handling input data received from different sources in various formats. Besides, redundancy must be taken into account concerning sensor feeds to preserve the system's proper functioning even in sensor failures or malfunctions.

## 4.4 Emotional States and Behavior Modeling

It is proven that stressed drivers tend to pay less attention to the traffic and environment. In this regard, the in-cabin behavior of the driver can indicate specific emotions with a high level of certainty [41, 191, 192]. However, the challenging aspects like modeling the in-cabin behavior and the exact mapping mechanisms to the emotional states according to the in-cabin environment must be revised and enhanced following the integration of sensory information into the emotion recognition architectures. Therefore, the first step in studying the behavioral-based factors in such systems is to adjust and revise the definitions according to the context of the environment. Our initial objective is to develop *behavior profiles* that represent driving style patterns influenced and triggered by certain emotions. For this purpose, we aim to design an empirical study through an online survey to analyze and validate the commonly agreed assumptions for the impact of emotions on driver behavior. These patterns are then utilized to classify the emotional states according to driver in-cabin behavior. In emotion recognition, the utilization of behavior and its sub-modalities is relatively neglected compared to the extensively investigated modality of facial expressions. It is also evident that multimodality is a solution to achieve a comparably more robust prediction pipeline which is also one of the main objectives of our work here. Following this, we aim to incorporate the behavior-based emotional indicators, the developed profiles, and the facial modality into a suitable architecture for the in-cabin environment and demonstrate the performance of such architecture with regard to the system's robustness for in-cabin environments.

### 4.4.1 Behavioral Indicators

In order to investigate the impact of the integration of driver behavior into emotion recognition pipeline, and respectively, identify the requirements and considerations on the way of building a structure capable of classifying the emotions continuously in an efficient and non-intrusive manner, as a first step, it is crucial to provide a proper definition for *in-cabin behavior*. We define in-cabin behavior as *"the response to various stimuli, whether internal or external, conscious or subconscious, overt or covert, and voluntary or involuntary inside the cabin"* [139]. Following that, we narrow down the in-cabin behavioral factors of the driver to only **vehicle acceleration intensity**, and **steering wheel angular velocity**, as the main under-examination behavior-based determinants of the driver's emotional states [191, 193]. Respectively, as stated before in Section 3.5, we form our proto-emotional groups as follow:

- **Negative**: anger, fatigue, stress, confusion and sadness,
- **Positive**: happy,
- **Neutral**

### 4.4.2 Multimodal Recognition Architecture

In order to identify an emotional state, we consider behavior-related input data and facial expressions as the primary modalities of our designed architecture. We model each modality separately, meaning we use decision-level fusion to combine their outcomes to increase the robustness of predictions. Therefore, each modality needs to be trained separately, and then the outcomes need to be weighted and combined. The facial emotion recognition module acts as a core component of the system. It can cover all six base emotions of happiness, sadness, fear, anger, surprise, disgust, and the neutral state of the driver. In order to classify the facial expression, each frame of the live stream recording is processed separately. This process can be divided into the following three parts:

1. **Face detection and pre-processing**: Almost all automated object recognition systems start with object detection. Face detection is also a subdomain of object detection; therefore, the same methods are applicable for face detection. Viola-Jones object detection algorithm using Haar feature-based cascade classifiers is one of the effective methods for face detection on the images. It has real-time performance and high true-positive results. However, this approach suffers and cannot detect faces on fluctuating illumination conditions. Therefore, we cannot consider this approach in-car environment. Nevertheless, another commonly used face detection algorithm based on Histogram of Oriented Gradients (HOG) features [194] combined with a linear classifier, an image pyramid, and a sliding window detection scheme performs more accurately and will be utilized in our studies.

2. **Feature extraction**: The first step in feature extraction is to keep only informative areas of the detected face. Therefore the region of interests (ROI) needs to be identified and cropped. Most expressions on a face are recognizable by eye, eyebrows, nose, mouth. To extract the ROI, one can use a facial landmark detector as designed by Kazem *et al.* [195], also available in the dlib library. Then in order to extract important facial features from the ROI and construct feature vectors for the classifier, we can use HOG descriptors.

3. **Classification**: SVMs, random forest, decision trees, and K-nearest neighbors (KNN) algorithms are among the widely used supervised machine learning techniques. Sequential minimal optimization (SMO) [196] is used to train SVM. The memory requirement for SMO is linear in the size of the training dataset. The computation time of SMO depends on SVM evaluation; thus, SMO is fastest for linear SVM and data sets with large sparsity. According to the previous research in this area, SVM usually outperforms the above-listed techniques in performance; however, we will evaluate the existing classifiers and select the one that fits more to our architecture.

Two (abrupt) car maneuvers counters will be developed using machine learning techniques and statistical methods; one based on steering wheel angular velocity and aggressive driver predictor, and the other based on a variation of acceleration intensity. In

the end, all three modules are combined into one final emotion classifier to address the emotional status of the driver in desired time frames.

## 4.5 Emotion Recognition API

In the end, we will have a system that consists of pre-trained classification modules, which can return the confidence across a set of emotions depending on the information input. However, the system resources are inaccessible to other developers or users for further evaluation. Therefore, a set of protocols must be defined to facilitate this issue and provide a platform for acquiring more resources in future studies. A potential approach to address this matter is utilizing an *application programming interface (API)*, which serves as a gateway to a software program. Thus, it allows other (future) systems and users to smoothly interact with the developed system and models. A web-based API is considered for this purpose to be deployed to enable the desired flexibility and accessibility. This API includes the integration of a subsystem into different systems at the same time. Standardization is, therefore, an unavoidable requirement. In the internet of things (IoT), plenty of efforts have been made to tackle the challenge of standardization [197]. Indeed, standardization is an important player because it can unify the elements common to the various systems and make them available to independent heterogeneous systems. An API, from a technical point of view, is a tool that provides a system in a *neutral* state to other systems and applications for integration and evaluation purposes. One of the vital considerations is that an emotion recognition system can exist without an API, and its performance should be identical to a system consisting of one. However, the API can significantly reduce the required workload of future developments with regard to timing, and resource dependency. As APIs are clear pre-defined sets of rules and protocols, their application expects all future developers to use the same rules and formats. Also, using unified API documentation simplifies the development process and helps new developers get onboard faster with more confidence. In other words, the API helps developers test the emotion recognition system in a secure environment and facilitates the integration of new system components. Last but not least, an API considerably facilitates the integration of the existing system into new applications in the future.

However, the communication barrier between the server and the client must be clearly defined for both parties involved. Therefore, the web API can be developed based on two interaction styles: *simple object access protocol* (SOAP) and *representational state transfer protocol* (REST). It is crucial to select the appropriate interaction style for building the desired API because it is the most fundamental design decision [198]. Due to the promising results of the conducted research on SOAP and REST with an identical set of configurations [199, 200], comparably better response time and throughput of REST make it a more suitable choice for the goals of our work. Additionally, traditional SOAP-based web services contain complex protocols that are rarely exposed to links and merely employ HTTP features. In contrast, the REST architecture can be summarized as *resource-oriented*, which adopts our multimodal emotion recognition system features. The resource is a collection of entities, which is addressable and can be returned in

different, yet commonly understandable formats (e. g. XML, JSON). Hence, it enhances flexibility and helps to reduce the complexity of the systems [201, 202]. *Usability* is also another considerable design objective that represents how easy it is for the user to learn, adapt and utilize the API up to different levels. REST indeed satisfies this design objective as well.

# 5 Experiments and Evaluations

## 5.1 Safety Violation Identification Framework

In driving scenarios, a significant safety measure can directly be derived from the definition of safety itself. Suppose the current situation causes any injury on any object (e. g. humans, cars, other objects in the environment, or even immaterial goods). In that case, it is concluded that a violation of safety has occurred. In cars, any injury is directly related and usually results from a collision. In a fundamental definition, a collision occurs when a vehicle collides with another vehicle, pedestrians, or other objects in the environment such as trees or animals. Different types of collision exist, namely single-vehicle collision (e. g. vehicle collides with an object of the environment without any influence on another road user), and longitudinal collision (e. g. vehicle collides with another vehicle that is driving in the same or the opposite direction). The severity of a collision depends typically on the collision type and parameters such as speed, crash hardness, and involved road users. For example, single collisions can cause fatalities or injuries to the body and property damage; conversely, some collisions end only in light car body damages. Therefore, the highest safety goal is to prevent collisions of **any kind** and generally favor light damages on cars against heavy damages and misfortunes. We aim to design a framework deployed in the CARLA simulator to provide the safety engineers and AI developers a shared space to specify safety concerns and enforce the safety rules during the development phase. This framework identifies the occurred violations of the AI-based agent and their severity level during the experimental phase in order to enlighten the required adjustments and effectiveness of safety mitigation mechanisms in advance while preserving the system's overall efficiency. Detection of collisions is simplified and straightforward in the CARLA simulator. The CARLA provides three scalars for collisions with other objects: `collision_vehicles`, `collision_pedestrians` and `collision_other`.

### 5.1.1 Violation Factors

Usually, collisions are seen as a violation of safety and its respective measure; therefore, avoiding them is indispensable for detecting safety-critical situations. In this part of our work, we categorize the safety violation factors as follow:

**Distance:** An intuitive example of collision avoidance safety measures is keeping appropriate distances to the vehicle ahead and maintaining a proper position in the located lane. It is essential from a safety perspective to keep a proper distance to the leading vehicle since the time for reactions and possible evasive maneuvers are highly limited if the

distance between two vehicles becomes too short. Computers are much faster in performing reactions than human operators; nevertheless, these systems rely on measurements from the environment (e. g. lidar/radar sensors). They also introduce latencies between measuring, detecting, and acting phases. In this regard, maintaining an appropriate distance reduces the risk of a collision in most cases. Defining appropriate in this context is not as straightforward as it seems in the first place. Also, legislators do not clarify this feature strictly for human drivers. Most countries specify (in)formal rules of thumb, such as the popular *2-second-rule* or the *half speedometer* factor in countries with the metric system. The *2-second* rule enforces a cushion of minimum distance the car drives in two seconds (for 100km/h → 55,5m). This value may slightly differ between different countries and situations. For example, most countries suggest a *4-second-rule* on wet road conditions [203]. In the proposed framework for safety violation identification, the distance constraint is variable based on an *x-second* rule. The safety engineer can specify the exact value in seconds according to the needs of the designed experiment.

**Lane:** The orthogonal position to the movement of the vehicle is an important safety consideration. The primary focus here is on staying in the correct lane while ignoring the exact position inside the lane itself since most of the regulations for road traffic and research are vague in this regard. However, several cases are accepted (or perhaps even expected) to violate this rule, for example, overtaking maneuvers on a 2-lane road or the bypass over the side-walk in case of an accident or obstacle blocking the road. Regarding this matter, the CARLA simulator provides two options of (`intersection_otherlane` and `intersection_offroad`) on the intersection. It is declared a *major* safety violation if the vehicle leaves the lane, either to the side-walk or to the other lane in a two-way street. Due to the simplicity of the maps used for towns and traffic situations in our designed scenarios, we prohibit the agent from leaving the current lane for any reason mentioned above.

**Safe driving behavior:** Traffic rules and guidelines are mainly designed to enforce defensive driving and are the most extensive known safety measures. The prevention of collisions was the primary goal for road users and countries for decades. Therefore, many rules are designed to reduce collisions and maintain safety as much as possible. Prominent examples are right-of-way regulations together with speed limits. Ignoring or misinterpreting right-of-way rules can cause *hazardous* or *catastrophic* accidents, especially in dense metropolitan areas. Therefore, we set driving agents to remain in line and follow the regulations accordingly, such as the right of the way. Maintaining an acceptable level of safety while alleviating the number of violations is only possible by vigorous enforcement of the traffic regulations and driving rights.

### 5.1.2 Mapping

After identification of the violation, it must be marked accordingly for the following detailed investigations. In the mapping phase, we group the violations by factors of **type**, **severity**, and **location**. Clustering by type and severity is trivial, but it is

necessary to utilize a grid for the location. The map is respectively divided into tiles of a predefined size. Each violation is added to a specific tile and grouped with the other violations of this tile. The degree of safety is measured in the quantity of safety violation occurred in the situation. The quantity index (*score*) defines the relevance of each tile. Equation (5.3) represents the calculation of the respective score function. A low index ($score_s < 0$) indicates that the violation occurred rarely compared to the average and is rather unspectacular. On the other hand, a high index ($score_s > 0$) indicates a comparably complicated situation. A score of 0 indicates an average situation regarding safety violations and yet does not imply any irrelevancy. Respective weights are assigned to each severity type in order to emphasize more on the critical ones. Equation (5.1) depicts the weights for each severity.

$$m(violation) = \begin{cases} 1, \text{if severity is Negligible} & (S0) \\ 2, \text{if severity is Minor} & (S0) \\ 4, \text{if severity is Major} & (S1) \\ 8, \text{if severity is Hazardous} & (S2) \\ 16, \text{if severity is Catastrophic} & (S3) \end{cases} \tag{5.1}$$

*Note: S0, S1, S2 and S3 indicate the severity classes defined in ISO 26262* [34]

$$x_s = \sum_{v \in V_s} m(v) \tag{5.2}$$

*where $V_s$ are all violations at location $s$*

$$score_s = \begin{cases} \dfrac{\mu - x_s}{\sigma}, & \text{if } \sigma \neq 0 \\ 0, & \text{else} \end{cases} \tag{5.3}$$

*where $\mu$ is the mean and $\sigma$ the standard deviation of all $x_s$*

### 5.1.3 Visualization

An essential part of this framework is the visualization of the given mapping in the previous stage. Scores and counts of the occurred violations are calculated in tiles with the given grid size. The visualization considerably helps both the developer and safety engineer identify and understand the agent's problems intuitively. We propose to use three different types of visualization methods to enhance the overall efficiency of the framework:

- A simple text output
- A 2D map with highlights of the safety violations
- An overlay in which the identified violations can be displayed directly on the simulation environment

The main idea behind the visualization phase is to provide a high-level overview of the agent and the nature of its interaction with other road users and objects. With the help of visualization, the safety engineers can observe more involved factors contributing to the context of the situation and influencing the driving subject (instead of focusing only on maneuver planning) that may have an impact in causing the safety violation. Furthermore, this option helps the safety engineer to acquire more flexibility to provide comparably better and richer policies and adjustments concerning safety standards and enforced rules to the AI developer of the system and the target agent.

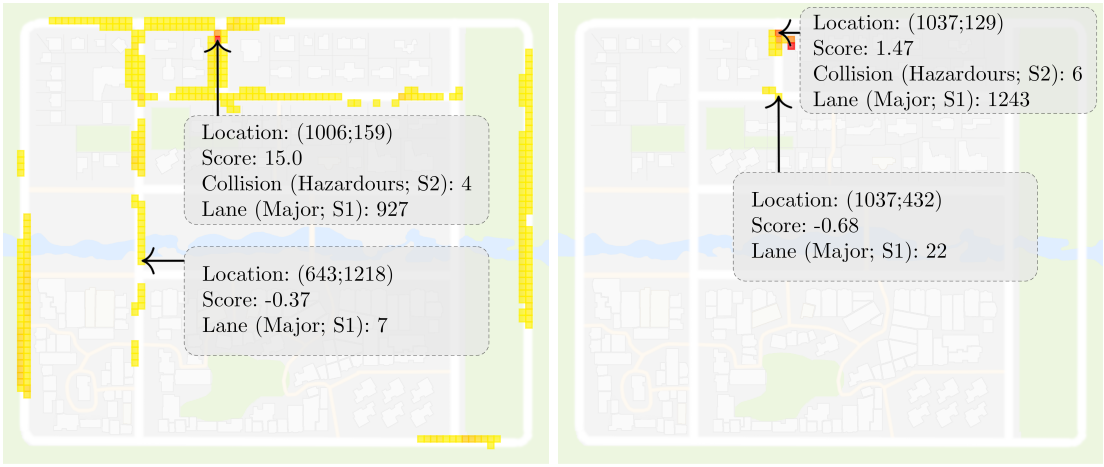
### 5.1.4 Evaluation Setup

To evaluate our framework, we compare two agents within the simulation environment of CARLA simulator [204]. The selected agents are a reinforcement learning (RL) and an imitation learning (IL) agent [205]. The reinforcement learning agent is trained as a proof of concept in the first CARLA draft. It is based on the *asynchronous advantage actor-critic (A3C)* algorithm and is trained for goal-directed navigation in CARLA. The reward is based on speed, distance to the goal, collision, and position in the assigned lane. We believe that the agent is driving at an acceptable level for an evaluation from a safety perspective. Nevertheless, the agent faces many issues, especially in navigating, and it has only limited awareness regarding other road users. The second agent is trained using *conditional imitation learning* (CIL) and is an improved version of the original imitation agent presented in the first CARLA draft. Imitation learning uses the knowledge of an expert and imitates the behavior of the same expert; a human driver in this case. As a result, this agent acts comparably much better at navigating, driving, and awareness regarding other road users. Nevertheless, this agent suffers from several limitations, such as preserving the right of way rules.

### 5.1.5 Test Environment

The agents initially are set to drive a total distance of 100km in the simulated environment with several iterations according to the preference of the safety engineer and the enforced rules. This test environment uses a distance stop criterion over time or interchangeably episodic criterion because the navigating capabilities of the agent strongly influence the episodes. We do not specify a time criterion in order to avoid punishing agents driving with higher speed. The episodes have a fixed number of critical situations (e. g. intersections), and driving slower through them will decrease the number of critical situations in total. The route is set to be straight from the origin to the destination; therefore, no advanced navigation capabilities are required. Nevertheless, the routes still contain critical situations such as intersections, pedestrians, and slower driving vehicles on the path. Situations with traffic are considered along with traffic-free scenarios for the testing. The test environment with traffic includes 100 other cars and 40 pedestrians distributed on the map. The scenarios are crowded by cars and pedestrians with this setup but excluding any stop-and-go situation or traffic jams. We apply the safety constraints based on *distance*, *lane*, and *collision* to evaluate the agents regarding safety

and testing the framework. In the traffic-free scenarios, the distance constraint is not relevant since no other cars are involved. The value for an appropriate distance is set to two seconds, as a common practice. Figure 5.1(a) represents the violations of the RL,




(a) State-Map of the RL agent

(b) State-Map of the IL agent

**Figure 5.1:** State-Map of the evaluation scenario without traffic

and Figure 5.1(b) depicts the violations by the IL agent in the traffic-free scenario. The observed collisions concentrate almost all in the same area (left side of the upper-middle street). Only 19 out of 71 violations ($\sim$ 26%) did occur outside of this area for the RL agent. The IL agent did not register any collision outside of this region. It is an indication of a problem for the agents here. The RL agent's number (and distribution) of lane violations imply a broader issue regarding lane keeping and collision avoidance matters. We assume a relationship exists between the collisions and the lane violations, but plenty of lane violations are observed without any specifically related collision. Technically driving on the wrong lane or the side-walk causes no collisions if there are no objects to collide with them. As depicted in Figure 5.2(b), the IL agent demonstrates comparably good performance, but there are several safety violations and safety-critical situations recorded during the rides of the agent. However, there are no lane violations or collisions outside the aforementioned hot spot. Eventually, according to the overall results, it is evident that the IL agent performs a *safer* drive than the RL agent with better performance in lane-keeping. Following the mapped states of Figure 5.1, Figure 5.2 focuses on the safety violations of the IL agent separated by the violation type in the scenario with traffic. The lane violations are similar to the traffic-free scenario. Most violations occurred in the same area but had a higher variance. According to Figure 5.2(a), it is notable that there is a massive increase in the number of recorded collisions. In this scenario, many violations got spotted all over the map. The safety-critical areas are like before, but several new hot spots are also considered afterward. The newly added hot spots are related to a collision with other vehicles or pedestrians. The distance viola-

tions depicted in Figure 5.2(c) indicate that most of the recorded incidents are related to rear-end collisions.

(a) Collision (b) Lane violations (c) Distance violations

**Figure 5.2:** State-Map of the IL agent with traffic

The developed framework's performance in identifying safety violations during the design-time in a simulation environment is a considerable success and one of the early attempts to form a collaborative environment for safety and artificial intelligence domains. As stated before, efficient enforcement of safety standards and rules was one of the main challenges in developing AI-based agents involved in driving tasks, especially for automated driving scenarios. This issue is caused due to the heterogeneity of the safety and AI development domains. On the one hand, most of the developers active in the artificial intelligence domain have no specific concern about the safety standards and focus more on enhancing the overall accuracy and improving the agents' performance. On the other hand, comprehending the behavior of the developments by AI experts is not an easy task for safety engineers. Most of the standards, for the experts of the safety domain, are just a set of hard-coded rules that are either followed or violated after the verification (and respectively mitigated); hence, understanding the nature and the reason of the safety violations was out of the context. This set of issues increases the inconsistency in the development chain and will eventually result in erosion during the process. The safety violation identification framework developed in this work and evaluated accordingly, as explained above, successfully demonstrates its efficient role in tackling this matter into account and facilitating the future collaboration between safety and artificial intelligence domains in developing driving algorithms and solutions for intelligent vehicle agents.

## 5.2 Runtime Safety Monitoring with CPN

The *crash prediction networks (CPN)* idea, which is already introduced in Section 4.3 is based on the monitoring concept (as known as safety envelopes) to ensure the safe operation of the developed AI-based application. One of the notable features of the safety monitoring techniques is its high flexibility and adaptability to different applications, regardless of the specific nature of the functions. This feature enables the monitors to be configured and get deployed on the target application easily. For example, consider an

end-to-end setup for the driving system with a range of sensors trained in a simulation environment. The driving module uses information about the state of the environment to decide whether to continue straight, steer, or apply brakes. The crash prediction network can be thought of as an envelope around this driving module. CPN aims to study the action decision obtained as output from the driving module to determine whether it is likely to lead to a crash given the sensory information about the state. Safety monitors need to be highly robust and reliable; thus, CPN is designed to be implemented as an ensemble of neural networks. Each network has its unique specification, either in architecture or the subset of sensory data it consumes as input [206]. The networks then work in sync to reach a consensus on whether the vehicle should continue with the currently decided action or abort and trigger an intervention. During the training phase of CPN (as depicted in Figure 5.3), a dataset is created by allowing a reinforcement learning-based driving agent to interact with the simulation environment in the CARLA simulator in order to capture and store the information of the states encountered and the actions taken along with the outcome. This step allows the training of the CPN to be posed as a classification problem with two classes of *safe* or *unsafe* states, each indicating whether the action decision with the given state information leads to a collision or not. During the operational phase (as depicted in Figure 5.4), the ensemble of networks that compose CPN observe the input from the various sensors as well as the decision proposed by the driving module to predict the *safeness* of the outcome. Suppose the proposed action is likely to lead to a catastrophic state according to the previously observed cases during the training phase. In that case, a predefined intervention is triggered, thereby filtering out potentially dangerous action decisions. Otherwise, the proposed action is executed without any interruption. The predefined intervention also referred to as the *fail-safe mode*, can either trigger other involved applications to intervene or perform a simple action like transferring the control to a human driver or even simply shutting down the engine and pulling the vehicle over to the shoulder of the road. The specific details of the intervention are out of this work's scope. The main focus here is developing a technique that determines only the *necessity* for intervention and the expected *execution time*.

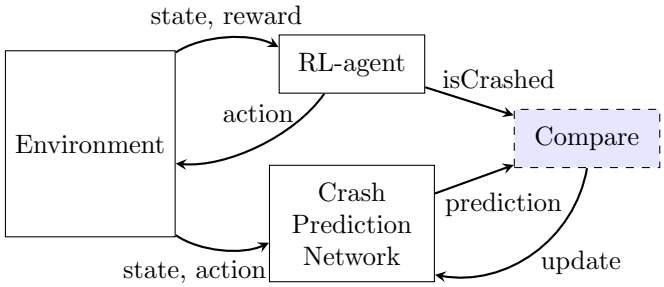



**Figure 5.3:** Training phase of the crash prediction network [7]

A potential problem that may be encountered is that CPN loses its relevance over time in the real world due to the operational environment's dynamic and constantly evolving

state. CPN setup makes it possible to train the network in an iterative manner, which can combat this issue in such cases. However, implementing iterative training with CPN would require the continuous collection of live driving data during the operational phase and training CPN on the newly collected data at regular intervals. Additionally, an advantage of this technique is that it is not tightly coupled with the nature of the driving agent. Our experiments utilize a reinforcement learning-based driving agent; one could easily swap it for any other type of driving agent based on a different sort of algorithm.

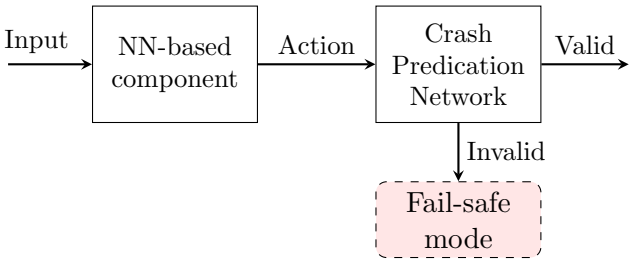



**Figure 5.4:** Operation phase of the crash prediction network [7]

Although our proposed architecture and techniques suggest using an ensemble of networks, each focusing on a subgroup of different sensors, the scope of our evaluation is limited only to visual data collected through RGB cameras mounted on the hood of the car in the simulation environment. This experiment studies the importance of accounting for temporal features in the data on the prediction accuracy of CPN by modeling two different network architectures, namely *simple CPN* and *Spatio-Temporal (ST)-CPN*, as depicted in Figure 5.5. Simple CPN uses a single frame, i. e. an image of the current state, to predict the following state's safety based on the driving module's proposed action. This goal is achieved by utilizing a VGG network [207] (trained from scratch with the dataset collected in CARLA simulator as described in the following) for feature extraction, which is then concatenated with the action decision to perform the classification. On the other hand, the ST-CPN architecture takes an N-frame of long history as input, i. e. the last N-frames encountered before reaching the current state, along with the proposed driving decision.

Additionally, the importance of accounting for uncertainty is also covered in this evaluation. Standard deep learning uses point estimates for predictions [68]. So when the model encounters inputs that are dissimilar to the ones it was trained upon, it might counter-intuitively generate a high probability score. This issue makes the probability scores an unreliable estimate of the model's confidence. However, it can be combated using Bayesian deep learning, which allows for a probabilistic approach to predictions by inferring distributions over the model parameters [69, 70]. Besides generating uncertainty estimates, Bayesian deep learning also helps to reduce over-fitting. However, such models are difficult to train and usually have intractable objective functions. Thus, in this work, we explore the need for accounting for uncertainty in our safety monitor approach based on CPN by utilizing MC-Dropout to approximate the Bayesian func-

tion. In addition to the RGB cameras placed on the hood of the experimental vehicle in the simulation environment, the networks' focus was on preventing *locally avoidable catastrophes* [208]. Such catastrophes can be avoided by adjusting the course of action when danger is imminent. This simplifying assumption eliminates the need for long-term strategic planning and focuses only on the point of failure. The experiments are conducted using *CARLA simulator version 0.9.6*. The CARLA simulator provides a *scenario runner*, which acts as an additional layer over the simulator to support the testing of driving scenarios laid out by NHTSA as a list of pre-crash typologies [209]. First, ST-CPN is compared against the single frame input of simple CPN to study the importance of temporal features in safety prediction. The evaluation in this work uses a history length of 10, meaning the last ten image frames encountered by the ego vehicle are fed as input to the ST-CPN model. Next, both the models are extended for further experiments with uncertainty by applying MC-Dropout. Following that, a dropout layer with a probability of 0.4 is applied during training and inference after each trainable layer in the models (i. e. conv, convlstm and dense).

### 5.2.1 Dataset

Creating a representative dataset is a vital part of each deep learning pipeline. Data for the experiments in this work is collected by allowing the ego vehicle, backed by a reinforcement learning agent, to drive in and interact with the simulated environment. The simulator provides pre-built environments called *Towns*. Towns 01, 03, and 04 are used for training and validation, while towns 02 and 05 were used for testing. For the initial tests of the proposed approach presented here, the scale of the experiment is relatively limited, with 18000 images (12000 safe and 6000 unsafe) used for training and 9000 (6000 safe and 3000 unsafe) used for testing.

Regarding the setup, as stated before, the ego vehicle used in the simulation environment is equipped with three RGB cameras, placed on the left, right, and center of the far front of the hood over the car. The cameras enable the ego vehicle to better perceive its surroundings by providing a wider field of view. As mentioned earlier, the two network architectures require two different formats of data. Thus, for the simple CPN model, single frames of images are stored. The RGB images from the three cameras are first converted to gray-scale to reduce the effect of color on the decision-making of the neural network. This step is essential since the network only needs to detect an obstacle, regardless of the type. The single-channel gray-scale images from the three cameras are then combined depth-wise to create a single three-channel image of dimension 84x84x3 (as depicted over Figure 5.6), such that each channel represents one of the gray-scale images. A similar procedure is followed to extend the data to the ST-CPN model by storing a concatenation of the last ten image frames per step. The ten image frames are stacked vertically to generate an (84x10)x84x3 image. The single long image is processed into a series of 10 images of dimensions before being fed to the model as input, 84x84x3 akin to a short video. The two networks perform binary classification, such that the final dense layer contains a single neuron. Thus, a decision threshold value of 0.6 is used so that if the output layer neuron produces a value greater than 0.6, then the state-action

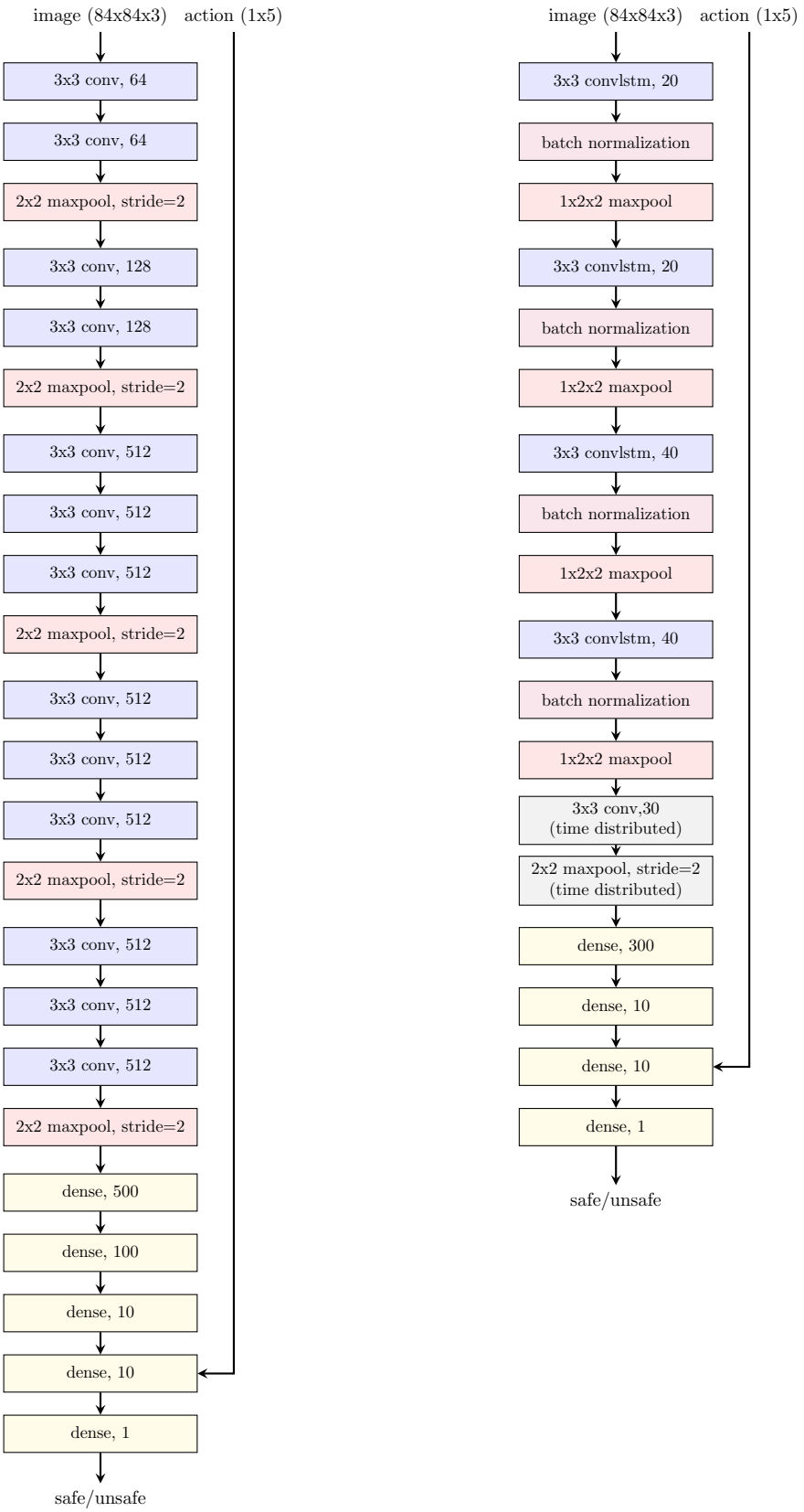





**Figure 5.5:** The Simple CPN (left), and ST-CPN (right) architectures

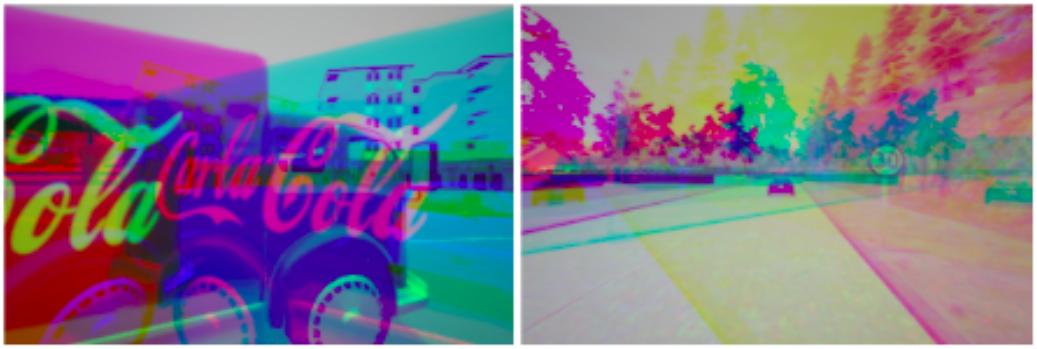


**Figure 5.6:** Format of the dataset used by Simple CPN

pair gets classified as unsafe. The test data consists of 3000 images of the unsafe class and 6000 images belonging to the safe class, of which 3000 images are those of frames that occurred just a few frames before the actual crash frame.

### 5.2.2 Evaluation Metrics

Unlike the fully controlled driving scenarios in the simulation environment, crash scenarios occur comparatively rare in the real world. This issue for evaluating systems built for ensuring the safety of autonomous vehicles often leads to the development of imbalanced datasets. This was also the case in our dataset by enforcing a mild imbalance, as explained in Section 5.2.1. Thus, *accuracy*, the most commonly used metric in deep learning-based solutions, does not suffice as it could lead to a false sense of success in an unpredictable fashion. Since falsely classifying unsafe states as safe is much worse than vice versa, the main focus of the models should be on reducing the occurrence of false negatives in the prediction pipeline. Therefore, *recall* is a more valuable metric for the CPN models, which have been captured via *precision-recall* curves. As stated earlier, we have designed two types of CPN, namely *simple* and *Spatio-temporal* (as known as ST-CPN), considered for evaluation under the presence of both static and dynamic objects on the road in a simulated environment. Besides, we also consider an *uncertainty* estimate in some of our experiments to demonstrate the efficiency of the crash prediction networks in dealing with this challenging phenomenon.

### 5.2.3 Simple CPN with Static Obstacles Only

Before moving on to scenarios with a complex environment and more involved factors, it is necessary to test whether a deep-learning-based model could help to predict the possibility of a crash based on state and action information. As a sanity check in this regard, the simple CPN model was tested in the initial experiment with only static obstacles. This means that potential crashes are limited only to walls, fences, rocks, crates on the road, and other static objects in an urban scenario, excluding the objects

in motion like pedestrians and vehicles on the road. In this situation, the evaluated model can predict the crash situations with a test accuracy of 0.7907 and an accuracy-precision score of 0.7136. However, this accuracy rate is inadequate to be practically usable.

### 5.2.4 Simple CPN with Dynamic Obstacles

Following the initial experiment results, the simulation environment for the new experiments is extended to include dynamic obstacles in the form of 2- and 4- wheeler vehicles on the road. The simple CPN model is tasked with taking an image of the current state as input and the proposed action decision to predict whether the next state would be *unsafe*. The developed model in this experiment can replicate the success of the previous one, with a minor enhancement and achieving a test accuracy of 0.8018 and an AUC-PRC score of 0.7624.

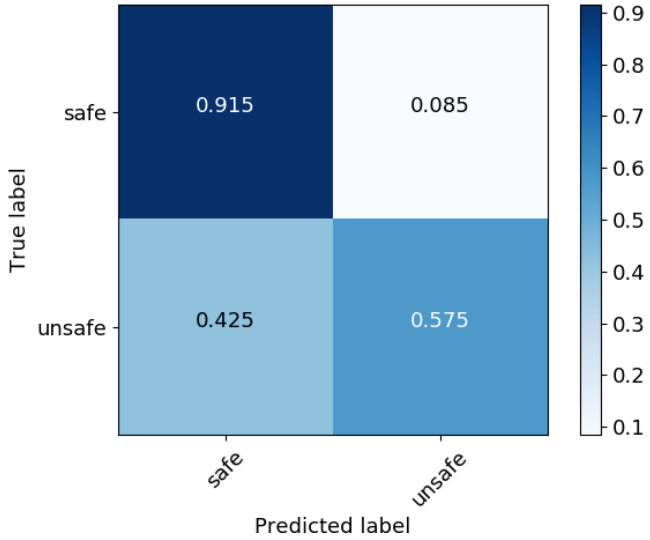



**Figure 5.7:** Simple CPN model on test set with dynamic obstacles

As depicted in the confusion matrix over Figure 5.7, the number of false negatives is considerably high. However, it is vital to minimize false negatives in the prediction outcome for safety-critical systems such as autonomous vehicles and their applications involved in maneuver planning. In order to improve the classification results, class weights are introduced in the simple CPN architecture. The class weights are used during training in the ratio of 1:2, such that the penalty of wrongly classifying a crash case applied to the loss is double that of the penalty for wrongly classifying a non-crash case. Depicted results over table 5.1 demonstrate a slight improvement of overall accuracy and an increased recall score of simple CPN after applying class weights.

### 5.2.5 ST-CPN with Dynamic Obstacles

Despite the presence of dynamic objects in the simulation environment, the simple CPN made its decision based on only a single frame of information. This feature does not allow the network to model the motion of the ego vehicle and other obstacles in the environment. Therefore, the ST-CPN model is introduced to deal better with moving objects, using Conv-LSTM to process a contiguous series of 10 frames of images. By comparing the results as depicted in Figure 5.8 and Figure 5.7, it is evident that the ST-CPN model can perform considerably better in identifying *unsafe* situations, thereby reducing the number of determined false negatives as desired. This enhancement was further visible in comparing the results of the ST-CPN model against the simple CPN model in table 5.1 on the test dataset. Although the performance is comparably better, the increased complexity of the model leads to an increase of the inference time for ST-CPN by a factor of 10 compared to simple CPN.

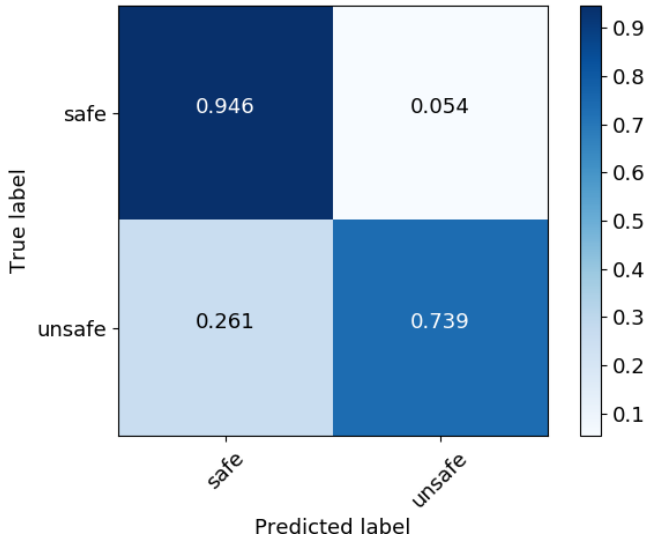



**Figure 5.8:** ST-CPN model on test data with dynamic obstacles

Following the comparably good performance of the ST-CPN model in the presence of dynamic obstacles, another evaluation is performed on the model's reaction to out-of-distribution data, namely the data that is slightly different from the conditions that the model was initially trained on it. For this purpose, the performance of the model is studied on the small test sets from previous experiments, referred to as *test small* with 3000 images randomly sampled from the original test set, and the other *test rainy* with data collected in the same town but during rainy conditions. As depicted in table 5.2 and Figure 5.10, the rainy condition causes the model to misclassify comparatively more *unsafe* scenarios, leading to a drop in the model's overall performance.

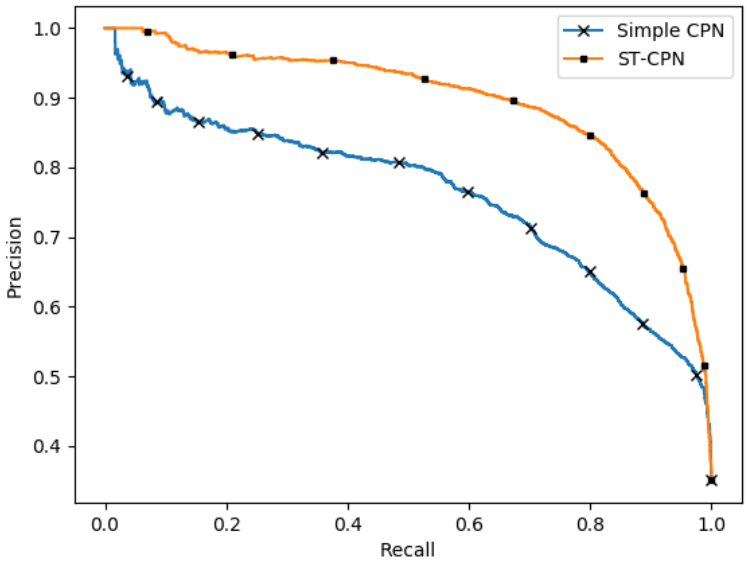



**Figure 5.9:** Simple CPN Vs. ST-CPN model on the test set with dynamic obstacles

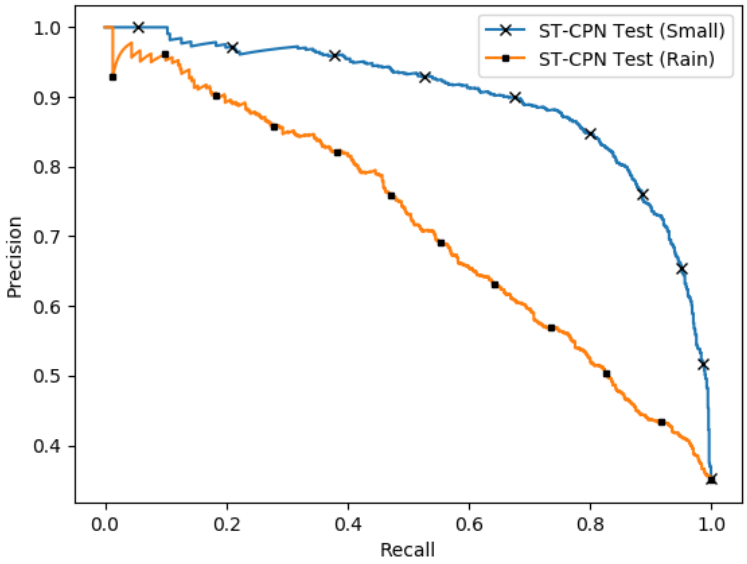



**Figure 5.10:** ST-CPN model on test set in different weather conditions

### 5.2.6 Simple CPN with Uncertainty in Dynamic Environment

As discussed earlier, accounting for temporal features in the prediction model helps to improve the performance; however, it suffers from a drop in performance when encountering out-of-distribution data. Since the real world is constantly evolving and cannot be completely modeled in the training data, it is necessary to have techniques for the developed models to deal with such data. As pointed out earlier, standard deep learning solutions provide no information about the model's confidence in its prediction outcome. Thus, both models from previous experiments are extended with MC-Dropout, by placing a dropout layer after every convolutional and dense layer, except the output layer. The dropout was then applied during training, and during the testing phase, wherein each input was used to generate $T$ predictions to calculate the mean and variance. The variance on each observation/data point indicates how confident the model is in its prediction, which shows how similar the test data is to the training data. Since performance improvements of using class weights are negligible, the Bayesian version of the simple CPN model was trained without class weights. The advantages of using uncertainty become apparent when using a dataset that differs considerably from the training data. Thus, the performance of Bayesian simple CPN is evaluated on both *test small* and *test rainy* set from the previous experiment. As depicted in table 5.2, the test set, including the clear weather conditions, has nearly similar uncertainty estimates as the main training set; however, using the test set, including the rainy weather conditions, increases the uncertainty estimate. It is fascinating that a minor change as small as a variation in weather conditions can increase the uncertainty of the predictions. The uncertainty estimates can therefore help build trust in the prediction of the CPN models. However, the benefit of estimating the model's confidence in its decisions comes at the cost of a significantly longer inference time. The developed model takes ten times more time to compute the class labels and their corresponding confidence values during evaluations.

| Type | ACCURACY | RECALL | PRECISION | AUC-PR | Note |
|---|---|---|---|---|---|
| Simple CPN | 0.8018 | 0.57 | 0.77 | 0.7624 | - |
| Simple CPN | 0.8131 | 0.68 | 0.74 | 0.7706 | with loss adaption |
| ST-CPN | 0.8773 | 0.74 | 0.87 | 0.8951 | - |
| Bay. Simple CPN | 0.8015 | 0.57 | 0.77 | 0.9740 | Uncertainty: 0.0162 |
| Bayesian ST-CPN | 0.8281 | 0.63 | 0.71 | 0.7679 | Uncertainty: 0.0166 |
| Bayesian Combined | 0.8328 | 0.62 | 0.71 | 0.5408 | Uncertainty: 0.0222 |

**Table 5.1:** Comparison of classification metrics on the test set in clear weather

### 5.2.7 ST-CPN with Uncertainty in Dynamic Environment

The ST-CPN network is extended by MC-Dropout to evaluate the effect of uncertainty estimates, combined with the benefits of modeling temporal features. Respectively, both models are trained to have a similar validation accuracy of about 0.80 to ensure that the

model is comparable to the Bayesian version of the simple CPN model of the previous experiments. Additionally, to capture the essence of the proposed CPN model with multiple independent neural networks, the outputs of the *Bayesian simple CPN* model and the *Bayesian ST-CPN* model are combined as a weighted average with a higher weight being assigned to the latter. Hence, this model is referred to as the *combined* model. As depicted in Figure 5.11, the combined model performs slightly better than both individual models regarding the performance over a range of probability thresholds. Thereby it demonstrates the potential benefit of modeling CPN as an ensemble of diverse networks functioning in the union to make a robust prediction about the future state.

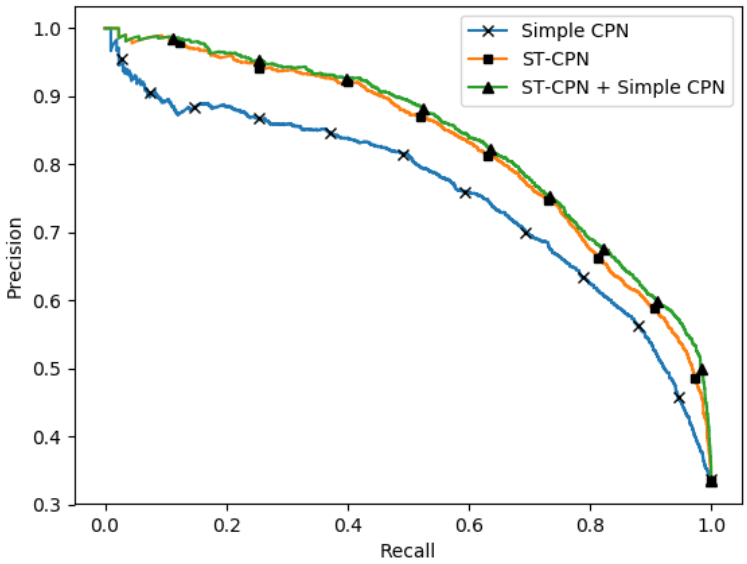



**Figure 5.11:** Bayesian versions of simple CPN, ST-CPN model and a weighted combination of two models

### 5.2.8 Simple CPN and ST-CPN Models in Live Simulation

In order to demonstrate the performance of the Bayesian versions of the simple CPN and ST-CPN on extrinsic evaluations, both the models are plugged into the ego vehicle in CARLA to simulate a real-world scenario which is adjusted to be executed for 50000-time steps. As noted earlier, the detection of safe and unsafe states suffers from an extreme imbalance in datasets; hence the ego vehicle experiences only 246 collisions in its lifetime during the experiment. On the other hand, the ST-CPN model can detect 147 cases for these recorded collisions successfully, while the simple CPN success rate is only 140 cases. Considering the recall as the most important metric for our evaluation here, the ST-CPN model outperforms the simple CPN model. The poor precision scores of the model can be attributed to the highly imbalanced nature of the test data. Figure 5.12

| Type | ACCURACY | RECALL | PRECISION | AUC-PR | Note |
|---|---|---|---|---|---|
| ST-CPN | 0.8816 | 0.75 | 0.88 | 0.8983 | test set |
| ST-CPN | 0.7770 | 0.45 | 0.79 | 0.7140 | rainy test set |
| Bay. Simple CPN | 0.9443 | 0.90 | 0.93 | 0.9740 | U*: 0.0139, training set |
| Bay. Simple CPN | 0.8030 | 0.57 | 0.78 | 0.7679 | U: 0.0163, test set |
| Bay. Simple CPN | 0.6173 | 0.69 | 0.45 | 0.5408 | U: 0.0212, rainy test set |

**Table 5.2:** Comparison of classification metrics in clear and rainy weathers - U*: Uncertainty

provides a better understanding of the relation between the precision and recall of the two models.

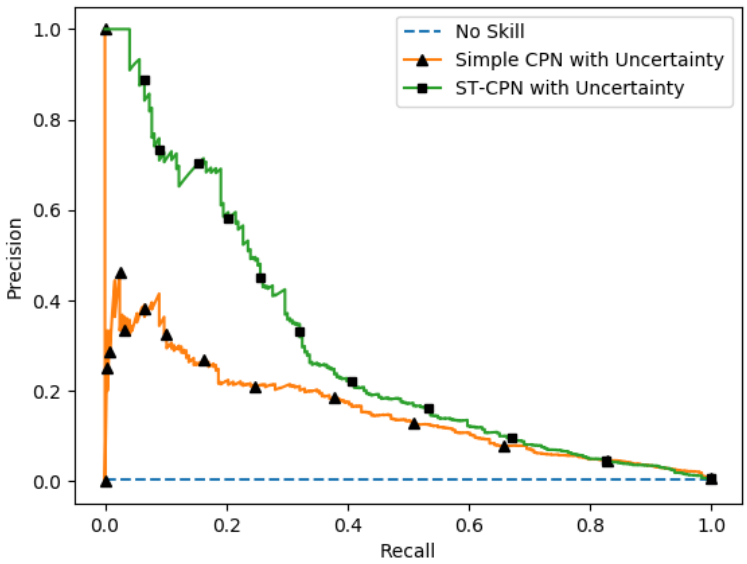



**Figure 5.12:** Bayesian versions of simple CPN model and ST-CPN model on a extrinsic evaluation performed by plugging the models directly into simulation

## 5.3 Empirical Study on Emotional Profiles

The occupants' behavior in the cabin environment (especially the driver) is directly influenced by their emotional states originating from internal and external stimuli. Those exposed changes in behavior are usually reflected in their interaction with in-cabin comfort applications and driving components such as entertainment systems or steering wheel and pedals of the vehicle. It is also known that emotions directly impact the intensity of steering wheel rotation and pressure on the gas and brake pedals, positioning the hands, head movements, changes in eye gaze, and maintaining the distance to other vehicles [210]. However, the true nature of the behavioral reactions in response to the changes in emotional status is unclear and could differ from subject to subject. One argument is that a positive change in the arousal of an emotional state may lead to more frequent sudden hands and head movements, like speedy eye gazes and/or comparably faster and sharper steering wheel maneuvers. Other factors, such as ethnicity and cultural differences, add to the complexity of the matter concerning identifying in-cabin behavioral-based emotional patterns. Therefore, performing a thorough empirical study on the driver's behavior in an in-cabin environment and the effects of different emotional states on the interaction between the subject and the driving components is beneficial to validate the existing *general* assumptions and form the common ground. For this purpose, surveys and questionnaires are great tools to establish a baseline for the respective empirical study; hence we prepared and distributed an online survey that consists of three main parts as depicted in Figure 5.13:

- The opening questions, with a focus on demographics in order to collect the basic information regarding the participants, such as age, gender, and origin (as known as ethnicity),
- Designed scenarios following narration of a pre-history situation on the road that aims to induce positive and negative emotions in the subject,
- General questions, which aim to collect the personal perspective of the subject regarding the overall idea of increasing the autonomy of the vehicle, delegating the control to the vehicle as well as the impact of different emotions on their driving behaviors,

In the first round of our study, in which we published the preliminary findings over [211], 103 people in total participated, 95 of them were between 20 to 30 years old, and eight people were more than 30 years old. 67% of all participants were male, and 33% identified themselves as female. Respectively, 85 of 103 who participated were drivers in Germany, 11 from Spain, and the seven remaining participants were from Azerbaijan, Turkey, UK, Poland, and South Korea. This initial phase helped us in developing user profiles which we utilize in our following experiments in table 5.3. In the second round of our survey, most of the participants in the extended version were between 20 to 29 years old, and the second largest group belongs to the range of 30 to 39 years old with a 13% distribution. The remaining participants were spread in the spectrum of 18 to

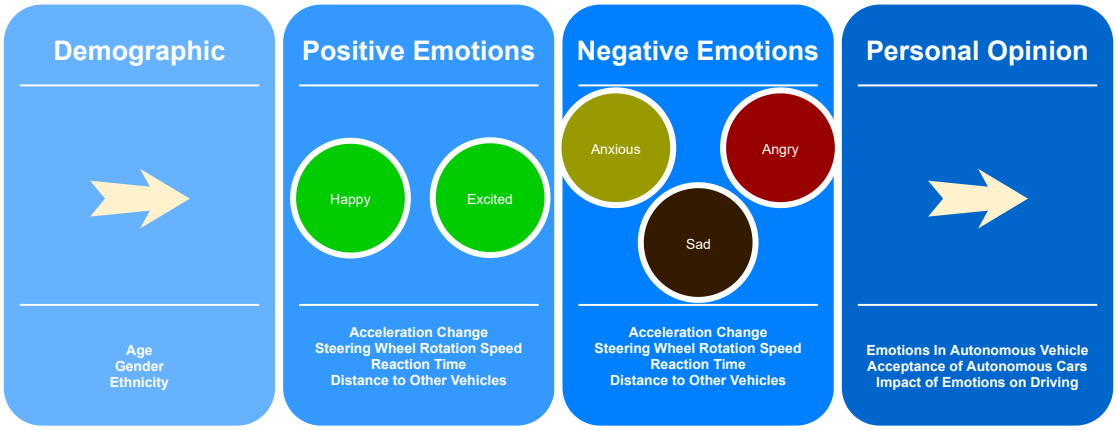



**Figure 5.13:** The overall structure of the questions in the survey

above 60 years old. Among all the participants, 43% were female, and 57% of the participants identified themselves as male. Regarding the origin of the participants, we divided them continent-wise, and the majority belong to Europe with 50%. The second leading group concerning ethnicity is Asia, with 43% of the participants. We gladly had participants from Africa, south and north America in our survey, but their participation rate remained below 5%.

The emotions understudy in this survey are grouped and averaged as positive (containing *happy* and *excited* emotional states) and negative (containing *anger*, *anxiety*, and *sadness* emotional states). The overall distribution of effects for these groups of emotions on the targeted metrics regarding the driving behavior of distance to leading vehicle, steering wheel maneuvers, reaction time, and acceleration intensity is depicted in Figure 5.14. The negative group of emotions demonstrated a comparably more significant influence on the metrics mentioned above, although the positive emotions still can not be overlooked. Despite having a lower impact than the negative group, they are essential in affecting and changing driving behavior. According to the participants of our study, the averaged impact of different emotional groups on driving metrics, as represented over Figure 5.14, supports the earlier claims on the considerable influence of negative emotions in triggering abrupt movements or actions that are highly reflected in interaction with driving components. It also explains why most of the research in the domain of safety and developing safety-critical applications is focused largely on utilizing negative emotional states. On the other hand, the studies focusing on the comfort domain mainly utilize positive emotional states.

Another important aspect of our study is evaluating the participants' perspective regarding the discrete emotions (instead of grouping them) and the individual impact of each one of them on in-cabin and driving behavior. As depicted in Figure 5.15, anxiety demonstrates comparably the highest impact on changing the status quo in the in-cabin environment and affecting driving behavior. It also shows a different level of activeness among the emotions belonging to each group concerning their reflection in behavior,

| Categories | Number | Percent |
|---|---|---|
| **Gender** | | |
| Male | 193 | 57% |
| Female | 144 | 43% |
| **Agegroup** | | |
| under 20 years | 15 | 4% |
| 20 - 29 years | 266 | 79% |
| 30 - 39 years | 45 | 13% |
| 40 - 49 years | 5 | 2% |
| 50 - 59 years | 4 | 1% |
| aboce years | 2 | 1% |
| **Ethnicity** | | |
| Africa | 12 | 4% |
| Asia | 146 | 43% |
| Europe | 169 | 50% |
| North America | 7 | 2% |
| South America | 3 | 1% |

**Table 5.3:** Demographics of 337 participants in empirical study

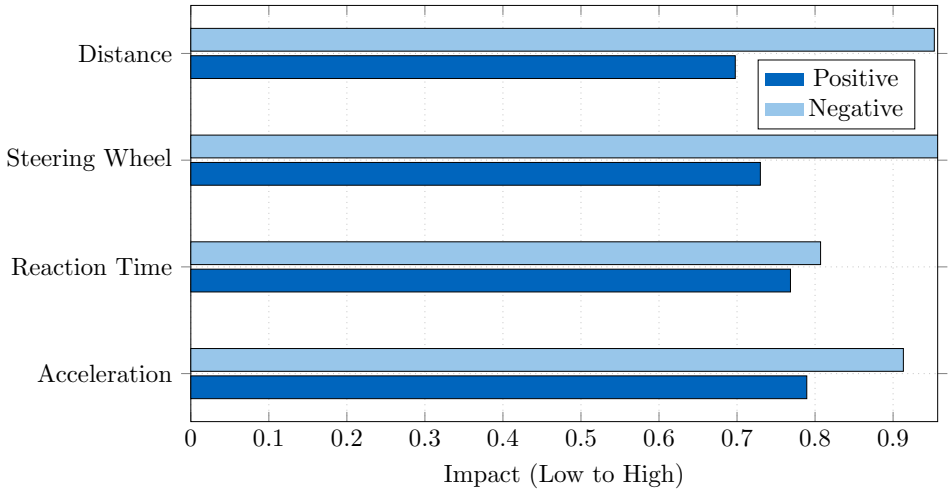



**Figure 5.14:** Average impact of positive and negative groups of emotions on driving metrics

which can be explained from the perspective of arousal-valence measure. For example, in the negative emotional group, the sadness stands a bit lower than anxiety. On the other hand, in the positive emotional group, excitement leads to more changes in driving behavior than happiness. Aside from these findings, previous studies mainly focused on evaluating the impact of anger rather than anxiety since the common term of *road rage* is believed to be derived originally from anger, even though the anxiety itself could result in the same effects. On the other hand, participants in our study might not imagine themselves totally in a *realistic* situation that leads to the inducement of anger and triggering angry emotional status, especially through a written scenario exposed to them. However, they might be able to imagine themselves in a tense situation instead, since the anxiety occurs more frequently than anger while driving due to the dynamically changing and complex environment around the vehicle. The personal opinion of the participants, which are depicted in Figure 5.16, represents the existence of such assumptions among them, as well as the overall high impact of negative emotions on their driving behavior compared with the positive emotions.

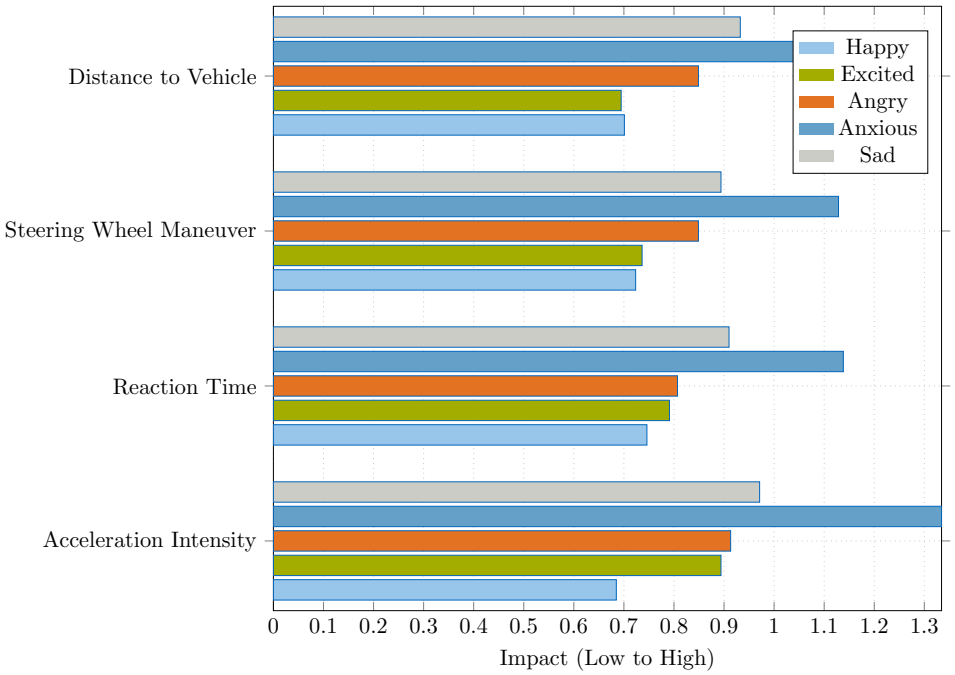



**Figure 5.15:** Impact of discrete emotions on changing driving behavior over driving metrics

One of the interesting and yet very challenging topics in automated driving is the moments in which the control of the driving task must be switched between the driver and the car. This phenomenon, in general, is referred to as taking over control by the issuance of *take over request* (TOR) and raises new challenges, especially regarding the integration of safety measures, due to its complexity and the variety of the factors affecting it [212]. Therefore, in our study, we also asked the participants about their preferences regarding this issue. According to the responses of the participants, as

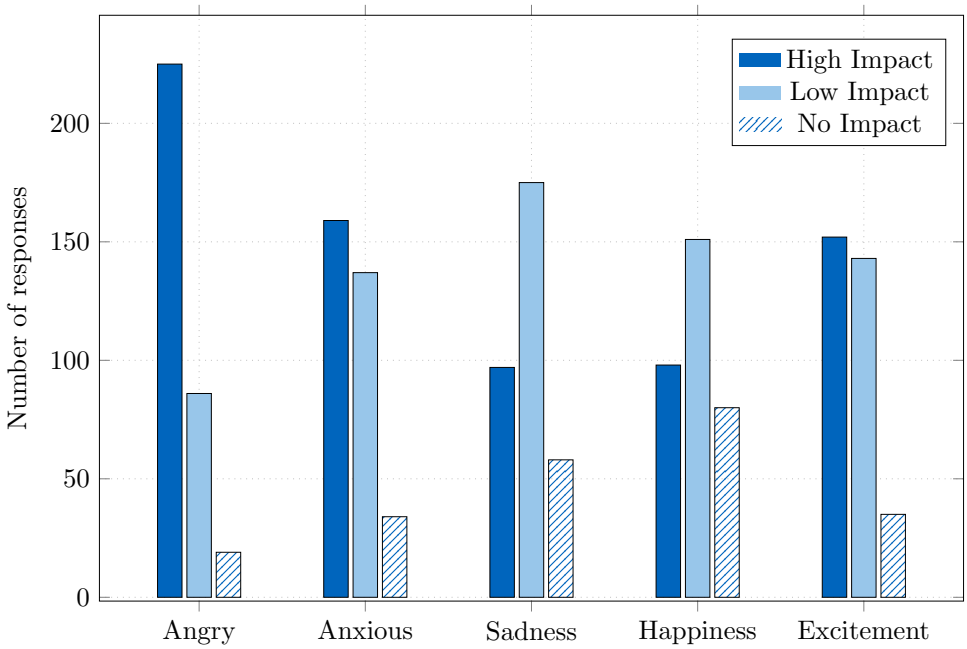



**Figure 5.16:** Personal opinion of the participants regarding the effects of emotions on their driving behavior

depicted in Figure 5.17, most of them are willing to take over the driving task in the vehicle when they find themselves in positive emotional states. On the other hand, the general preference in negative emotional situations is mainly toward delegating the driving tasks to the vehicle.

One of the crucial factors in analyzing the risk factors that lead to safety-critical situations is the reaction time under the direct influence of the driver's attention and emotional status [213]. The outcome of our empirical study, as depicted in Figure 5.18, outlines a set of interesting facts in this regard. The majority of the participants emphasized that they perform slower braking when they find themselves in a sad emotional status. Respectively, the anxiety triggers comparably faster reactions according to the participants. The affected behavioral aspects of the driver can also reflect in external factors of driving, such as the preserved distance to other vehicles on the road and maintaining it according to the contextual changes [214, 215].

As is depicted in Figure 5.19, the majority of the participants tend to maintain comparably longer distances to other vehicles on the road when they are in positive, primarily happy, emotional states. They also tend to reduce and maintain shorter distances when they find themselves in situations with the probability of increased anxiety. Similar behavior is also observed concerning acceleration intensity during the driving scenarios, as shown in Figure 5.21. Another driving module in close and continuous coordination with the driver that bears the emotional changes of the driver is the steering wheel maneuvers and changes in its angular velocity [216]. It is especially important in the presence of

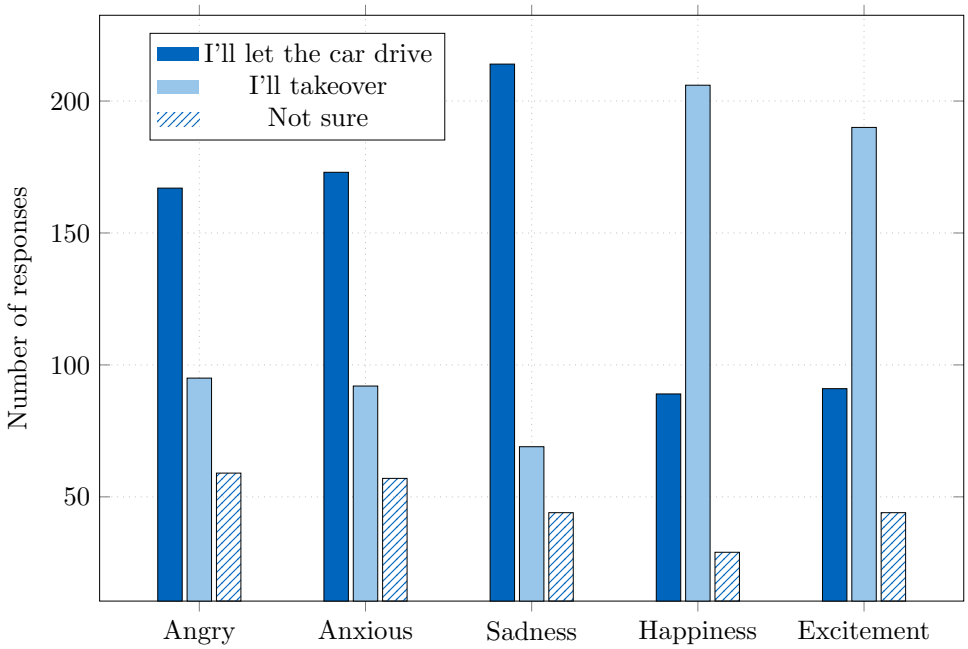



**Figure 5.17:** Personal opinion of the participants on taking control of the highly automated vehicle in different emotional state


**Figure 5.18:** Reaction Time

negative emotional states with high arousal that may distract the driver from driving task and lead to abnormal, hence sharper steering of the vehicle, as it is also observed and approved by the majority of our participants according to Figure 5.20.

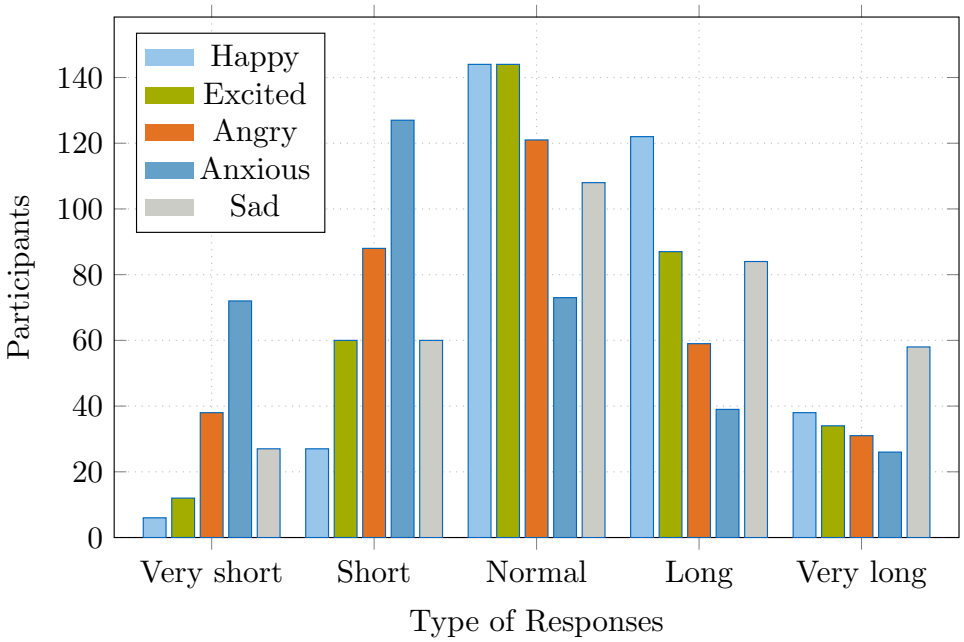



**Figure 5.19:** Distance

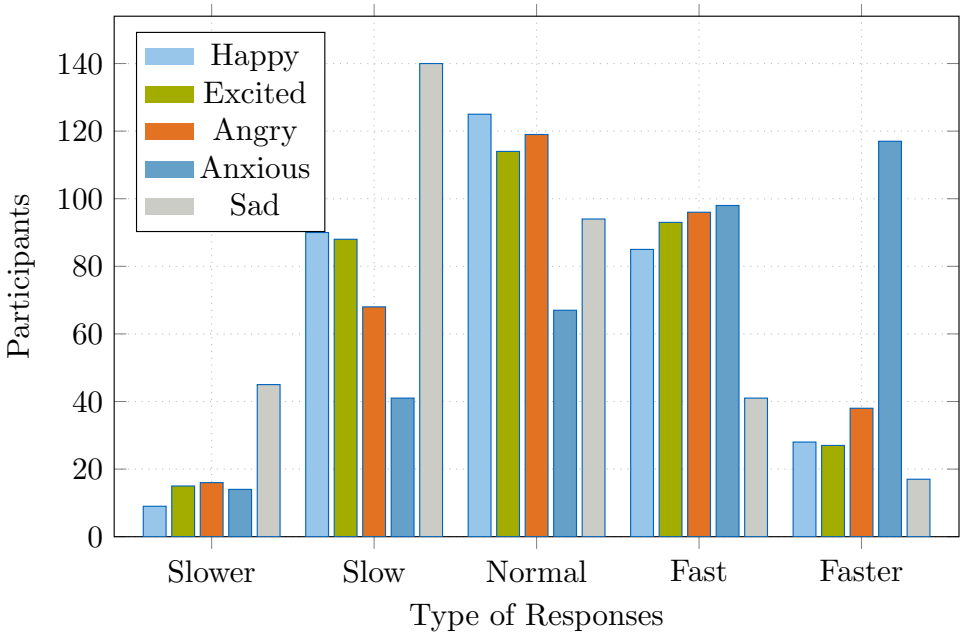



**Figure 5.20:** Steering Wheel

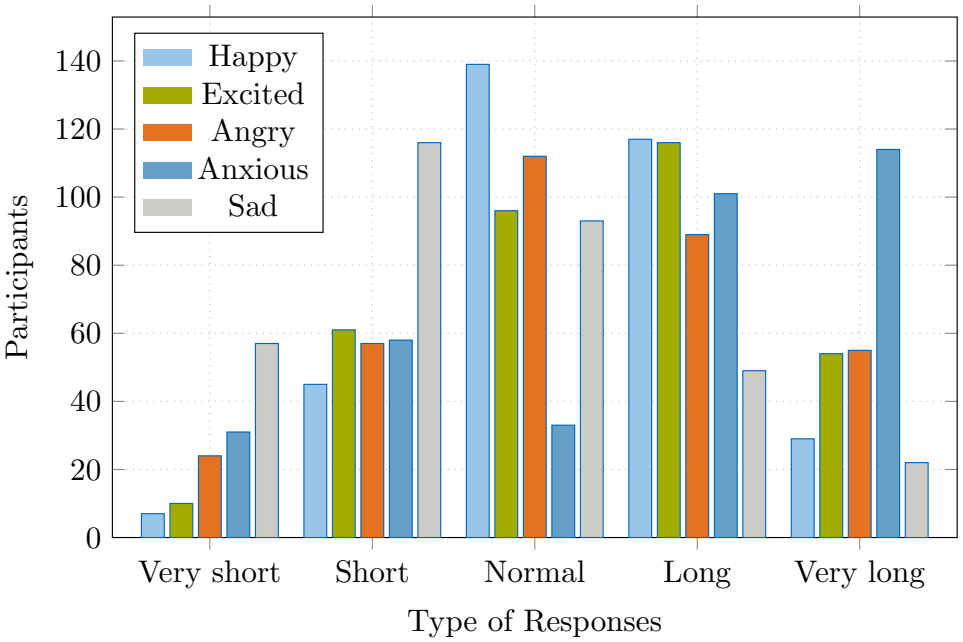



**Figure 5.21:** Acceleration

## 5.4 Experimental Driver Simulator Setup

In order to investigate the relationship between the driver emotions and in-cabin behavior and respectively design a multimodal recognition architecture, we extended our work to the lab environment. We set up a real-life driver simulator to populate a corresponding dataset containing in-cabin behavioral signals. Following the perspectives acquired in our study at Section 5.3 and with the help of the experimental setup, we can induce certain emotions and trigger the respective emotional state through this highly controlled environment of the simulator and effectively track the subject's responses to any desired and triggered stimuli. The subjects' personality plays a vital role in inducing the emotions; hence we integrate a brief questionnaire between each driving scenario to incorporate the direct input of the subject in the creation of the dataset and the prediction pipeline.

Our experimental simulator car (as depicted in Figure 5.22) is directly connected to the simulation software provided by VIRES Simulationstechnologie GmbH, called v-SCENARIO v-TRAFFIC [217]. The VIRES *Virtual Test Drive* (VTD) simulator environment is fully functional integrated into an original *SMART* vehicle, all internal and external lights are functioning, and surrounding screens visibly adjust the exterior view. This option was an essential factor in order to maintain a realistic environment for the participants. We connect the internal bus system to the simulation software and forward the relevant signals of the steering wheel and acceleration/braking pedals to a central storage unit in order to be utilized later on in populating the desired dataset. Furthermore, we place two surround sound speakers in the back of the cabin to replicate the

natural engine's sound, the environment's noise, the sound of other vehicles on the road, and the multimedia entertainment system for boosting the desired emotional states. As depicted in Figure 5.22(a) the simulator's interior contains no visible modifications to preserve the originality of the driving environment for the subject. We set the car's driving mode to the automatic transmission, meaning the vehicle automatically shifted the gear ratio instead of being changed by the subject. Like an ordinary gear system, the gear handle inside the cabin can only be used to change the car's direction, to drive forward or backward. Nevertheless, this option was mainly excluded from our experiments in the data collection process. In order to maintain the isolation and comfort level of the environment for the subject, we activate the internal air conditioning system as well. The dashboard of our simulator is fully functional and represents the current velocity of the vehicle in the simulated world. The subjects are instructed to use the blinkers and the honk when required or requested by the operator during experiments. The steering wheel has a built-in vibration system that makes it harder to steer depending on certain factors like the velocity or braking intensity on the road. A standard RGB camera is placed directly in front of the subject inside the cabin to simultaneously record the driver's facial expressions with car signals. The provided tools and libraries of the simulation software make it possible to define autonomous traffic, deterministic traffic, events and triggers, and pedestrians in the simulation environment.


(a) The interior area of the simulator testbed

(b) The 180 degrees surrounding front area of the simulator with high resolution screens

**Figure 5.22:** The VIRES VTD simulator testbed

### 5.4.1 Driving Scenarios

Different scenarios are designed to be deployed in the simulator to induce the participants' positive and negative emotions. From a high-level point of view, a *base scenario*, an *aggressive scenario*, and a *depressive scenario* are created for this purpose. In order to measure changes in driving style, we use the same driving path in each scenario and keep the road map identical because the driver behavior is highly dependent on the

route, and preserving it, will assure a common baseline of reactions in different scenarios. We only alter specific environmental settings in the simulation to induce the target emotions and leave the other driving factors intact. Each scenario lasts approximately 15 minutes. The flag annotated as *HomeToWork* is the starting position for the participating driver, circled by a lighter green circle over the visual overview of the driving map in Figure 5.23. The order of the streets for driving is numbered as a meaningful path for the subject during each ride. The path starts outside of the city on a freeway and reaches and goes through a city afterward. This path includes six right-turns, seven left-turns, and 11 straight-ahead at intersections. The annotations in blue on the map represent coded event triggers. These triggers are placed with an absolute position before each intersection or turn and execute the code if a specific event occurs. In this case, the triggers execute the code directly to display a text on the screen, as navigational instructions to lead the subject via the dedicated path. These notifications are activated if the participant drives next to them within a radius of 10 meters.

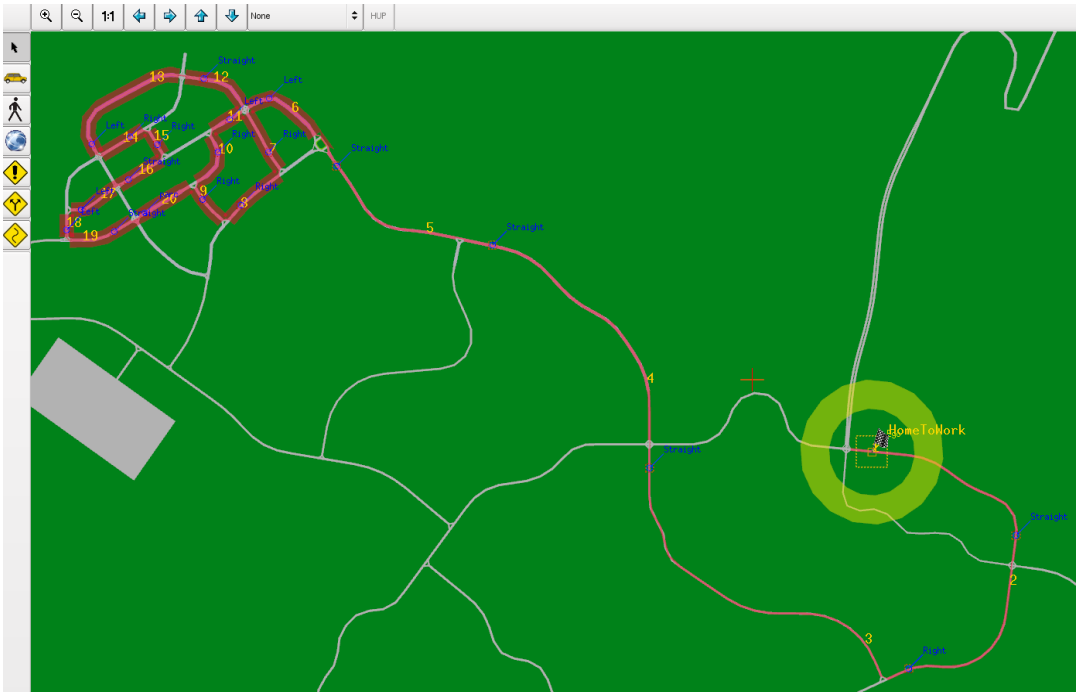



**Figure 5.23:** The map of the simulated environment for the driver

In order to measure the changes in driver behavior, it is necessary to implement (autonomous) traffic in a simulation environment that makes interactions with the subject on the road. It also allows us to analyze how the subject interacts with other vehicles and road users under certain emotional statuses. The circled area in Figure 5.23 around the subject is the area in which autonomous traffic is generated. This radius remains the same during all scenarios. If an autonomous car drives outside the radius, it disappears and appears again in the gray zone. There are three different configuration possibilities

for the generated traffic. First, it is required to define driver profiles that represent different driving styles of autonomous cars. Second, it is required to define the specific type of cars representing a traffic flow. Furthermore, we define the radius in which the cars spawn and the direction in which the cars are being driven. Considering our findings from Section 5.3, we implement three driver profiles for the traffic: the *hasty driver profile*, the *brisk driver profile*, and the *comfortable driver profile*. Each profile differs in desired velocity, acceleration, observing speed limits, distance keeping to other vehicles, lane keeping, the urge to overtake, lane change dynamics, the acceleration in curves, and the response to tailgating. The hasty driver has very high values in all of these variables, the brisk driver has medium values, and the comfortable driver has low values, respectively. Each car type is equipped with its maximum velocity and acceleration value; therefore, it is possible by chance that a motorcycle with a hasty driver profile drives slower than a car with a brisk driver profile. It is also possible to define the distribution of different types of vehicles such as cars, trucks, buses, and bikes. table 5.4 shows the distribution in the percentage of the car types in our defined traffic for the simulations. We observe that buses and trucks are too large for the city, and if they match with the hasty driver profile, they tend to create crashes in increasing numbers. Therefore, we get rid of buses from the simulation and only deploy trucks in this category. Bikes can be mopeds and motorcycles. The mopeds have a speed limit of driving up to 40 km/h. Eventually, we define how the traffic is generated around the subject. We choose the radius of 250 meters for the outer area and 150 meters for the inner radius, such that the subject does not recognize the sudden appearance or disappearance of traffic. In total, there are 26 cars in a radius of 250 meters at all times, which from our point of view is an *acceptable* traffic distribution and, at the same time, does not lead to any congestion. Regarding the direction of the approaching vehicles, 40% of them appear in front of the subject, 10% in the back of the subject, 25% on the left side, and 25% on the right side. 65% of the cars drive in the same lane, and 35% appear in the opposite lane. Our experiments demonstrate that more cars on the same lane lead to more touchpoints with the subjects. The presented settings for traffic are used in all scenarios.

| **Type** | **Probability** |
|---|---|
| Cars | 60% |
| Vans | 15% |
| Buses | 0% |
| Trucks | 5% |
| Bikes | 20% |

**Table 5.4:** Distribution of different vehicle types on driving route during the rides

### 5.4.2 Developed Multimodal Database

The evolved database consists of the face recordings of the driver through the camera, driving signals (namely steering wheel rotation and acceleration intensity), and direct questionnaire input, as depicted in Figure 5.24.

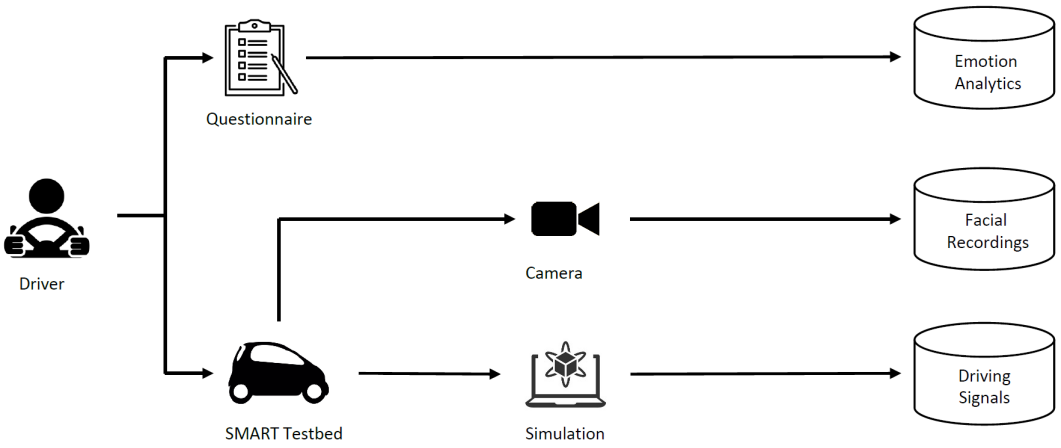



**Figure 5.24:** Different type of signals collected through the simulation testbed

The questionnaire is in the form of multiple questions analyzing how the subject personally felt during each ride and in total in each scenario. Our second data source is the camera placed inside the cabin of the testbed in front of the driver. This RGB camera records the subject's face while driving and captures the facial expressions during each ride in different scenarios. Our third data source is the car signals generated as the subject interacts with driving components during each ride. The database resulting from this experiment includes 2430000 data points of driver signals and facial recordings of around 11.25 hours and the data from the questionnaires of 15 subjects. In the end, we experimented with 25 participants and successfully recorded the data of 15 subjects in total for all the designed phases and scenarios. The experiment of 10 participants failed due to either hardware flaws in our setup or the subjects' inability to finish the rides. Each participant was recorded during three different scenarios, each lasting around 15 minutes. Thus, we have collected approximately 45 minutes of recordings from each subject. Our setup collects driver signals in 60 frames per second, leading to a data point of 90 bytes around every 16.67 milliseconds. After a preliminary pre-processing, this leads to approximately 162000 data points for each participant. The limitations of this experiment are the difficulty of handling different personalities and certainly a 100% adequate inducement of emotions. Many different emotional states can potentially affect driving and related behavior (e. g. boredom, sadness, excitement). The difficulty of considering these additional emotional states goes back to their complex nature in being produced and measured. Induction of some emotional states can be dependent upon factors outside of the driving environment, as it is, for instance, the case of fatigue.

It can also be challenging to distinguish the difference between emotional states, e. g. between angriness and stress. Besides the unobserved variables, we believe that there are more passive emotions with a high impact on the driving and in-cabin behavior, like tiredness, which are hard to induce. However, this set of experiments can undoubtedly be extended by different scenarios and more detailed emotional groups. Therefore, this study limits the observed variables that impact driver behavior in driving scenarios to only three classes of emotions, as depicted in Figure 5.25.

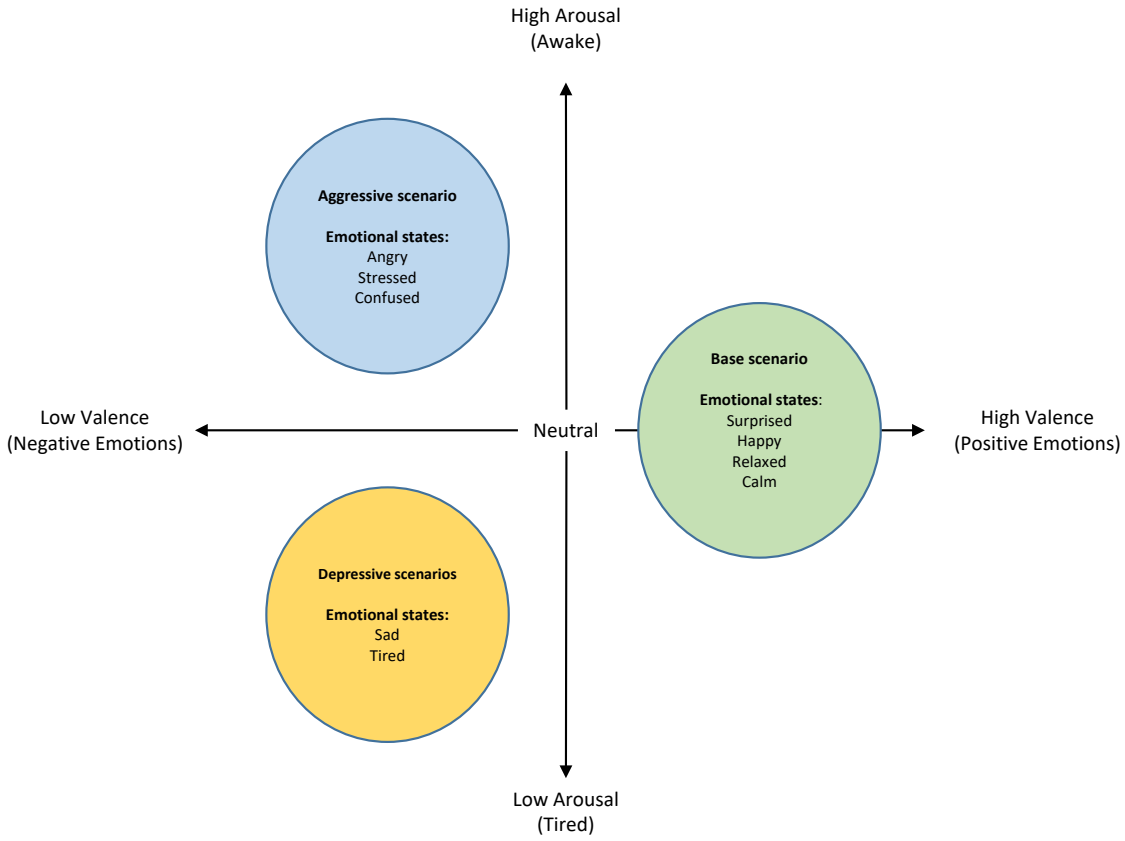



**Figure 5.25:** Mapping the driving simulation scenarios into levels of arousal-valence

Drivers may have different resistance levels to becoming agitated while driving. The difficulty of studying the effects of emotions on driver behavior lies in the fact that emotions are internal matters experienced inside a person. Only the person him/herself can describe it well, how it feels, and up to what degree its impact was. We approach this limitation by aggregating our database with the questionnaire inputs. Our questionnaire after each driving scenario provides an insight regarding the success of the inducement of the target emotion. Nevertheless, this only incorporates the personal belief of the subject. Therefore, the motive is to increase the accuracy and robustness of the data collection process as much as possible. According to the general feedback of the participants in our experiments, the simulator testbed was able to simulate the real-world

driving experience with a high level of realism, hence capturing the participants' driving in-cabin behavior can be seen as a success. Furthermore, comparing the questionnaire entries with the labeled data and facial recordings demonstrates that the inducement of our aimed emotions on the 15 participants was successful.

## 5.5 Multimodal Architecture

One of the aims of this work is to demonstrate the importance and relevance of in-cabin behavioral factors in identifying emotional states. This objective follows the primary intention of maintaining the systems' robustness while contributing to the overall context awareness. The current status of the developments is focused mainly on facial recording through the cameras mounted inside the cabin. However, privacy concerns aside, such solutions are not very reliable in dynamically changing environments. As stated earlier, changing the surrounding visual context of the vehicle, like sudden entering into a tunnel or any change in weather conditions or even the reflection of a direct beam of sunlight, can easily falsify the input feed of the cameras; therefore, the predictions end up in undesired and wrongful areas. Hence, there is a need to utilize more modalities in the prediction pipeline to deal with this issue and increase the robustness of the systems in confronting such situations. According to the outcome of our preliminary study in Section 5.3, along with the data collection phase on the VIRES VTD simulator setup, it is evident that most of the driver interaction with vehicle components inside the cabin is focused on adjusting the speed of the vehicle by the pedals and positioning the car by performing maneuvers through the steering wheel. Hence, by recalling our definition from the in-cabin behavior, represented in Section 4.4.1, the *vehicle acceleration intensity* and *steering wheel angular velocity* are chosen as the most representative emotional factors based on the driver's in-cabin behavior. For this purpose, we have designed an experimental multimodal emotion recognition architecture as depicted in Figure 5.26.

### 5.5.1 Facial-based Modality

The camera-based modality plays a crucial role in identifying the subject's emotional state in a multimodal emotion recognition architecture. This module typically consists of different steps to detect a face, pre-process the captured image, extract features and classify the emotion accordingly. The majority of the face-detection algorithms focus on the frontal part of the human face. It is also true in the conditions of our experiments in an in-cabin environment; therefore, only algorithms able to work with 2D images are considered in this stage. There are three main approaches to train a machine to locate and detect the face out of each frame. They are namely feature-based methods, initially proposed by Viola and Jones [122], the classic *Histogram of Gradients* (HOG) based methods, and *Convolutional Neural Network* (CNN) based approaches. However, as the primary goal of our work is to design a real-time system that can be deployed on commodity hardware integrated inside the cabin environment (i. e. our VIRES VTD simulator), the CNN approach is excluded from this list due to its dependency on comparably high computational processing power. Among the remaining approaches:

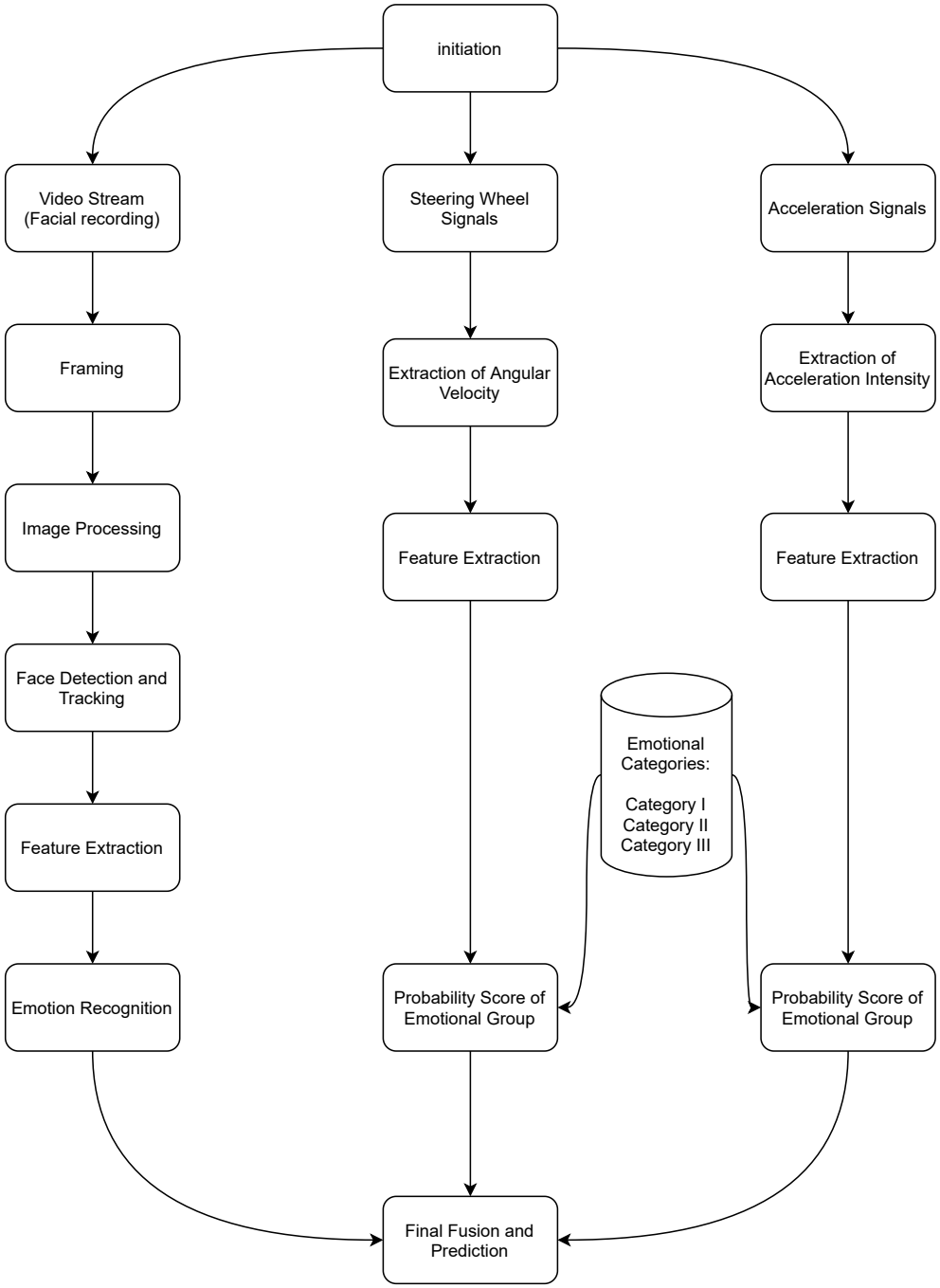
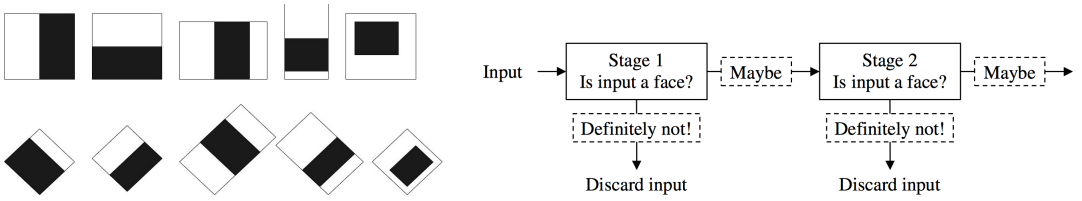



**Figure 5.26:** Overall data flow in designed the system

**Viola and Jones** is one of the widely used methods in object detection, which can handle up to 15 frames per second. This approach also is famous for its slow training and efficient use of Haar-like features. In order to implement this method:

- The input frame is turned into an integral image. For this purpose, the value of each pixel is taken equal to the sum of pixels on the left and above the corresponding pixel. An intuitive example of this is shown in Figure 5.27(a). This feature provides an option to compute any rectangle inside of the image only by summing four corner values,
- Then minimum and maximum size of the sub-windows are selected, and later, each one of the sub-windows starts to slide with a fixed chosen step,
- A collection of Haar-like features analyzes each sub-window, and a single value is calculated by subtracting the sum of white rectangles from the sum of black ones. The possible number of Haar features can rise to 160.000 for a 24x24 pixel area; hence we use a set of cascade-connected classifiers to find the most suitable features,
- Each sub-classifier holds some portion of Haar-like feature extractors, and if the sub-classifier locates the target, the face of the subject, in this case, based on this feature set, passes this area to the successive sub-classifier in the chain, or breaks the chain in case of failure to locate the target as depicted at Figure 5.27(b).




(a) Haar-like features (b) Cascade of classifiers

**Figure 5.27:** Main characteristics of Viola-Jones (feature-based) method

**Histogram of Oriented** descriptors is used by applying a fixed-sized sliding window over an image pyramid built upon them. The normalized HOG orientation features make this method capable of reducing false-positive rates far better than the state-of-the-art Haar Wavelet-based detectors [194]. The widely used implementation of this approach is provided initially by the dlib library [218]. The pre-processed gray-scaled image is an input object of dlib face detector, and the output is the rectangle coordinates containing the face.

In order to evaluate both methods mentioned above and choose the most suitable one for our purpose, a preliminary evaluation of both of them is performed on the *YouTube Faces Database* [219]. The results listed in table 5.5 demonstrate that the HOG-based face detection method outperforms the Viola-Jones concerning accuracy. Therefore, despite the smaller average required processing time in Viola-Jones, the HOG-based method is eventually chosen due to the importance of accuracy in the predictions for our multimodal architecture.

| Method | Frames | Faces found | True positive | False positive | Average time |
|---|---|---|---|---|---|
| Viola-Jones | 60525 | 58267 | 57034 | 1233 | 41 ms |
| HOG based (dlib) | 60525 | 59855 | 59704 | 151 | 117 ms |

**Table 5.5:** Performance comparison of Viola-Jones and HOG-based face detection methods on *YouTube Faces Database*

Pre-processing the captured images from the camera for face detection is necessary and is typically divided into several stages. First of all, the resolution of the image should be normalized. In our experimental setup, raspberry pi camera module v2, records videos in HD format (resolution of 1280x720) with 60fps. The subject face is comparably a big object and usually covers a considerable portion of the image based on the camera angle placed inside a cabin; hence, we can easily decrease the resolution to 320x180 pixels without causing any critical negative impact on the accuracy. The second step is to convert a 24-bit colorful image to an 8-bit gray-scale one. The transformation from RGB to gray-scale is performed by using the formula of:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \tag{5.4}$$

As shown above, each color is assigned with a different weight because human color perception is not evenly balanced, and we, as humans, are more sensitive to green color than blue and red. Therefore, the last step of image pre-processing is an improvement of image contrast. For this purpose, OpenCV provides *contrast limited adaptive histogram equalization* (CLAHE) method. The basic version of this method is global histogram equalization which sets global contrast for the whole image. However, it is unsuitable when the background belongs to the dark spectrum, and the face region is closer to soft and light colors. In such cases, after equalizing the histogram value, the facial part can lose most of the information due to over-brightness. In order to deal with this problem, the CLAHE method is widely applied, which divides images into small parts called tiles. The default size of each tile is 8x8. Then these tiny regions are histogram equalized as usual. Now, the facial part is safe from information loss because equalization is applied to small regions.

After face detection, we need to extract a set of features that hold viable emotional state information. Finding a reduced set of primary features, also referred to as feature extraction, is a standard practice in machine learning, pattern recognition, and image processing [220]. In reality, only some parts of the face are affected by human emotions. They are namely mouth, eyebrows, eyes, nose, and jawline. We use a facial landmark detector designed by Kazem *et al.* [221] to extract important facial features from the *region of interests* (ROI) and construct feature vectors for the classifier. The proposed method for facial landmark detection using an ensemble of regression trees is widely used and implemented in the dlib library. The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face. The Figure 5.28 depicts the visualization of all those 68

coordinates in this regard. Dlib facial landmark predictor was initially trained on iBUG 300-W dataset [8], and these annotations are part of it.

After predicting facial landmarks, we would like to crop out only the ROIs of the face. As it is shown in Figure 5.28, the left eye is represented by 6 points where point 37 is left and point 40 is the right corner of the left eye. The top corner is a point with a higher Y-axis value between points 38 and 39. Accordingly, the bottom corner is a point with a lower Y-axis value between points 41 and 42. After defining all corner points, the rectangle containing all these points is constructed and cropped from the image. In the same way, all other ROI parts are cropped out. The last and the most critical step is the extraction of HOG features from the resulting ROI images. Parameters of the HOG descriptor are listed in table 5.6. These parameters are identified by Thibaud *et al.* [222] after making intensive tests using different values for HOG block size, stride, and nbins. As a result, we get a feature vector with a length of 2400 (25x16x16).

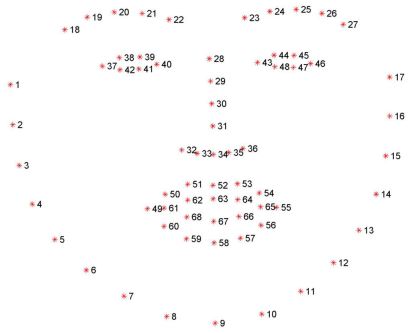


**Figure 5.28:** Visualization of 68 facial landmarks [8]

| Parameter | Value |
|---|---|
| Window Size | 96x96 |
| Stride | 16 |
| HOG block size | 8x8 |
| nbins | 6 |

**Table 5.6:** Parameters of HOG feature descriptor

After extracting the feature vector from the image, we can pass it directly to a classifier. However, before this step, we have to build and train the model respectively. We use the extended Cohn-Kanade (CK+) [9] dataset to train our model in this work. This dataset was originally published to promote research in the automatic facial expression detection area. The CK+ holds 327 sequences where the first frame is neutral, and the last is an apex of one of the emotions. One hundred eighteen actors participated, and all sequences were recorded in a posed way, as is depicted in Figure 5.29. The number

of sequences for each emotion is not equal. For instance, there are only 25 sequences for *fear*, whereas 64 are provided for *surprise* emotion.

**Figure 5.29:** Inducement of different emotions in CK+ dataset [9]

We fetch the first and the last frame from each sequence and store them in a separate directory. Also, the label of each frame is written to a joint CSV file. This way, we can easily make cross-fold and benchmark tests when it was required. Some of the sequences are colorful, and others are gray-scaled; hence all images are pre-processed to be used in model training. As a result, we get 739 labeled frames where 499 of them represent a neutral face, 35 anger, 46 disgust, 19 fear, 54 happy, 21 sadness, and 65 surprises. The number of neutral frames naturally is higher than the others. Given labeled training data, an SVM can categorize new examples by producing an optimal separating hyperplane. Basically, in two-dimensional space, the hyperplane divides it into two parts where ideally, each section contains only the same type of data. One of the essential factors in separation is the margin left by the hyperplane between two categories. Ideally, it should be as comprehensive as possible, but there exists a trade-off as well. Regularization parameters, also called the C parameter in python, are controllers between maximum margin width and classification error. SVM optimization will choose a smaller margin hyperplane to classify all training data correctly by setting a larger C value. On the other hand, a lower C value will lead to a hyperplane with a higher margin, leading to the misclassification of some points. Although SVM is mainly used for linear separation of binary data, the SVM kernel functionality also provides polynomial, radial basis function, and sigmoid types of classification. In this work, we employ SVM implementation provided by the scikit-learn library [223], which supports three classes of SVC, NuSVC, and LinearSVC capable of performing multi-class classification on the dataset. These classes accept slightly different parameters and end up in marginally different solutions. According to a comparison made in [224], a *one-to-one* method outperforms *one-to-rest* in facial emotion recognition. Hence, we utilize the SVC class with the linear kernel to make the final classification. After several experiments, the best-fitted parameters of the SVM model are found and listed in Section 5.5.1.

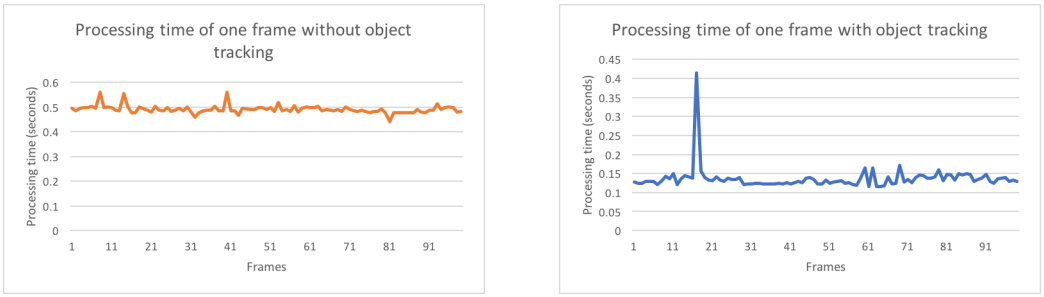
In order to train our model, the K-fold cross-validation method was used with a K value set to 10. In K-fold cross-validation, original data is randomly divided into k equal

| Parameter | Value |
|---|---|
| Kernel | Linear |
| C-penalty | 10 |
| Gamma | 0.1 |
| Decision function | One-vs-rest |
| Tolerance for stopping criterion | 1e-3 |

**Table 5.7:** Parameters of SVM model

partitions, one partition is kept for validation, and all others are used for training. This process is repeated k times where each partition is used once as validation data. Finally, the results from each interaction are averaged to deliver a single estimation.

The object tracking systems usually are faster than object detectors. It is due to this fact that object detectors process each frame individually without holding any information about the results of the previous frame (localizing the object). In contrast, tracking objects use location, direction, and motion of the object in previous frames, hence performing a prediction about the object's location in the next frame by quick search around the object's estimated location. The performance difference between these two systems becomes even more considerable when a frame owns a very high resolution. However, the demanded object occupies only a tiny portion of this ample space. This fact demonstrates the importance of using an object tracking system in our approach.




(a) Without face tracking

(b) With face tracking

**Figure 5.30:** Speed of facial emotion recognition algorithm on Raspberry Pi 3 Model B

In this work, we use the fast object tracking method originally proposed by Danelljan *et al.* [225], which demonstrates higher accuracy and performance compared to state-of-art methods such as ASLA [226], SCM [227], Struck [228], and LSHT [229]. The outcome of the experiments performed on Raspberry PI 3 Model B, shows that we gain a significant improvement in the performance of the emotion recognition algorithm with the help of a face tracking system, as depicted in Figure 5.30. The initial version of the algorithm requires 0.49 seconds on average to process one frame. However, after em-

powering it with a face tracking system, dramatic performance improvement is achieved. The Figure 5.30(b) shows that the average time required to process one frame is reduced to 0.14 seconds. The spike observed on the graph results from a switch between tracker and detector when the tracked object is lost. The designed algorithm, which is used to classify the facial emotions in real-time, is depicted in algorithm 1.

**Algorithm 1** The HOG-based facial emotion classifier

```
featureVector ← init list
SVMClassifier ← load model
while newFrame is exist do
    frame ← FetchVideoStream()
    grayFrame ← GrayscaleImage(frame)
    if faceTracker(grayFrame).Score < threshold then
        face ← detectFace(grayFrame)
    else
        face ← faceTracker(grayFrame).Position
    end if
    ROIarray ← FetchROI(face)
    for each ROI in ROIarray do
        hog ← HOGDescriptor(ROI)
        featureVector ← featureVector + hog
    end for
    result ← SVMClassifier(featureVector)
end while
```

One of the crucial steps to maintain the system's performance while preserving its real-time prediction behavior is to filter out the frames that do not hold relevant information for the prediction pipeline; for example, when the driver's tilt is so high that the desired parts of the face, containing facial expressions, become invisible. As a result, the classifier could easily be falsified. In order to overcome this issue, the head position needs to be estimated in each frame, and only the faces with small tilt and pan angle should be considered for further processing. Technically the pose of an object is defined according to its relative orientation and position to the camera. Thus, the pose of the object changes either by moving the object or moving the camera. This problem is referred to as *Perspective-n-Point* or shortly PNP in computer vision [230]. The goal of the PNP problem is to determine the six-degree-of-freedom (DOF) pose of the camera concerning the surrounding world while having the information about calibrated intrinsic camera parameters, positions of n 3D points on the object, and the corresponding 2D projections in the image. For this purpose, the following formula is defined:

$$sp_c = K[R|T]p_w \tag{5.5}$$

Where the $p_w = [x, y, z, 1]^T$ is the coordinates of the 3D point, $p_c = [u, v, 1]^T$ is the coordinates of the corresponding point in the 2D image plane, K is the matrix holding parameters of the calibrated camera and s is the scale factor of an image. The focal

length of the camera ($f_x$ and $f_y$), the optical center in the image ($u_0$ and $v_0$), and the radial distortion parameters ($\gamma$) are the intrinsic parameters of the camera. This leads to the following equation:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5.6}$$

This equation can be solved using the *direct linear transform* (DLT) method, and we get the desired r (rotation) matrix and t (translation) vector of the camera respectively out of it. The next step is to compute Euler angles ($\Phi\ \Theta\ \Psi$) from the rotation matrix. The conversion process is thoroughly explained over [10], and the suggested algorithm is depicted in Figure 5.31. After calculating Euler angles, we put minimum and maximum possible values for each one of them. All head pose rotation angles not fulfilling this condition will be skipped to maintain the overall accuracy.

```
if (R31 ≠ ±1)
   θ1 = −asin(R31)
   θ2 = π − θ1
   ψ1 = atan2(R32/cos θ1, R33/cos θ1)
   ψ2 = atan2(R32/cos θ2, R33/cos θ2)
   φ1 = atan2(R21/cos θ1, R11/cos θ1)
   φ2 = atan2(R21/cos θ2, R11/cos θ2)
else
   φ = anything; can set to 0
   if (R31 = −1)
      θ = π/2
      ψ = φ + atan2(R12, R13)
   else
      θ = −π/2
      ψ = −φ + atan2(−R12, −R13)
   end if
end if
```

**Figure 5.31:** Computing Euler angles from a rotation matrix as described in [10]

### 5.5.2 Steering Wheel Angular Velocity as Abnormal Behavior Indicator

Steering wheel angular velocity is one of the primary behavior-based emotional factors recorded during all rides from the participants in the experiments. Because some situations like avoiding obstacles on the road (e. g. vehicles, or pedestrians) may require a sharp steering movement, the driver's usage of the steering wheel must be observed for periods of time instead of following an instance-based approach by focusing on specific

points in time. Afterward, the average angular velocity in the steering wheel is calculated for a small fraction of time. This averaged value will be used later as the threshold. We aim to detect angular velocity higher than the dynamically defined threshold and mark this fraction of time as abnormal. An accelerometer is a device that measures the acceleration of an object relative to free-fall. At rest position, it measures 1g, which is the earth's gravitation pull ($g = 9.81m/s^2$). It is widely used in inertial navigation systems to detect the subject's orientation (and, in some cases, stabilization). However, it can detect tilt angle reliably only when it is fixed and stationary. Therefore, it is mainly used in conjunction with a gyroscope that complements the accelerometer's drawbacks, and together they can obtain better results in tilt angle measuring. Our work uses the MPU-6050 device, which contains a 3-axis accelerometer and 3-axis gyro data on a single chip. The maximum sample rate provided by this device is 1KHz. In order to gather data from the device, the I2C protocol is used.

| Sampling Rate | | 30 (Hz) |
|---|---|---|
| Measurement Range | Accelerometer | 5 (G) |
| | Gyroscope | 300 (deg/s) |
| | Magnetometer | 750 (mGauss) |

**Table 5.8:** Specification of MPU6050 sensor

Three-axis accelerometer data provided by MPU-6050 comes in a range of [-16384.0, 16384.0]. Therefore, as the first step, we scale the raw data to the range of [-1, 1] and then use them to calculate an angle. The eq. (5.7) shows the rotation angle around the X-axis based on provided X, Y, and Z values. Similarly, we can also calculate rotation around Y, but this is not required in our work as a steering wheel rotates only around one axis.

$$A_x = arctan(\sqrt{\frac{X}{Y^2 + Z^2}}) \tag{5.7}$$

Theoretically, the obtained rotation angle is enough to calculate angular velocity; however, it suffers from inaccuracy, hence must be filtered accordingly:

$$angle = \alpha * (angle + gyro * dt) + (1 - \alpha) * (A_x) \tag{5.8}$$

As mentioned before, the accelerometer is very volatile to all forces that act upon the object. Considering that the device will be placed inside a non-stable environment like the cabin, even a little forcing work may falsify the whole measurement [231]. On the other hand, an angular position obtained by the gyroscope is more stable and is not under high impact from the external forces. However, after spending some time, it also acquires an adrift and is not able to return to its original position [232]. In order to deal with this issue and take advantage of both methods, a complementary filter is used. The concept of the complementary filter was initially proposed by Colton at [233].

For detection of tilt angle, this filter makes low-pass filtering on data obtained from the accelerometer. The high-pass filtering for tilt estimation is applied to data obtained from the gyroscope. The fusion of both estimations gives us an all-pass estimation [234].

After obtaining a tilt angle of the steering wheel, to measure the change of the angle in a small fraction of time, we calculate angular velocity using the eq. (5.9) where $\theta_i$ is the initial angle, $\theta_f$ is the final angle, and $\Delta t$ refers to the time passed during the angle change.

$$\omega = \frac{\theta_f - \theta_i}{\Delta t} \tag{5.9}$$

All these calculations are repeated for each newly obtained data, so shortly after the process, evaluation of data and identification of aggressive driving patterns can be initiated. Continuous monitoring of behavior-based indicators using appropriate sensors results in real-time data flow in time series. This obtained data is later used to detect normal and abnormal values. From a technical point of view, this process is called anomaly or outlier detection. An outlier is defined as "*patterns in data that do not conform to a well-defined notion of normal behavior*" [235]. Anomaly detection is a process to find these outliers on data by comparing them with some pre-defined patterns or rules. This problem has gotten significant attention in recent years and is researched in various fields such as statistical analysis, artificial intelligence, and machine learning. Time series anomaly detection is also researched and applied on different kinds of problems, such as detecting anomalous flight sequences using sensor data from aircraft [236] or detecting outlier heartbeat pulses using ECG data [237].

A time-series $X = x(t)|1 < t < m$ is a sequence of d-dimensional observations vector $x(t) = (x_1(t), x_2(t), ...x_d(t)$ ordered in time. In most cases, data is collected in a static time interval. However, when the interval is not fixed, an average number of samples collected in one second is considered the sampling frequency. Time series analysis comprises methods for analyzing time-series data to extract meaningful statistics and other data characteristics. Segmentation is one of these methods which helps to find out sequential anomalies. The main goal of segmentation is to split time series of $t$ into small sub-series of $< t_1, t_2, ..t_k >$ where $t_k$ is a sub-sequence of $t$ such that:

$$t = \bigcup_{i=1}^{k} t_i \tag{5.10}$$

and

$$t_i \bigcap t_j = \emptyset, i \neq j \tag{5.11}$$

The sliding window is a widely used technique to detect anomalies or pre-defined patterns on data streams. In the sliding window method, the most recent data is more valuable than a fixed window-based approach since the boundaries of the window change over time, making it fit perfectly with the amount of data generated in real-time. Generally, there are three types of sliding windows: **pure sliding window** where the step

between two successive windows is always less than the window size, **jumping sliding window** is the one where the step between two successive windows is equal to window size and eventually, the **hopping sliding window** where the size of the window is always larger than the step between two windows [238]. We use the *pure sliding window* in this work since we aim to cover all the possible abnormal values. The angular velocity is calculated and compared to the threshold for every window average. Depending on the comparison results, 1 or 0 value achieved where 1 is the sign of aggressive driving.

$$\forall(m), |A_m| > \alpha \tag{5.12}$$

$$A_m = (1/n) * \sum_{i=1}^{n} x_i, n = w_l, w_l > w_s \tag{5.13}$$

Here m is the index of the window, $A_m$ is the average for the window m. The window length is named as $w_l$, and window step is $w_s$. The $\alpha, w_l, and w_s$ are the primary hyper-parameters and considerably impact the detection accuracy. In order to define the length of the window, several factors must be considered. When the length of the window is very small, then there is a high chance of missing abrupt steering wheel maneuvers, as the time taken to make this maneuver will be divided into small time fractions where average angular velocity is not so high and can be difficult to differentiate from normal smooth vehicle rotations. Respectively, too big window length can also lead to missing some abnormal values. In this case, the usual driving after sharp maneuvers will decrease the average angular velocity over time. The window length should be based on the driver's reaction time to overcome this problem. According to the earlier experiments by Zang *et al.* [239] and Khasbat *et al.* [240], the driver's reaction time in a real-life environment is around 800ms±50ms. Also, if we consider the situations where to avoid the obstacles, a driver can make sharp steering during a short period, then around 200ms more can be added on top of the pre-defined reaction time. This gives us approximately 1 second, which is enough for the driver to make biased decisions. During this period, any high angular velocity not compensated with the angular velocity in the opposite direction will be considered an aggressive driving maneuver. As mentioned before, our sensors work in 100Hz frequency, then during 1 second, 100 values are gathered. Therefore the length of the window is set to 100 in this work. The base value of another essential hyper-parameter threshold ($\alpha$) is taken equal to 30 degrees/seconds after making several tests on the simulator. As mentioned, the threshold needs to be dynamically set due to the impact of the speed. For instance, 30 degrees/seconds angular speed with the car speed in 120 km/h is more dangerous and looks aggressive than making the same maneuver with 20 km/h. Therefore the following equation is used in the formulation of the threshold.

$$\alpha = k * \alpha_i, k = 100/v \tag{5.14}$$

Where the $\alpha_i$ is the pre-defined threshold, and $v$ is the current velocity of the vehicle. In order to evaluate the level of aggressiveness for the driver, we can calculate the frequency of happening such *non-friendly road behavior* as follow:

$$f_a = \frac{(n-1) * f_{a-1} + f_l}{n} \tag{5.15}$$

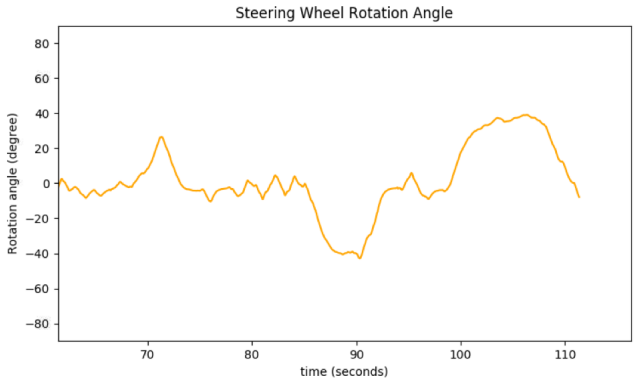
In this formula, the time difference between the last and currently observed anomaly is defined as $f_l$. Additionally, n states the number of observed anomalies and $f_{a-1}$ shows the last calculated average frequency.

The Figure 5.32 and Figure 5.33 illustrate the position and velocity of the steering wheel correspondingly in a relaxed and aggressive driving style for the driver. According to them, the tilt angle of the steering wheel varies in the same range [-40, 40] for both cases, but the result of the anomaly detector differs. The reason relates to the way the rotation has been performed. As in the first scenario, the driver turns the steering wheel smoothly, resulting in a gradual increase and decrease in tilt angle. However, in the second case, we can observe a set of sharp spikes due to abrupt steering wheel rotations by the driver. The $t$ becomes even more distinguishable when comparing a range of angular velocities in Section 5.5.2 and Section 5.5.2. Aggressive driving style demonstrates angular velocity fluctuation between -100 and 100 degrees/seconds, at least two times more than the numbers obtained in relax mode for the same driver. Consequently, the system detects four anomalies in the second scenario against 0 in the first one.

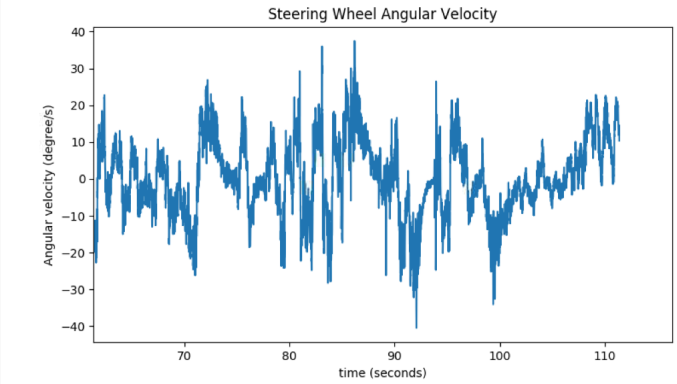
Our proposed approach contains several advantages and disadvantages. One of the notable features of this method is its reliance on low processing powers and functionality on commodity hardware such as embedded devices. Another advantage of this technique is that delay between the prediction phase, and data gathering does not exceed 1 second. This time difference is acceptable for such a system and makes it usable with other system components in the following phases. Respectively, our problem differs from traditional ones where anomaly detection is adopted on already collected data, such as detecting abnormal temperature on some given time-series or detecting CPU spikes on servers. In this situation, adopting anomaly detection in the current problem can lead to false negatives when the driver turns the steering wheel a little bit after a long stable drive. Therefore, using hyperparameters helps to avoid such situations and can be counted as an advantage in the current implementation. However, the disadvantage of this approach can be a necessity to define these hyperparameters depending on the sensitivity of the steering wheel system in a car.

### 5.5.3 Vehicle Acceleration Intensity as Emotional Indicator

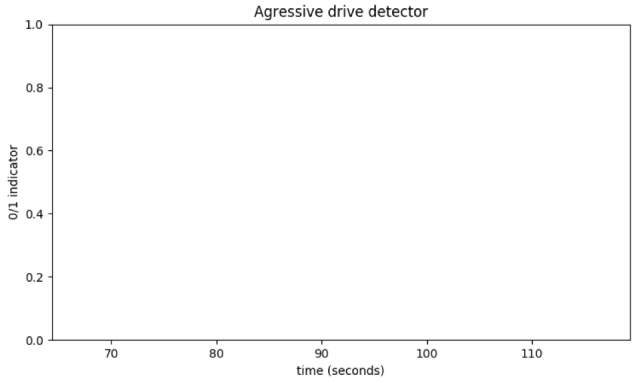
The vehicle acceleration intensity is another considered behavior-based emotional indicator in our designed multimodal recognition system. The correlation between vehicle acceleration/deceleration behavior with identity and driver age has already been studied before [216, 241]. In this work, however, we construct a model to recognize the patterns in the time series of vehicle acceleration that can be utilized to identify the driver's emotional state. The required data, which is used for feature selection and training of the model, is gathered from our VIRES VTD simulator setup, the same as the previous experiment. Participants with driving skills are asked to drive pre-defined scenarios in




(a) Steering wheel rotation angle



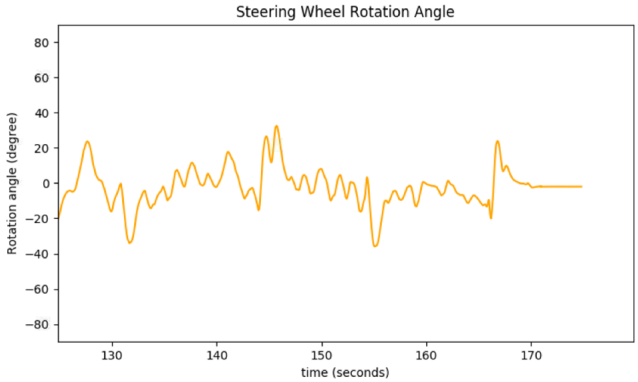

(b) Steering wheel angular velocity



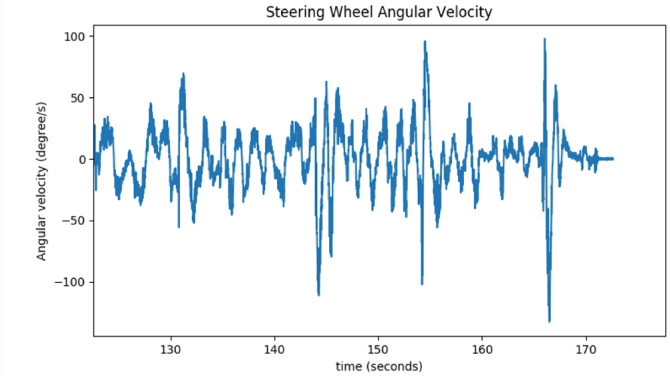

(c) aggressive drive detector

**Figure 5.32:** Data collected in relaxed driving mode for one driver

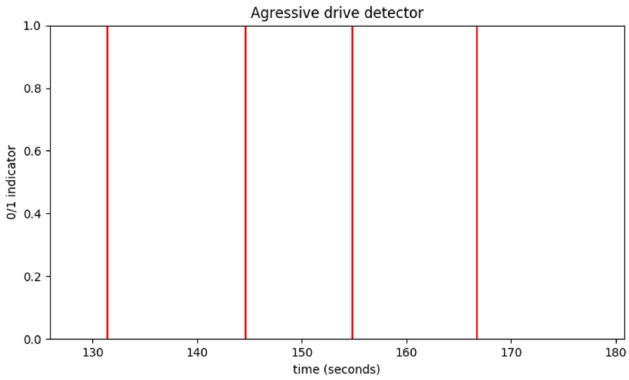
our real-car simulator under various emotional states to collect the required data for our experiments.




(a) Steering wheel rotation angle




(b) Steering wheel angular velocity




(c) aggressive drive detector

**Figure 5.33:** Data collected in aggressive driving mode for one driver

According to the results of our empirical study in Section 5.3, drivers tend to drive more actively and make abrupt movements more often when they are emotionally *excited*,

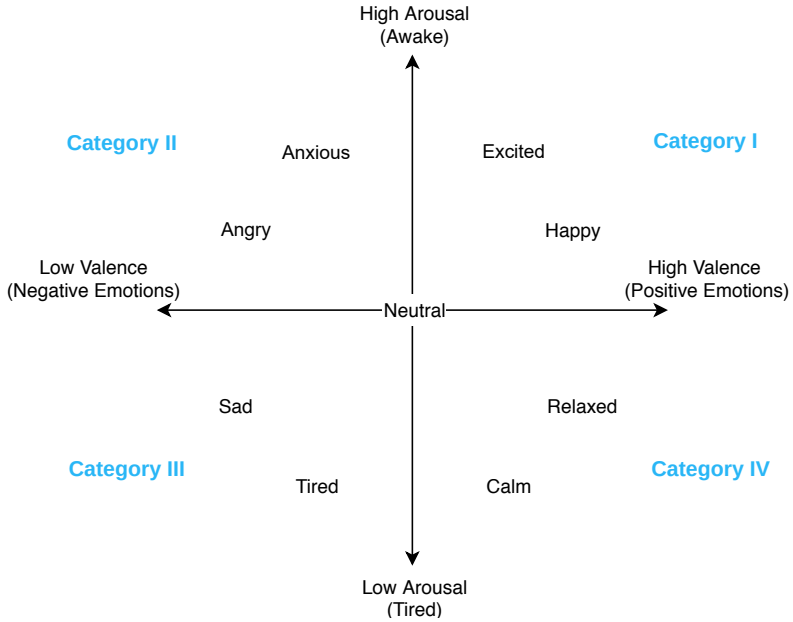
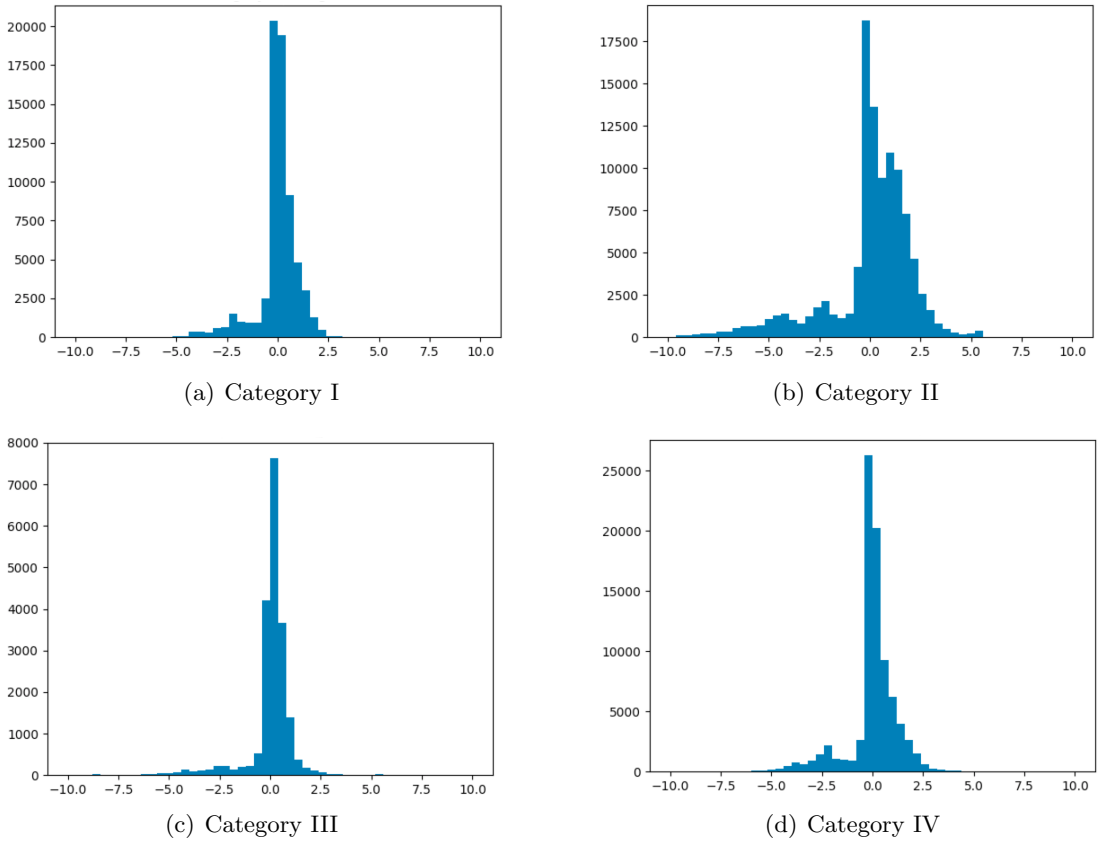



**Figure 5.34:** Different emotional states adapted into arousal-valence measure

*angry*, *happy* and *anxious* (including both the *confused* and *stressed* states). On the other hand, the drivers' driving behavior becomes more passive and most likely reduces the sudden eye/body movements when they are sad or tired. These emotions match the emotional categories we adopted to *valence and arousal* measures [242] depicted over Figure 5.34. Valence is positive or negative affective and defines the description level on a scale from pleasantness/positive emotions to unpleasantness/negative emotions. Similarly, the arousal measure indicates how calm or excited the subject is and implies the level of reactiveness of the subject to a stimulus [243]. Here we outline the main groups of patterns based on the fact that active-aggressive driving skills are related to categories I and II. In contrast, category III represents passive-defensive driving behavior, and category IV depicts (relatively) a neutral state of the driver. Of course, neutral states are not representative as the positive and negative groups of emotions. However, naturally, most of the emotional status of the drivers during driving ends up in this category.

In order to compare the vehicle acceleration distribution during different emotional states, the histogram of the frequency distribution of the rides in our simulator testbed is plotted in Figure 5.35. The vehicle acceleration collection for categories I, III, and IV is very similar and does not hold considerable signs for distinction. On the other hand, the Figure 5.35(b) demonstrates a comparably more comprehensive range of values (-10 to +6) for category II, in comparison with other categories, which indicates that the drivers in the emotional state of angry or anxiety(stressed & confused) tend to accelerate and decelerate relatively faster.




(a) Category I

(b) Category II

(c) Category III

(d) Category IV

**Figure 5.35:** Frequency distribution of vehicle acceleration in 4 groups of emotional status

In order to demonstrate that a driver with high-arousal and negative valence (category II) can be detected by using acceleration data generated during each ride, we label all data related to Category II as 1 and the rest as 0. One of the commonly used machine learning techniques for binary classification, as stated before, is SVM. The table 5.9 represents the main parameters of our SVM classifier. During the training processes, 69 samples are used. Class 1 contains only 25% of samples, which makes our dataset relatively unbalanced. We assign a 4X weight to class 1, compared to class 0, to deal with this issue. Each sample holds an array of 8000 acceleration values over 2 minutes. 10-fold cross-validation achieves 96% accuracy on our collected dataset through the simulator testbed.

## 5.6 Fusion Model

The standard practices show that the combination of multiple classification decisions generates a comparably better result than utilizing single classifiers [244]. This technique is mainly referred to as *ensemble learning* in machine learning, which combines

| Parameter name | Value |
|---|---|
| Kernel | Radial basis function |
| C - penalty parameter | 10.0 |
| Gamma | 0.0001 |
| Decision Function | 1e-3 |
| Class – Weight | 0 – 1 and 1 – 4 |

**Table 5.9:** SVM parameters for vehicle acceleration intensity-based emotion recognition

several machine learning techniques into one predictive model. A commonly used class of ensemble algorithms are forests of randomized trees (as known as *random forests*). Random forest is an averaging algorithm on top of multiple decision trees built on the same training data set. The randomness of this algorithm helps to increase the bias of the forest slightly. However, the variance decreases due to the averaging of less correlated trees, making it comparably a better model. Nevertheless, the random forest does not keep a white box model of a single decision tree, and we can not extract a tree and learn about the influences of a single feature. Therefore, initially, we construct a single decision tree using 50 samples and export its flowchart tree in Figure 5.36. Visualization of a single decision tree over Figure 5.36 reveals a noticeable impact of the vehicle acceleration in the prediction of emotions from category II as is already mentioned in Figure 5.35. All 18 samples of category II are grouped using only one condition from vehicle acceleration (VA), which shows the critical role of this module in predicting the *anger*, and *anxiety* (*nervousness* or *stress*) of the drivers. The second condition in the decision tree uses the proportion of sadness emotional state felt by the driver. This feature helps to group all five samples from category III. After this step, only the samples from categories I and IV are left un-grouped. Conditions formulated by SW module and *happy*, *sadness* and *neutral* features from the facial module are utilized to deal with this issue.

After analyzing a single decision tree, we use the same feature vectors from 50 samples to train a random forest classifier (as known as the collection of decision trees) to achieve a higher accuracy rate while maintaining the robustness of the model. The first value of the generated feature vector is the result of the vehicle acceleration intensity (VA) module. The second value of the feature vector is the result of the steering wheel angular velocity (SW) module. Furthermore, the last five values of the vector represent the output of the facial emotion recognition module, as depicted in table 5.10. The frequency of generating the feature vector is set to 2 minutes. For this period, SW counts the number of abnormal steering wheel rotations. VA decides whether the driver is under stress (category II: angry/anxiety). The facial module counts the number of times the driver felt every seven basic emotions and gets normalized accordingly afterward. Finally, Hyper-parameters of the random forest are tuned using grid search. In the finest run, we set the *minimum sample lead* to 2, the *maximum depth* to 2, the *number of estimators* to 6, and the maximum features variable to *auto*.

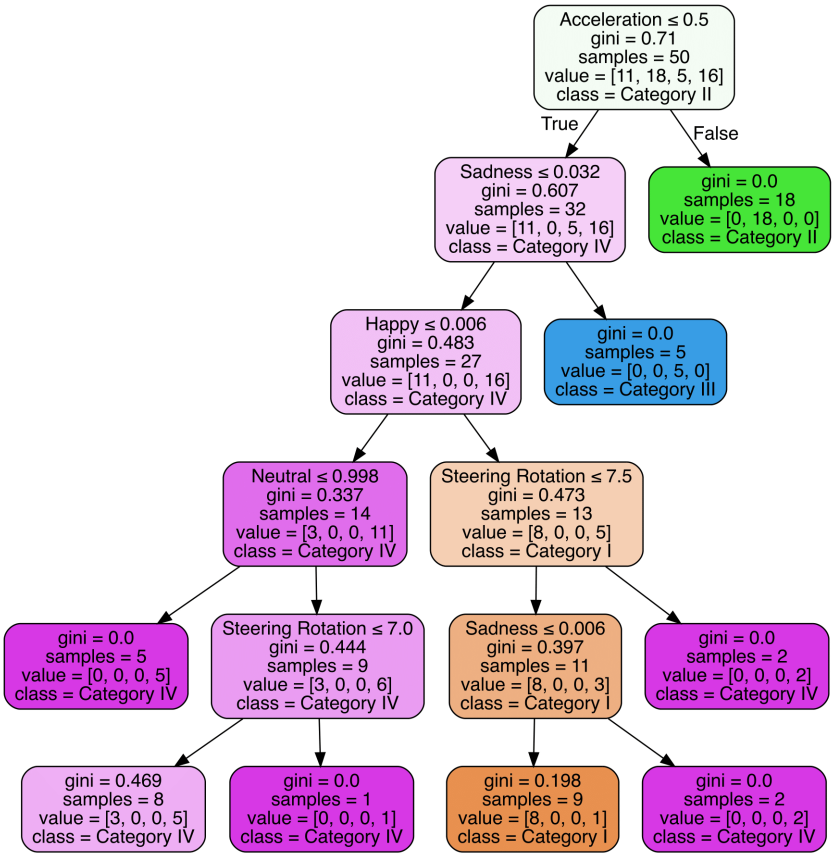



**Figure 5.36:** Decision tree of combining 3 modules of VA, SW and facial expressions

| Module | Vector index | Parameter Name | Value |
|---|---|---|---|
| VA | 1 | Acceleration | 0 or 1 |
| SW | 2 | Steering Rotation | 0 to $\infty$ |
| Facial Expression | 3 | Neutral | 0 to 1 |
| | 4 | Anger | 0 to 1 |
| | 5 | Disgust | 0 to 1 |
| | 6 | Fear | 0 to 1 |
| | 7 | Happy | 0 to 1 |
| | 8 | Sadness | 0 to 1 |
| | 9 | Surprise | 0 to 1 |

**Table 5.10:** Feature vector structure of the final emotion classifier

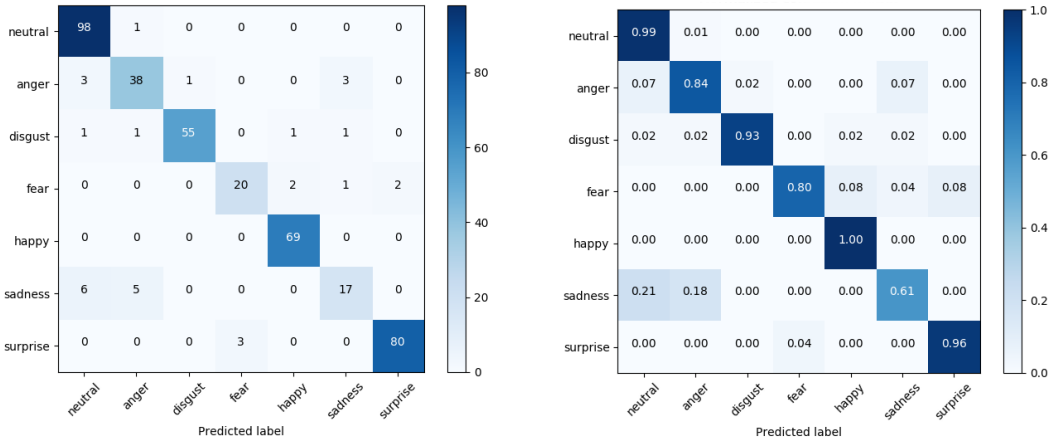
The proposed method for the facial expression-based module at Section 5.5.1 successfully achieves 93% accuracy after ten folds cross-validation in recognizing six main emotions. The comparison of the achieved result with the state-of-art methods that were similarly tested on the CK+ dataset is shown in table 5.11. Obtained accuracy is higher than most of the previously proposed methods and only 2% less than the work of Khan *et al.* [245] and Donia *et al.* [246]. It is also worth mentioning that Khan method additionally requires eye-tracker apparatus.

| **Authors** | **Method** | **Accuracy** |
|---|---|---|
| J.F.Cohn and T.Kanade *et al.* [9] | Active Appearance Models | 83% |
| H.Alshamsi *et al.* [247] | BRIEF Feature Extractor | 89% |
| W.Swinkels *et al.* [248] | Ensemble of Regression Trees | 89.7% |
| Sébastien Ouellet [224] | Convolutional Network | 94.4% |
| R.A.Khan *et al.* [245] | HOG-based | 95% |
| M.F.Donia *et al.* [246] | HOG-based | 95% |
| **Our Method** | **HOG on ROI regions** | **93%** |

**Table 5.11:** Comparison of different facial expression-based recognition methods on CK+

An evaluation of our facial module on the data collected through the simulator testbed, as depicted in Figure 5.37, demonstrates a nearly perfect performance in detection of *happiness* (100%), *surprised/excited* (96%), and *disgust* (93%). A human face is mainly in a neutral state, which is also true in driving situations; therefore, it is essential to detect the *neutral* state accurately. In this case, our method achieves 99% of the positive rate for predictions. On the other hand, there is a slightly considerable low margin of 20% in recognition of the *sad* emotional state caused by the low number of samples for this emotion and the high similarity shared with a neutral state in a human, especially in an in-cabin environment.

In order to evaluate the unified system on the collected data from our simulator testbed, ten rides from 8 different drivers are considered. Each ride is divided into sub-samples with a length of 2 minutes, giving us 79 samples for evaluation at the end. Prediction of the driver emotion in one single ride is obtained from the prediction results of its sub-samples. As represented in table 5.12, the outcome of the tests on a single ride demonstrates that the desired multimodal approach in this work achieves better results compared to each of the modules individually. As initially considered, the facial expression-based module plays a crucial role in final decision prediction, and steering wheel maneuvers and vehicle acceleration changes are complementary modules in this regard. The higher F1 score of the multimodal version indicates that steering wheel and vehicle acceleration modules together convey highly related and beneficial information regarding the emotional states, which previously was unknown to the facial module.




(a) Without normalization

(b) With normalization

**Figure 5.37:** Prediction of each emotion individually by facial module on simulator testbed data

According to table 5.12, 77.27% accuracy is obtained by including all three modalities on data samples with 2 minutes of length in a single ride experiment. However, this condition is still prone to errors and wrong prediction outcomes. In real-life situations, the 2-minutes range could be easily falsified by the dynamics of the situations like being stuck behind a red light longer than usual or traffic jams. In order to cope with such situations and increase the reliability of the results, we consider the decision-taking step at the end of each ride by summarizing the emotion predictions performed for only sub-samples and selecting the most frequently felt emotion accordingly as the final prediction. This way, our system successfully achieves 94.4% of accuracy for classification into four emotional categories.

| Method | Accuracy | Precision | F1 Score | Recall |
|---|---|---|---|---|
| Facial-based module | 54.54% | 54.75% | 50.45% | 49.86% |
| SW-based module | 37.5% | 10.3% | 13.6% | 25% |
| VA-based module | 68.18% | 35.51% | 37.76% | 41.37% |
| Fusion of all three modules | **77.27%** | **73.39%** | **73.59%** | **75.89%** |

**Table 5.12:** The results of each module in comparison with the fused one, in a single ride

A high-level overview of the state-of-the-art unimodal and multimodal emotion recognition approaches which can be deployed in an in-cabin environment is presented in table 5.13. Different modules are considered to evaluate different modalities in each of the multimodal experiments, so the one-by-one comparison regarding the performance

| System | Type | Method | Classes | Accuracy |
|---|---|---|---|---|
| [249] | Unimodal | Electrodermal Activity (EDA) | 3 | 70% |
| [250] | Unimodal | Facial Emotion Recognition | 6 | 70.2% |
| [251] | Unimodal | Speech Emotion Recognition | 3 | 88.1% |
| [252] | Unimodal | Speech Emotion Recognition | 2 | 80% |
| [253] | Multimodal | EDA and Skin Temperature | 4 | 92.42% |
| [254] | Multimodal | Speech & Facial Emotion Recognition | 7 | 57% |
| [255] | Multimodal | Acoustic & Facial Emotion Recognition | 3 | 90.7% |
| Our System | Multimodal | Facial and Vehicle Signals | 4 | **94.4%** |

**Table 5.13:** Comparison of different unimodal and multimodal emotion recognition systems based on accuracy and different number of emotional classes

is not intended here. However, it enlightens a set of fascinating findings and achievements. Most of the existing state-of-the-art multimodal approaches focus mainly on the fusion of speech and facial modules, where the highest achieved accuracy among them is 90.7% by Hoch *et al.* [255]. The low applicability of audio-based approaches aside, they did consider only three classes as neutral, positive, and negative emotion. Another notable method was proposed by Ali *et al.* [253]. They used the combination of EDA and skin temperature parameters of a driver as the input for a convolutional neural network and acquired 92.4% accuracy. The floating and unsteady context of the in-cabin and outside environment exposes the occupants (and among them, certainly the driver with the highest impact) to different emotional states and changes. The camera-based approaches fed only by video/image streams of the subject are unreliable solutions for this uncertain environment. This matter also becomes notable when the lighting inside the cabin changes due to the environmental changes on the route. There is no need to mention the typical related issues to facial-based emotion recognition, such as cultural or ethnic-related differences, which significantly impact the prediction outcomes, and overcoming them is still a significant challenge. Similarly, speech or audio-based approaches are not practical solutions for an in-cabin environment despite their outstanding results. The vocal communication between the occupants or the driver and the vehicle assistance systems is not remarkably continuous during the entire driving time; hence is not available all the time and is mainly limited to hums and random noises generated by bored drivers. Therefore, constant identifying the emotional states of the driver based on audio modality can not be seen as a reliable approach. Respectively, for the physiological-based solutions, despite their outstanding achievements, the limitation and drawback are the need for extra (wearable) sensors that directly collect the signals from the subject's body. This group of solutions is the perfect choice when the accuracy of the predictions is desired, and the required hardware is available, and the subject is delicate with using them. There is a bright future in sight for physiological signal-based solutions in emotion recognition with the growing advancements of technology in design-

ing new wearable service devices like smartwatches and the growing sensory environment inside the vehicle. However, the dependency on extra hardware here is still a weak point for these solutions, affecting their practicality.

From the perspective of behavioral modalities in interaction with driving components, vehicles nowadays are equipped with typical gas/braking pedals and certainly the steering wheel. The driver constantly interacts with them during the rides. The proposed system in our work, utilizing the original signals of the vehicle controlling systems (steering wheel rotation and acceleration intensity) and the real-time facial expression-based approach, achieves an accuracy rate of 94.4%. This outcome demonstrates the positive impact that incorporating such signals in the recognition pipeline can have on the outcome of the current status of the emotion prediction systems. One of the most significant advantages of such a solution is their reliance on the car's natural, already existing signals originally generated from the driver's interaction with driving components inside the cabin. Furthermore, such signals as a continuous input feed during the entire driving period ensure a reliable data source when the camera fails to provide the correct input feed due to any external un/expected factors. Besides, the integration of these signals can considerably increase the system's robustness and prediction outcomes.

## 5.7 Multimodal Emotion Recognition API

Acquiring a sufficient amount of representative data is critical in developing machine learning-based models that can generalize successfully. In emotion recognition, facial recordings of the subjects play a crucial role; however, it comes with privacy concerns. Besides, acquiring vehicle signals and driving-related data in a car suffers from similar challenges. Generally, such problems will challenge any data that can be traced back and eventually identify the subjects. This issue is not acceptable and tolerable in the automotive domain either. All these concerns undoubtedly create problems for providing sufficient data to develop efficient models, especially in emotion recognition which too many personal factors play a role. Additionally, as mentioned in chapter 2, emotional and context awareness are multimodal challenges; hence utilizing more modalities of data is unavoidable to increase the robustness, efficiency, and accuracy of the systems. In order to overcome such challenges and facilitate the access to resources, we designed an application programming interface (API) that provides a reliable platform to evaluate the developed multimodal models and extend them accordingly, based on the data provided by the data owners, while ensuring the data/model-related privacy concerns. The provided data by the data owners should not be archived on any of the machines and be solely used in a set of pre-defined emotion recognition analyses by the developed models without extracting any further information or unnecessary data analytics. Upon delivering the analysis, the data gets purged from the machines accordingly. One of the critical features of an API lies in its accessibility from the outside world through the Internet to other data owners, preferably over a set of secure communication links. The development of an API also must be done in a maintainable fashion for future integration and upgrades. In an ideal case, it must be flexible, stand-alone, and highly independent

from third-party-specific hardware/software resources on both sides of the server and the client. Besides, preserving the privacy concerns regarding user data is a *must* that needs to be taken into account in every aspect of the development chain of the API. This matter can be guaranteed by providing a reliable and well-maintained encryption mechanism.

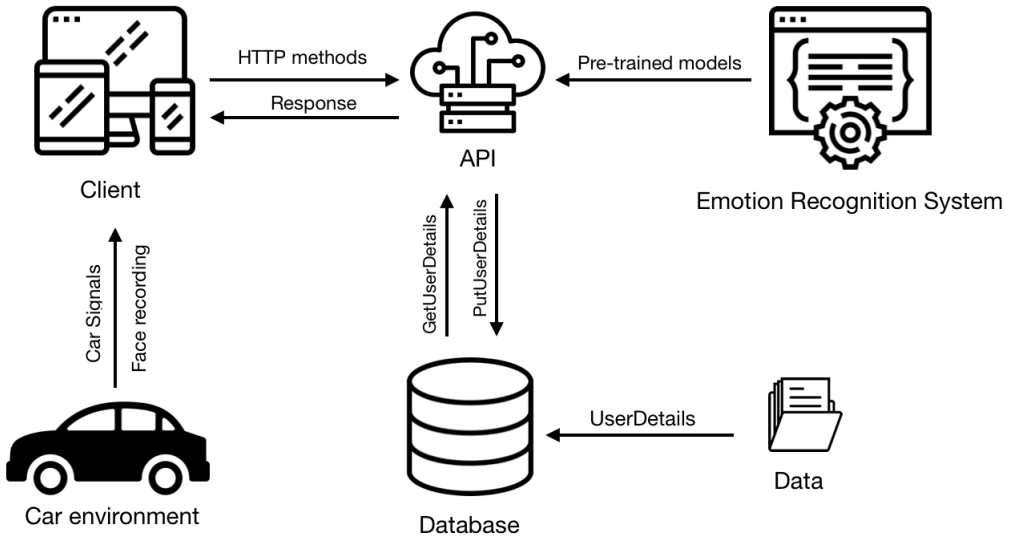



**Figure 5.38:** Top level data flow of the designed API

The current status of the developed API accepts only the following resources that can be accessed via respective URLs, listed over table 5.14:

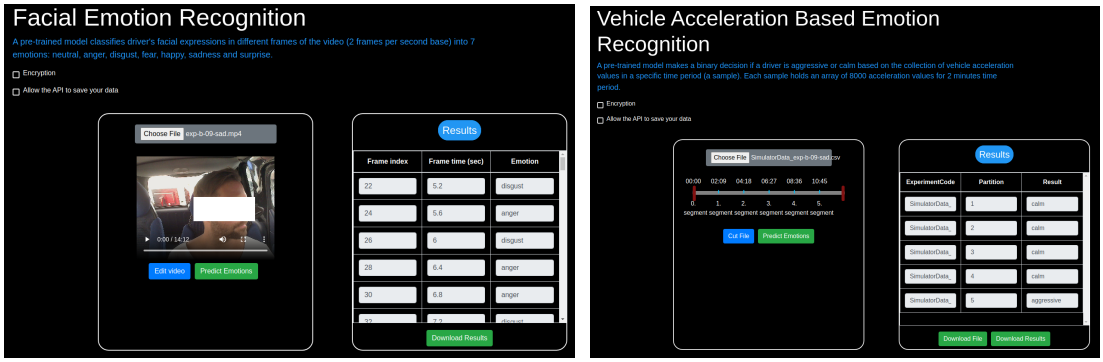
- **Facial recordings through camera:** that can be accessed by a POST request containing the frames of facial recordings in its body. Using the pre-trained integrated facial modality, the prediction can be performed upon the provided frames, and the detected emotional state gets returned inside the response message to the client.
- **Vehicle acceleration signals:** that can be accessed by a POST request containing the collected signals of vehicle acceleration intensity in its body. Using pre-trained model of acceleration modality retrieved from the behavior-based evaluation module, the abnormal acceleration patterns are extracted in preset time frames. The result is returned to the client accordingly. Suppose the user indicates the interest in performing a multimodal emotion recognition and provides the other required modalities to the API; in that case, the identified patterns will be used for this purpose in the developed multimodal model afterward.
- **Steering wheel signals:** that can be accessed by a POST request, embedded with steering wheel angular velocity signals of the vehicle.The integrated model can

calculate the number of steering-wheel abnormal rotations in different time frames, and then the identified set of patterns are forwarded back to the client. Suppose the user indicates the interest in performing a multimodal emotion recognition and provides the other required modalities to the API; in that case, the identified patterns will be used for this purpose in the developed multimodal model afterward.

- **Multi signals:** which similarly can be accessed by a POST request embedded with all frames of facial recordings as well as car signals (both steering wheel rotation and acceleration intensity) in its body. A prediction of the emotional status is performed upon the provided data with the help of the developed fused model after the necessary normalization, and the outcome is forwarded back to the client accordingly. This phase is planned to be extended with other different modalities and more efficient models in the future.

| URL | Web Component |
|---|---|
| /login | Login Module |
| /register | Register/Sign Up Module |
| /emotion-recognition/facial | Facial-based Emotion Detection |
| /emotion-recognition/acceleration | Acceleration Intensity Detection Module |
| /emotion-recognition/steeringwheel | Steering Wheel Angular Velocity Detection Module |
| /emotion-recognition/multimodal | Multimodal Emotion Recognition Module |

**Table 5.14:** Client URLs to access resources




(a) Facial recordings module (b) Vehicle acceleration intensity module

**Figure 5.39:** Different modules for the web interface of the client side

RESTful web APIs are typically based on the HTTP method, which is a stateless protocol meaning that each request is handled independently from the previous ones

and planned requests of the same client [256]. From a technical point of view, this contradicts the need for authentication and authorization of access requests. To overcome the stateless nature of HTTP requests, we are obliged to use an authentication strategy outlined in the following. At the final stage, a database is also considered to be deployed to manage and store the user registration and login credentials. The internal architecture of the API concerning user authentication is depicted in Figure 5.40. The web interface provides a set of practical tools for the user to edit, synchronize, and adjust the set of frames in the uploaded data files to facilitate the manual normalization and pre-processing phase.

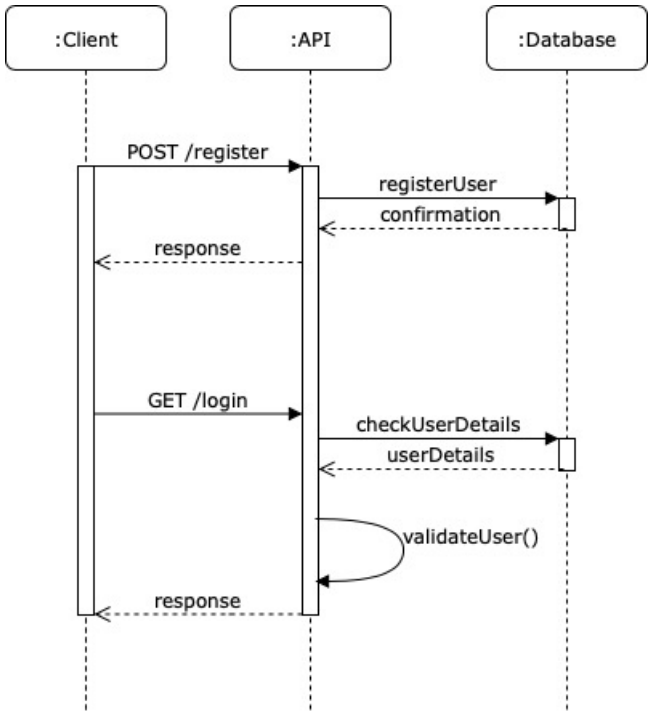



**Figure 5.40:** User authentication and login mechanism

As stated before, to preserve privacy concerns regarding the evaluation data, it is vital to consider a reliable and flexible encryption mechanism for the communication and message exchange of the API. For this purpose, we use a hybrid approach method based on both asymmetric and symmetric encryption methods. For asymmetric encryption, RSA is chosen, although it is relatively a slow algorithm. However, by employing large prime numbers for key generation, it provides a highly secure crypto-system. For the symmetric part of our hybrid encryption approach, AES is selected to be deployed. AES algorithm can support any combination of data (128 bits) and a key length of 128, 192, and 256 bits [257]. Based on experimental results provided at [258], the AES algorithm in the encryption phase consumes fewer resources and, in decryption, outperforms the other counterparts. The hybrid mechanism considered in this work consists of the encryption and decryption of data with a standard shared secret key via AES. The secret key, which

is used for the AES encryption, is secured by the RSA crypto-system. The front-end side of the client employs a web interface as a controller component that contains all the logic of the view. As depicted in Figure 5.41, the controller component on the client side manages the registration and authentication of the users in the login phase and retrieves different API resources. The client front-end is connected to the webserver that contains the API resources and provides access to the user database. An overview of the API resources and the integrated functions are presented in the class diagram of Figure A.1.

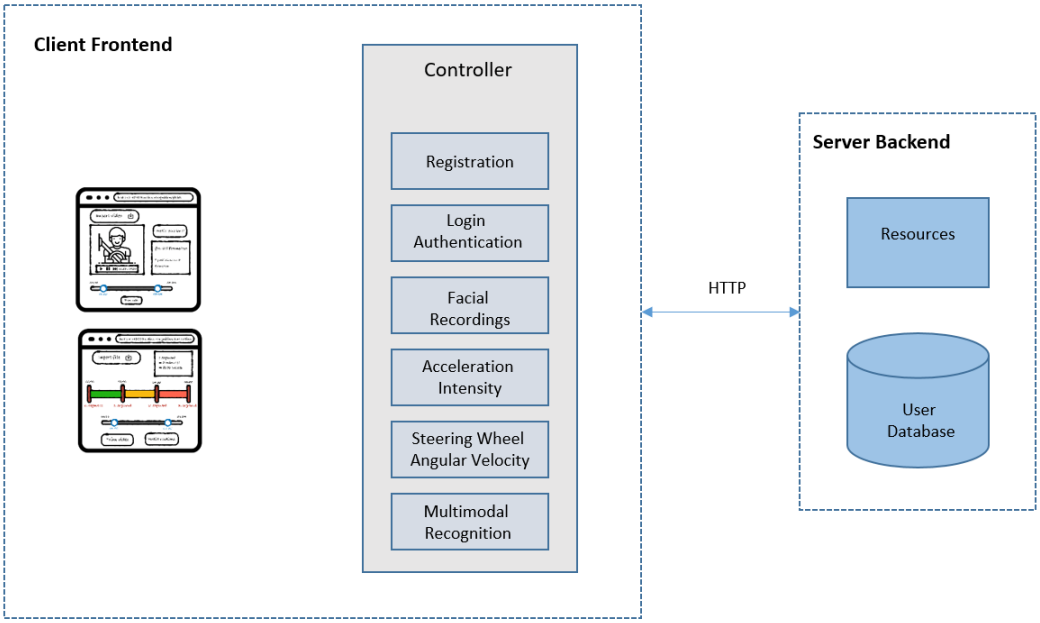



**Figure 5.41:** Controller mechanism between the client interface and the server

# 6 Epilogue

## 6.1 Conclusion

Context-awareness plays a crucial role in the era of intelligent vehicles, and it will also be an important factor when autonomous cars flood the streets in the near future [13]. However, humans still stand at the epicenter of the affecting factors in forming the context for the applications of intelligent vehicles. Hence, incorporating the role of a human user in defining the factors that impact maintaining context awareness and accordingly addressing the newly raised challenges is inevitable. This work took a user-centered approach on two binding domains of safety assurance and emotional awareness as the main enablers of context prediction architectures in intelligent vehicle platforms. One of the most critical challenges in the safety domain is the integration and enforcement of safety standards in the development chain of AI-based applications during the design-time phase. This concern is also extended to the run-time phase due to the high level of non-homogeneity between safety and artificial intelligence domains. One of the prominent strategies to deal with this type of issue is developing an external monitoring mechanism that preserves the safe operation of the system in run time. We provided a safety violation identification framework deployed on top of the CARLA simulator to tackle this issue during design time. This framework can identify the exact type of violation with the coordination of the incident along with the severity level and visualize the related details accordingly on the grid map of the simulator for safety engineer [259]. The provided information by the framework helps the safety engineers to understand the behavior of the intelligent driving agents in the driving environment, thus acquiring broader knowledge and more flexibility to apply and observe the outcome of the required adjustments in real-time. As a result, the safety measures in the development chain of AI-based applications will be enforced with more efficiency and effectiveness. The evaluated algorithms for this framework in our experiments were based on basic variants of reinforcement and imitation learning; however, due to the independent nature of the framework from the evaluated algorithms, the setup can be extended and utilized with more complex path planning methods as long as the evaluation remains in the CARLA simulator. Our work in the safety assurance domain was extended to the run-time phase by defining a novel monitoring architecture to preserve the safe action of the developed applications based on machine learning methods, more exclusively reinforcement learning approaches [186]. The concept of Crash Prediction Networks (CPN) was designed in this work by the ensemble of networks in CARLA simulator, observing and simultaneously being trained on a set of pre-defined safe operations of the application, and then being deployed on an actual driving agent to monitor its actions and preserve its safe operation accordingly [7]. The simple version of the CPN was also extended by

integrating spatio-temporal features and evaluated in different scenarios in the presence of static and dynamic obstacles. We also demonstrated that CPN is capable of dealing with uncertainty, one of the most challenging issues of machine learning, especially in the automotive domain, in the presence of an imbalanced dataset and respectively exhibiting acceptable performance in this regard [260]. The second part of our work in enabling context prediction architectures of the intelligent vehicle platforms was dedicated to emotional awareness and the impact of integrating behavior-based emotional indicators in multimodal recognition architectures. In order to identify and validate the common in-cabin emotional factors and determinants, we performed a preliminary empirical study with the help of an online survey distributed initially among 103 participants in the first round [211]. Afterward, the designed questionnaire was enhanced further to incorporate the participants' personal opinion regarding the projection of emotional states on their in-cabin behavior during driving and vice versa. In the second round, the survey reached 337 participants from different countries. The outcome of our empirical study on the distributed survey was utilized to design the behavioral profiles, which later were used in our experiments over the VIRES VTD simulator on more than 15 participants. The evaluated results and findings of our work on studying the steering wheel angular velocity and vehicle acceleration intensity, as the selected behavioral-based emotional indicators, demonstrate the positive impact on maintaining the robustness of the system and prediction outcome by integrating such indicators as different modules in an experimental multimodal emotion recognition system [139]. Utilizing these in-cabin behavior-based emotional signals improve the context prediction architectures of the intelligent vehicles that rely on the respective services. Therefore, it brings numerous opportunities for the applications, especially the machine learning-based ones, which require human user feedback for closing their decision-making loop. Preserving privacy is also one of the challenging aspects of each research work that deals directly with user-related data. This issue becomes even more critical in the automotive domain by the demand for the users' driving-related data during the experiments and following evaluations in safety and emotional awareness domains. To tackle this issue, we developed an application programming interface (API) with a user-friendly web interface to evaluate the developed models on different (privately collected) datasets over secure communication links without exposing the data directly or storing any information on the server. This API enables the researchers to access and utilize the previously developed models and evaluate them further with their own data without directly publishing or sharing the data with the model owners. The built-in tools of the web interface of the developed API can also provide a useful set of functions for normalizing and processing the data before evaluation. This API facilitates the collaboration between different research and development domains and can be extended in the future to support more modalities and features if required.

## 6.2 Discussion and Outlook

Our work in the safety domain considered a relatively passive role for the vehicle's driver and focused solely on preventive measures in design time. Most of the experiments in our work have been designed and evaluated in CARLA simulator, one of the state-of-the-art solutions and widely used simulators for autonomous driving applications because it provides a rich set of tools and libraries to support the required demands of the designed experiments for the safety domain and safety-critical applications. However, regardless of its high quality and richness, a simulated environment can not always be an entirely reliable substitute for the real world due to its apparent limitations. Hence, the necessity of acquiring real-world data should not be overlooked in future works, especially in the safety domain, during the training and the validation phase, for applications that must satisfy a high level of safety requirements.

Regarding the safe operation monitoring and uncertainty issue originating from machine learning methods and training data, we initially considered multiple safety-critical situations in autonomous driving that are vulnerable in this regard. First, we proposed the assuring candidates for monitoring approaches to preserve the safety of such a system. Then, we evaluated and examined the most promising candidate on the designed scenarios in the simulation environment. This work's developed application for the monitoring approach, CPN, has utilized the basic reinforcement learning algorithms in simplified simulation scenarios as the proof of concept. Extending the aforementioned safety monitoring approach during the training phase is necessary. This goal must be achieved by examining more complex path planning methods while adding further dynamics to the scenarios and the evaluation criteria to benchmark the capabilities of the monitoring system in identifying potential weak points and developing more generalized accurate models. It should be noted that applying just one particular technique is not enough to verify the functionality of adaptive software inside the vehicle, as each method has its own set of pros and cons. Instead, we need to focus on building a toolbox of different verification and validation techniques that can be applied based on specific needs and specifications of the system. We suggest using a layered approach in future works where each layer of monitoring for data and the application, independent of each other, focuses on one aspect of the safety requirement in target applications.

On the emotional awareness side, we also face a similar situation. The topic of emotions and affect recognition is complex by nature, and the involved factors can differ considerably subject-wise. In this work, we tried to lay our hypothesis on a very concrete set of definitions following a preliminary empirical study, which we have performed in advance. However, there are still valid concerns regarding the number of participants and their diversity in our empirical study. For example, the considerable positive value of skewness and a high value of kurtosis for the age group of the participants demonstrate an existing bias that must be overcome in future evaluations by incorporating diversity through utilizing more participants. This issue is also valid regarding the ethnicity of the participants and their country of origin and has undoubtedly been reflected in the simulator testbed-related experiments during data collection. Nevertheless, it is vivid that adding more variety to the groups of participants and increasing their number can

form a more representative backbone for further evaluations. Moreover, our experiments utilized a relatively low number of drivers due to resource limitations during evaluations; hence, we believe that increasing the number of participants is vital to develop more representative datasets and models, especially for behavior-based modalities, that can be generalized efficiently.

Last but not least, the developed API in this work can be extended further to support more modalities and provide real-time prediction on the live data feed. Besides, the efficient pre-processing of the vehicle data on the server side is still a considerable challenge due to the diverse range of existing components and the amount of generated signals inside a vehicle.

# A Appendix

**Figure A.1:** Functions and resources of the developed API

# Bibliography


[1] M. Machin, J. Guiochet, H. Waeselynck, J.-P. Blanquart, M. Roy, and L. Masson. Smof: A safety monitoring framework for autonomous systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(5):702–715, 2016.

[2] A. Paiva, I. Leite, and T. Ribeiro. Emotion modeling for social robots. *The Oxford handbook of affective computing*, pages 296–308, 2014.

[3] R. Plutchik. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350, 2001.

[4] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

[5] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency. Memory fusion network for multi-view sequential learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] A. B. Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018.

[7] S. Nair, S. Shafaei, S. Kugele, M. H. Osman, and A. Knoll. Monitoring safety of autonomous vehicles with crash prediction networks. In *SafeAI@ AAAI*, 2019.

[8] M. Pantic, G. Tzimiropoulos, and S. Zafeiriou. 300 faces in-the-wild challenge (300-w). In *ICCV Workshop*, volume 5, 2013.

[9] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, pages 94–101. IEEE, 2010.

[10] G. G. Slabaugh. Computing euler angles from a rotation matrix. *Retrieved on August*, 6(2000):39–63, 1999.

[11] A. Ortony and T. J. Turner. What's basic about basic emotions? *Psychological review*, 97(3):315, 1990.

[12] W. G. Parrott. *Emotions in social psychology: Essential readings*. psychology press, 2001.

[13] S. Shafaei, F. Mueller, T. Salzmann, M. H. Farzaneh, S. Kugele, and A. Knoll. Context prediction architectures in next generation of intelligent cars. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2923–2930. IEEE, 2018.

[14] S. international. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE International,(J3016)*, 2016.

[15] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[16] G. Rigoll. The connecteddrive context server–flexible software architecture for a context aware vehicle. *Advanced Microsystems for Automotive Applications 2007*, 201, 2007.

[17] M. H. Tran, A. Colman, and J. Han. Service-based development of context-aware automotive telematics systems. In *2010 15th IEEE International Conference on Engineering of Complex Computer Systems*, pages 53–62. IEEE, 2010.

[18] S. Loughran. Towards an adaptive and context aware laptop. *HP LABORATORIES TECHNICAL REPORT HPL*, (158), 2001.

[19] A. Boytsov. *Context reasoning, context prediction and proactive adaptation in pervasive computing systems*. PhD thesis, Luleå tekniska universitet, 2011.

[20] D. J. Cook. Prediction algorithms for smart environments. *Smart environments: Technologies, protocols, and applications*, pages 175–192, 2005.

[21] K. Kaowthumrong, J. Lebsack, and R. Han. Automated selection of the active device in interactive multi-device smart spaces. In *Workshop at UbiComp*, volume 2. Citeseer, 2002.

[22] J. Krumm. A markov model for driver turn prediction. 2016.

[23] D. W. Albrecht, I. Zukerman, and A. E. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User modeling and user-adapted interaction*, 8(1-2):5–47, 1998.

[24] J. Petzold, A. Pietzowski, F. Bagci, W. Trumler, and T. Ungerer. Prediction of indoor movements using bayesian networks. In *International Symposium on Location-and Context-Awareness*, pages 211–222. Springer, 2005.

[25] M. C. Mozer. The neural network house: An environment hat adapts to its inhabitants. In *Proc. AAAI Spring Symp. Intelligent Environments*, volume 58, 1998.

[26] A. Gellert and L. Vintan. Person movement prediction using hidden markov models. *Studies in Informatics and control*, 15(1):17, 2006.

[27] E. Al-Masri and Q. H. Mahmoud. A context-aware mobile service discovery and selection mechanism using artificial neural networks. In *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*, pages 594–598. ACM, 2006.

[28] W. Lin, W. Wu, and Q. Zhang. Handover strategy of smart mobile terminals among heterogeneous wireless networks. In *Proceedings of the 2008 International Conference on Advanced Infocomm Technology*, page 30. ACM, 2008.

[29] S. Sigg. *Development of a novel context prediction algorithm and analysis of context prediction schemes*. kassel university press GmbH, 2008.

[30] J. H. da Rosa, J. L. Barbosa, and G. D. Ribeiro. Oracon: An adaptive model for context prediction. *Expert Systems with Applications*, 45:56–70, 2016.

[31] N. Ye, A. Somani, D. Hsu, and W. S. Lee. Despot: Online pomdp planning with regularization. *Journal of Artificial Intelligence Research*, 58:231–266, 2017.

[32] S. Gelly and D. Silver. Combining online and offline knowledge in uct. In *Proceedings of the 24th international conference on Machine learning*, pages 273–280, 2007.

[33] R. Salay, R. Queiroz, and K. Czarnecki. An analysis of iso 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*, 2017.

[34] ISO. Road vehicles–Functional safety (ISO 26262), 2011.

[35] A. Wassyng and M. Lawford. Software tools for safety-critical software development. *International Journal on Software Tools for Technology Transfer*, 8(4-5):337–354, 2006.

[36] C.-H. Cheng, F. Diehl, Y. Hamza, G. Hinz, G. Nührenberg, M. Rickert, H. Ruess, and M. Troung-Le. Neural networks for safety-critical applications-challenges, experiments and perspectives. *arXiv preprint arXiv:1709.00911*, 2017.

[37] B. J. Taylor, M. A. Darrah, and C. D. Moats. Verification and validation of neural networks: a sampling of research in progress. In *Intelligent Computing: Theory and Applications*, volume 5103, pages 8–17. International Society for Optics and Photonics, 2003.

[38] B. J. Taylor. *Methods and procedures for the verification and validation of artificial neural networks*. Springer Science & Business Media, 2006.

[39] Y. Liu, T. Menzies, and B. Cukic. Data sniffing-monitoring of machine learning for online adaptive systems. In *Tools with Artificial Intelligence, 2002.(ICTAI 2002). Proceedings. 14th IEEE International Conference on*, pages 16–21. IEEE, 2002.

[40] M. L. Cunningham, M. A. Regan, et al. The impact of emotion, life stress and mental health issues on driving performance and safety. *Road & Transport Research: A Journal of Australian and New Zealand Research and Practice*, 25(3):40, 2016.

[41] F. Eyben, M. Woellmer, T. Poitschke, B. Schuller, C. Blaschke, B. Faerber, and N. Nguyen-Thien. Emotion on the road-necessity, acceptance, and feasibility of affective computing in the car. *Advances in human-computer interaction*, 2010, 2010.

[42] M. Wagner, A. Meroth, and D. Zoebel. Developing self-adaptive automotive systems. *Design Automation for Embedded Systems*, 18(3):199–221, 2014.

[43] K. Pei, Y. Cao, J. Yang, and S. Jana. Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*, 2017.

[44] S. Burton, L. Gauerhof, and C. Heinzemann. Making the case for safety of machine learning in highly automated driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 5–16. Springer, 2017.

[45] S. Russell, D. Dewey, and M. Tegmark. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.

[46] J. Taylor, E. Yudkowsky, P. LaVictoire, and A. Critch. Alignment for advanced machine learning systems. *Machine Intelligence Research Institute*, 2016.

[47] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[48] A. Kane, O. Chowdhury, A. Datta, and P. Koopman. A case study on runtime monitoring of an autonomous research vehicle (arv) system. In *Runtime Verification*, pages 102–117. Springer, 2015.

[49] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.

[50] C.-H. Cheng, F. Diehl, G. Hinz, Y. Hamza, G. Neuhrenberg, M. Rickert, H. Ruess, and M. Truong-Le. Neural networks for safety-critical applications-challenges, experiments and perspectives. In *2018 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1005–1006. IEEE, 2018.

[51] B. J. Taylor. Automated test generation for testing neural network systems. In *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*, pages 229–256. Springer, 2006.

[52] G. Bagschik, T. Menzel, and M. Maurer. Ontology based scene creation for the development of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1813–1820. IEEE, 2018.

[53] S. Ray. *Scalable techniques for formal verification*. Springer Science & Business Media, 2010.

[54] S. A. Seshia, D. Sadigh, and S. S. Sastry. Towards verified artificial intelligence. *arXiv preprint arXiv:1606.08514*, 2016.

[55] E. J. Fuller, S. K. Yerramalla, and B. Cukic. Stability properties of neural networks. In *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*, pages 97–108. Springer, 2006.

[56] S. Yerramalla, E. Fuller, M. Mladenovski, and B. Cukic. Lyapunov analysis of neural network stability in an adaptive flight control system. In *Symposium on Self-Stabilizing Systems*, pages 77–92. Springer, 2003.

[57] B. J. Taylor and M. A. Darrah. Rule extraction as a formal method for the verification and validation of neural networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 5, pages 2915–2920. IEEE, 2005.

[58] L. Fu. Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8):1114–1124, 1994.

[59] S. Thrun. Extracting rules from artificial neural networks with distributed representations. *Advances in neural information processing systems*, pages 505–512, 1995.

[60] J. R. Zilke, E. L. Mencia, and F. Janssen. Deepred–rule extraction from deep neural networks. In *International Conference on Discovery Science*, pages 457–473. Springer, 2016.

[61] U. Gasser and V. A. Almeida. A layered model for ai governance. *IEEE Internet Computing*, 21(6):58–62, 2017.

[62] B. Cukic, E. Fuller, M. Mladenovski, and S. Yerramalla. Run-time assessment of neural network control systems. In *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*, pages 257–269. Springer, 2006.

[63] M. Törngren, X. Zhang, N. Mohan, M. Becker, L. Svensson, X. Tao, D.-J. Chen, and J. Westman. Architecting safety supervisors for high levels of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1721–1728. IEEE, 2018.

[64] A. C. Rao, R. McMurran, and R. P. Jones. A critical analysis of model-based formal verification efforts within the automotive industry. *SAE international journal of passenger cars-electronic and electrical systems*, 1(2008-01-0220):77–83, 2008.

[65] W. Xiang, P. Musau, A. A. Wild, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, and T. T. Johnson. Verification for machine learning, autonomy, and neural networks survey. *arXiv preprint arXiv:1810.01989*, 2018.

[66] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. V. Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.

[67] R. Salay, R. Queiroz, and K. Czarnecki. An analysis of iso 26262: Using machine learning safely in automotive software. *arXiv preprint arXiv:1709.02435*, 2017.

[68] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[69] Y. Gal. Uncertainty in deep learning. 2016.

[70] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.

[71] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[72] R. M. Nesse and P. C. Ellsworth. Evolution, emotions, and emotional disorders. *American Psychologist*, 64(2):129, 2009.

[73] H. C. Breiter, N. L. Etcoff, P. J. Whalen, W. A. Kennedy, S. L. Rauch, R. L. Buckner, M. M. Strauss, S. E. Hyman, and B. R. Rosen. Response and habituation of the human amygdala during visual processing of facial expression. *Neuron*, 17(5):875–887, 1996.

[74] J. Armony and P. Vuilleumier. *The Cambridge handbook of human affective neuroscience*. Cambridge university press, 2013.

[75] X. Gu, P. R. Hof, K. J. Friston, and J. Fan. Anterior insular cortex and emotional awareness. *Journal of Comparative Neurology*, 521(15):3371–3388, 2013.

[76] J. T. Cacioppo and W. L. Gardner. Emotion. *Annual review of psychology*, 50(1):191–214, 1999.

[77] R. J. Davidson. Cognitive neuroscience needs affective neuroscience (and vice versa). *Brain and Cognition*, 42(1):89–92, 2000.

[78] M. Z. Soroush, K. Maghooli, S. K. Setarehdan, and A. M. Nasrabadi. A review on eeg signals based emotion recognition. *International Clinical Neuroscience Journal*, 4(4):118, 2017.

[79] L. G. Hernández Rojas, O. Martinez Mozos, J. M. Ferrández, and J. M. Antelis Ortiz. Eeg-based detection of braking intention under different car driving conditions. *Frontiers in neuroinformatics*, 12:29, 2018.

[80] S. Shimojo and L. Shams. Sensory modalities are not separate modalities: plasticity and interactions. *Current opinion in neurobiology*, 11(4):505–509, 2001.

[81] T. Zhang, A. H. Chan, and W. Zhang. Dimensions of driving anger and their relationships with aberrant driving. *Accident Analysis & Prevention*, 81:124–133, 2015.

[82] M. Chan and A. Singhal. The emotional side of cognitive distraction: Implications for road safety. *Accident Analysis & Prevention*, 50:147–154, 2013.

[83] M. A. Assari and M. Rahmati. Driver drowsiness detection using face expression recognition. In *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 337–341. IEEE, 2011.

[84] Drowsy driving. URL: `https://www.sleepfoundation.org/professionals/drowsy-driving`.

[85] A. Lotz, K. Ihme, A. Charnoz, P. Maroudis, I. Dmitriev, and A. Wendemuth. Recognizing behavioral factors while driving: A real-world multimodal corpus to monitor the driver's affective state. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[86] J. Lu, X. Xie, and R. Zhang. Focusing on appraisals: How and why anger and fear influence driving risk perception. *Journal of safety research*, 45:65–73, 2013.

[87] M. Walch, K. Lange, M. Baumann, and M. Weber. Autonomous driving: investigating the feasibility of car-driver handover assistance. In *Proceedings of the 7th international conference on automotive user interfaces and interactive vehicular applications*, pages 11–18, 2015.

[88] N. Du, F. Zhou, E. M. Pulver, D. M. Tilbury, L. P. Robert, A. K. Pradhan, and X. J. Yang. Examining the effects of emotional valence and arousal on takeover performance in conditionally automated driving. *Transportation research part C: emerging technologies*, 112:78–87, 2020.

[89] H.-J. Vögel, C. Süß, T. Hubregtsen, E. André, B. Schuller, J. Härri, J. Conradt, A. Adi, A. Zadorojniy, J. Terken, et al. Emotion-awareness for intelligent vehicle assistants: A research agenda. In *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, pages 11–15. IEEE, 2018.

[90] H. A. Meshram, M. G. Sonkusare, P. Acharya, and S. Prakash. Facial emotional expression regulation to control the semi-autonomous vehicle driving. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–5. IEEE, 2020.

[91] M. Egger, M. Ley, and S. Hanke. Emotion recognition from physiological signal analysis: a review. *Electronic Notes in Theoretical Computer Science*, 343:35–55, 2019.

[92] J. Sini, A. C. Marceddu, and M. Violante. Automatic emotion recognition for the calibration of autonomous driving functions. *Electronics*, 9(3):518, 2020.

[93] J. Sini, A. C. Marceddu, M. Violante, and R. Dessì. Passengers' emotions recognition to improve social acceptance of autonomous driving vehicles. In *Progresses in Artificial Intelligence and Neural Systems*, pages 25–32. Springer, 2021.

[94] M. Braun, B. Pfleging, and F. Alt. A survey to understand emotional situations on the road and what they mean for affective automotive uis. *Multimodal Technologies and Interaction*, 2(4):75, 2018.

[95] M. Maurer, J. Christian Gerdes, B. Lenz, and H. Winner. *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016.

[96] C. Nass, I.-M. Jonsson, H. Harris, B. Reaves, J. Endo, S. Brave, and L. Takayama. Improving automotive safety by pairing driver emotion and car voice emotion. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1973–1976, 2005.

[97] M. Weber. *Automotive emotions: a human-centred approach towards the measurement and understanding of drivers' emotions and their triggers*. PhD thesis, Brunel University London, 2018.

[98] P. Ekman. Facial expressions of emotion: an old controversy and new findings. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 335(1273):63–69, 1992.

[99] J. A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[100] L.-l. Chen, Y. Zhao, P.-f. Ye, J. Zhang, and J.-z. Zou. Detecting driving stress in physiological signals based on multimodal feature analysis and kernel classifiers. *Expert Systems with Applications*, 85:279–291, 2017.

[101] J. S. K. Ooi, S. A. Ahmad, H. R. Harun, Y. Z. Chong, and S. H. M. Ali. A conceptual emotion recognition framework: stress and anger analysis for car accidents. *International journal of vehicle safety*, 9(3):181–195, 2017.

[102] H. Salih and L. Kulkarni. Study of video based facial expression and emotions recognition methods. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 692–696. IEEE, 2017.

[103] K. Zhang, Y. Huang, Y. Du, and L. Wang. Facial expression recognition based on deep evolutional spatial-temporal networks. *IEEE Transactions on Image Processing*, 26(9):4193–4203, 2017.

[104] C. D. Katsis, G. Rigas, Y. Goletsis, and D. I. Fotiadis. Emotion recognition in car industry. *Emotion Recognition: A Pattern Analysis Approach*, pages 515–544, 2015.

[105] V. Govindarajan and R. Bajcsy. Human modeling for autonomous vehicles: Reachability analysis, online learning, and driver monitoring for behavior prediction. *Electrical Engineering and Computer Sciences University of California at Berkeley. Technical Report No. UCB/EECS-2017-226 http://www2. eecs. berkeley. edu/Pubs/TechRpts/2017/EECS-2017-226. html*, 2017.

[106] A. Mcmanus. Driver Emotion Recognition and Real Time Facial Analysis for the Automotive Industry, 2017. URL: `https://blog.affectiva.com/`.

[107] A. Mehrabian et al. *Silent messages*, volume 8. Wadsworth Belmont, CA, 1971.

[108] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1113–1133, 2014.

[109] S. K. D'mello and J. Kory. A review and meta-analysis of multimodal affect detection systems. *ACM Computing Surveys (CSUR)*, 47(3):1–36, 2015.

[110] S. Poria, E. Cambria, R. Bajpai, and A. Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.

[111] S. Poria, E. Cambria, A. Hussain, and G.-B. Huang. Towards an intelligent framework for multimodal affective data analysis. *Neural Networks*, 63:104–116, 2015.

[112] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88, 2016.

[113] K. R. Scherer. What are emotions? and how can they be measured? *Social science information*, 44(4):695–729, 2005.

[114] C. Harmon-Jones, B. Bastian, and E. Harmon-Jones. The discrete emotions questionnaire: A new tool for measuring state self-reported emotions. *PloS one*, 11(8):e0159915, 2016.

[115] P. Desmet. Measuring emotion: Development and application of an instrument to measure emotional responses to products. In *Funology*, pages 111–123. Springer, 2003.

[116] H. A. Elfenbein, A. A. Marsh, and N. Ambady. Emotional intelligence and the recognition of emotion from facial expressions. 2002.

[117] S. Kaplan, R. S. Dalal, and J. N. Luchman. Measurement of emotions. *Research methods in occupational health psychology: State of the art in measurement, design, and data analysis. New York: Routledge*, 2013.

[118] J. B. Torre and M. D. Lieberman. Putting feelings into words: Affect labeling as implicit emotion regulation. *Emotion Review*, 10(2):116–124, 2018.

[119] L. F. Barrett. Navigating the science of emotion. In *Emotion measurement*, pages 31–63. Elsevier, 2016.

[120] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1):259–275, 2003.

[121] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero. Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1548–1568, 2016.

[122] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[123] A. Fernández, R. Usamentiaga, J. L. Carús, and R. Casado. Driver distraction using visual-based sensors and algorithms. *Sensors*, 16(11):1805, 2016.

[124] P. Ekman and W. V. Friesen. Measuring facial movement. *Environmental psychology and nonverbal behavior*, 1(1):56–75, 1976.

[125] S. Zhao and R.-R. Grigat. Robust eye detection under active infrared illumination. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 481–484. IEEE, 2006.

[126] B. Cyganek and S. Gruszczynski. Eye recognition in near-infrared images for driver's drowsiness monitoring. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 397–402. IEEE, 2013.

[127] R. Reisenzein, M. Junge, M. Studtmann, and O. Huber. Observational approaches to the measurement of emotions. *International handbook of emotions in education*, pages 580–606, 2014.

[128] N. Dael, M. Mortillaro, and K. R. Scherer. Emotion expression in body action and posture. *Emotion*, 12(5):1085, 2012.

[129] S. Mota and R. W. Picard. Automated posture analysis for detecting learner's interest level. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, volume 5, pages 49–49. IEEE, 2003.

[130] M. N. Rastgoo, B. Nakisa, A. Rakotonirainy, V. Chandran, and D. Tjondronegoro. A critical review of proactive detection of driver stress levels based on multimodal measurements. *ACM Computing Surveys (CSUR)*, 51(5):1–35, 2018.

[131] K. S. Quigley, K. A. Lindquist, and L. F. Barrett. Inducing and measuring emotion and affect: Tips, tricks, and secrets. 2014.

[132] R. W. Picard. Automating the recognition of stress and emotion: From lab to real-world impact. *IEEE MultiMedia*, 23(3):3–7, 2016.

[133] M. A. Tischler, C. Peter, M. Wimmer, and J. Voskamp. Application of emotion recognition methods in automotive research. In *Proceedings of the 2nd Workshop on Emotion and Computing-Current Research and Future Impact*, volume 1, pages 55–60, 2007.

[134] S. Jerritta, M. Murugappan, R. Nagarajan, and K. Wan. Physiological signals based human emotion recognition: a review. In *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, pages 410–415. IEEE, 2011.

[135] H. T. Reis, H. T. Reis, C. M. Judd, et al. *Handbook of research methods in social and personality psychology*. Cambridge University Press, 2000.

[136] J. Kim and E. André. Emotion recognition based on physiological changes in music listening. *IEEE transactions on pattern analysis and machine intelligence*, 30(12):2067–2083, 2008.

[137] V. L. Kinner, L. Kuchinke, A. M. Dierolf, C. J. Merz, T. Otto, and O. T. Wolf. What our eyes tell us about feelings: Tracking pupillary responses during emotion regulation processes. *Psychophysiology*, 54(4):508–518, 2017.

[138] Y.-G. Cherng, T. Baird, J.-T. Chen, and C.-A. Wang. Background luminance effects on pupil size associated with emotion and saccade preparation. *Scientific reports*, 10(1):1–11, 2020.

[139] S. Shafaei, T. Hacizade, and A. Knoll. Integration of driver behavior into emotion recognition systems: a preliminary study on steering wheel and vehicle acceleration. In *Asian Conference on Computer Vision*, pages 386–401. Springer, 2018.

[140] V. Vapnik, I. Guyon, and T. Hastie. Support vector machines. *Mach. Learn*, 20(3):273–297, 1995.

[141] J. Zhang, Z. Yin, P. Chen, and S. Nichele. Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. *Information Fusion*, 59:103–126, 2020.

[142] P. J. Bota, C. Wang, A. L. Fred, and H. P. Da Silva. A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals. *IEEE Access*, 7:140990–141020, 2019.

[143] M. S. Kankanhalli, J. Wang, and R. Jain. Experiential sampling on multiple data streams. *IEEE transactions on multimedia*, 8(5):947–955, 2006.

[144] F.-J. Huang and Y. LeCun. Large-scale learning with svm and convolutional nets for generic object categorization. In *Proc. Computer Vision and Pattern Recognition Conference (CVPR'06)*, 2006.

[145] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[146] W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

[147] K. Sikka, K. Dykstra, S. Sathyanarayana, G. Littlewort, and M. Bartlett. Multiple kernel learning for emotion recognition in the wild. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 517–524. ACM, 2013.

[148] J. Chen, Z. Chen, Z. Chi, and H. Fu. Emotion recognition in the wild with feature fusion and multiple kernel learning. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 508–513. ACM, 2014.

[149] N. Jaques, S. Taylor, A. Sano, and R. Picard. Multi-task, multi-kernel learning for estimating individual wellbeing. In *Proc. NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*, volume 898, 2015.

[150] M. Sunagawa, S.-i. Shikii, W. Nakai, M. Mochizuki, K. Kusukame, and H. Kitajima. Comprehensive drowsiness level detection model combining multimodal information. *IEEE Sensors Journal*, 20(7):3709–3717, 2019.

[151] F. Rahdari, E. Rashedi, and M. Eftekhari. A multimodal emotion recognition system using facial landmark analysis. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 43(1):171–189, 2019.

[152] Y. Huang, J. Yang, P. Liao, and J. Pan. Fusion of facial expressions and eeg for multimodal emotion recognition. *Computational intelligence and neuroscience*, 2017, 2017.

[153] S. Bianco, P. Napoletano, and R. Schettini. Multimodal car driver stress recognition. In *Proceedings of the 13th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 302–307, 2019.

[154] W.-L. Zheng, B.-N. Dong, and B.-L. Lu. Multimodal emotion recognition using eeg and eye tracking data. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5040–5043. IEEE, 2014.

[155] W.-L. Zheng, W. Liu, Y. Lu, B.-L. Lu, and A. Cichocki. Emotionmeter: A multimodal framework for recognizing human emotions. *IEEE transactions on cybernetics*, 49(3):1110–1122, 2018.

[156] J.-J. Guo, R. Zhou, L.-M. Zhao, and B.-L. Lu. Multimodal emotion recognition from eye image, eye movement and eeg using deep neural networks. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3071–3074. IEEE, 2019.

[157] H. Ranganathan, S. Chakraborty, and S. Panchanathan. Multimodal emotion recognition using deep learning architectures. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[158] H. Tang, W. Liu, W.-L. Zheng, and B.-L. Lu. Multimodal emotion recognition using deep neural networks. In *International Conference on Neural Information Processing*, pages 811–819. Springer, 2017.

[159] L.-H. Du, W. Liu, W.-L. Zheng, and B.-L. Lu. Detecting driving fatigue with multimodal deep learning. In *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 74–77. IEEE, 2017.

[160] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou. End-to-end multimodal emotion recognition using deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1301–1309, 2017.

[161] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain. Convolutional mkl based multimodal emotion recognition and sentiment analysis. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 439–448. IEEE, 2016.

[162] M. N. Rastgoo, B. Nakisa, F. Maire, A. Rakotonirainy, and V. Chandran. Automatic driver stress level classification using multimodal deep learning. *Expert Systems with Applications*, 138:112793, 2019.

[163] H.-T. Choi, M.-K. Back, and K.-C. Lee. Driver drowsiness detection based on multimodal using fusion of visual-feature and bio-signal. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1249–1251. IEEE, 2018.

[164] Y. Shu and S. Wang. Emotion recognition through integrating eeg and peripheral signals. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2871–2875. IEEE, 2017.

[165] N. S. Karuppusamy and B.-Y. Kang. Multimodal system to detect driver fatigue using eeg, gyroscope, and image processing. *IEEE Access*, 8:129645–129667, 2020.

[166] K. P. Seng, L.-M. Ang, and C. S. Ooi. A combined rule-based & machine learning audio-visual emotion recognition approach. *IEEE Transactions on Affective Computing*, 9(1):3–13, 2016.

[167] V. Chaparro, A. Gomez, A. Salgado, O. L. Quintero, N. Lopez, and L. F. Villa. Emotion recognition from eeg and facial expressions: a multimodal approach. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 530–533. IEEE, 2018.

[168] J. A. Domínguez-Jiménez, K. C. Campo-Landines, J. C. Martínez-Santos, E. J. Delahoz, and S. H. Contreras-Ortiz. A machine learning model for emotion recognition from physiological signals. *Biomedical signal processing and control*, 55:101646, 2020.

[169] D. Matsumoto. American-japanese cultural differences in the recognition of universal facial expressions. *Journal of cross-cultural psychology*, 23(1):72–84, 1992.

[170] L. Shu, J. Xie, M. Yang, Z. Li, Z. Li, D. Liao, X. Xu, and X. Yang. A review of emotion recognition using physiological signals. *Sensors*, 18(7):2074, 2018.

[171] D. A. Trevisan, M. Bowering, and E. Birmingham. Alexithymia, but not autism spectrum disorder, may be related to the production of emotional facial expressions. *Molecular autism*, 7(1):1–12, 2016.

[172] S. Boonmee and P. Tangamchit. Portable reckless driving detection system. In *2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 1, pages 412–415. IEEE, 2009.

[173] W. Li, Y. Cui, Y. Ma, X. Chen, G. Li, G. Zeng, G. Guo, and D. Cao. A spontaneous driver emotion facial expression (defe) dataset for intelligent vehicles: Emotions triggered by video-audio clips in driving scenarios. *IEEE Transactions on Affective Computing*, 2021.

[174] R. A. Naqvi, M. Arsalan, A. Rehman, A. U. Rehman, W.-K. Loh, and A. Paul. Deep learning-based drivers emotion classification system in time series data for remote applications. *Remote Sensing*, 12(3):587, 2020.

[175] A. F. Requardt, K. Ihme, M. Wilbrink, and A. Wendemuth. Towards affect-aware vehicles for increasing safety and comfort: recognising driver emotions from audio recordings in a realistic driving study. *IET Intelligent Transport Systems*, 14(10):1265–1277, 2020.

[176] A. Leone, A. Caroppo, A. Manni, and P. Siciliano. Vision-based road rage detection framework in automotive safety applications. *Sensors*, 21(9):2942, 2021.

[177] General Motors. 2018 self-driving car report, 2018. URL: `https://www.gm.com/content/dam/company/docs/us/en/gmcom/gmsafetyreport.pdf`.

[178] Statistische Aemter des Bundes und der Laender. Unfallatlas — kartenanwendung. `https://unfallatlas.statistikportal.de/`, 2019. (Accessed on 03/26/2019).

[179] Agilysis. Crashmap - uk road safety map. `https://www.crashmap.co.uk/`, 2019. (Accessed on 03/26/2019).

[180] J. Eggert. Predictive risk estimation for intelligent adas functions. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 711–718. IEEE, 2014.

[181] L. González, E. Martí, I. Calvo, A. Ruiz, and J. Pérez. Towards risk estimation in automated vehicles using fuzzy logic. In *International Conference on Computer Safety, Reliability, and Security*, pages 278–289. Springer, 2018.

[182] S. Hallerbach, Y. Xia, U. Eberle, and F. Koester. Simulation-based identification of critical scenarios for cooperative and automated vehicles. *SAE International Journal of Connected and Automated Vehicles*, 1(2018-01-1066):93–106, 2018.

[183] M. M. Morando, Q. Tian, L. T. Truong, and H. L. Vu. Studying the safety impact of autonomous vehicles using simulation-based surrogate safety measures. *Journal of advanced transportation*, 2018, 2018.

[184] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*.

[185] NVIDIA. Safety force field. https://www.nvidia.com/en-us/self-driving-cars/safety-force-field/, 2019. (Accessed on 06/19/2019).

[186] S. Shafaei, S. Kugele, M. H. Osman, and A. Knoll. Uncertainty in machine learning: A safety perspective on autonomous driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 458–464. Springer, 2018.

[187] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[188] M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt. Variational inference for on-line anomaly detection in high-dimensional time series. *arXiv preprint arXiv:1602.07109*, 2016.

[189] M. Feld and C. Müller. The automotive ontology: managing knowledge inside the vehicle and sharing it between cars. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 79–86. ACM, 2011.

[190] L. Fridman, L. Ding, B. Jenik, and B. Reimer. Arguing machines: Human supervision of black box ai systems that make life-critical decisions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[191] C. Pecher, C. Lemercier, and J.-M. Cellier. The influence of emotions on driving behaviour. *Traffic Psychology and driving behaviour, New-York: Hindawi Publishers*, 2009.

[192] E. Roidl, B. Frehse, and R. Hoeger. Emotional states of drivers and the impact on speed, acceleration and traffic violations-a simulator study. *Accident Analysis and Prevention*, 70:282–292, 2014.

[193] J. Izquierdo-Reyes, R. A. Ramirez-Mendoza, M. R. Bustamante-Bello, S. Navarro-Tuch, and R. Avila-Vazquez. Advanced driver monitoring for assistance system (admas). *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 12(1):187–197, 2018.

[194] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[195] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.

[196] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[197] D. Bandyopadhyay and J. Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.

[198] S. Kumari and S. K. Rath. Performance comparison of soap and rest based web services for enterprise application integration. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1656–1660. IEEE, 2015.

[199] P. K. Potti, S. Ahuja, K. Umapathy, and Z. Prodanoff. Comparing performance of web service interaction styles: Soap vs. rest. In *Proceedings of the Conference on Information Systems Applied Research ISSN*, volume 2167, page 1508, 2012.

[200] T. Aihkisalo and T. Paaso. Latencies of service invocation and processing of the rest and soap web service interfaces. In *2012 IEEE Eighth World Congress on Services*, pages 100–107. IEEE, 2012.

[201] L. Richardson and S. Ruby. *RESTful web services.* " O'Reilly Media, Inc.", 2008.

[202] L. Xiao-Hong. Research and development of web of things system based on rest architecture. In *2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications*, pages 744–747. IEEE, 2014.

[203]

[204] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[205] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[206] Y. Zhao, J. Gao, and X. Yang. A survey of neural network ensembles. In *2005 International Conference on Neural Networks and Brain*, volume 1, pages 438–442. IEEE, 2005.

[207] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[208] W. Saunders, G. Sastry, A. Stuhlmueller, and O. Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[209] W. G. Najm, J. D. Smith, M. Yanagisawa, et al. Pre-crash scenario typology for crash avoidance research. Technical report, United States. National Highway Traffic Safety Administration, 2007.

[210] B. J. Higgs. *Emotional Impacts on Driver Behavior: An Emo-Psychophysical Car-Following Model*. PhD thesis, Virginia Tech, 2014.

[211] M. Kuscu, S. Shafaei, and A. Knoll. Abnormal driver behavior detection for automated emotion recognition (poster). 2018.

[212] W. Morales-Alvarez, O. Sipele, R. Léberon, H. H. Tadjine, and C. Olaverri-Monreal. Automated driving: a literature review of the take over request in conditional automation. *Electronics*, 9(12):2087, 2020.

[213] J. Gao and G. A. Davis. Using naturalistic driving study data to investigate the impact of driver distraction on driver's brake reaction time in freeway rear-end events in car-following situation. *Journal of safety research*, 63:195–204, 2017.

[214] L. Malta, P. Angkititrakul, C. Miyajima, and K. Takeda. Multi-modal real-world driving data collection, transcription, and integration using bayesian network. In *2008 IEEE Intelligent Vehicles Symposium*, pages 150–155. IEEE, 2008.

[215] S. Zepf, J. Hernandez, A. Schmitt, W. Minker, and R. W. Picard. Driver emotion recognition for intelligent vehicles: a survey. *ACM Computing Surveys (CSUR)*, 53(3):1–30, 2020.

[216] N. C. Fung, B. Wallace, A. D. Chan, R. Goubran, M. M. Porter, S. Marshall, and F. Knoefel. Driver identification using vehicle acceleration and deceleration events from naturalistic driving of older drivers. In *2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 33–38. IEEE, 2017.

[217] V. V. T. D. Vires simulationstechnologie gmbh. URL: `https://vires.com/company/`.

[218] D. E. King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[219] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE, 2011.

[220] E. Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[221] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[222] T. Senechal, D. McDuff, and R. Kaliouby. Facial action unit detection using active learning and an efficient non-linear kernel approximation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015.

[223] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[224] S. Ouellet. Real-time emotion recognition for gaming using deep convolutional network features. *arXiv preprint arXiv:1408.3750*, 2014.

[225] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. Bmva Press, 2014.

[226] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *2012 IEEE Conference on computer vision and pattern recognition*, pages 1822–1829. IEEE, 2012.

[227] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *2012 IEEE Conference on Computer vision and pattern recognition*, pages 1838–1845. IEEE, 2012.

[228] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2015.

[229] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2427–2434, 2013.

[230] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2387–2400, 2013.

[231] M. Amin, S. S. Nasir, M. B. I. Reaz, M. Ali, A. Mohd, T.-G. Chang, et al. Preference and placement of vehicle crash sensors. *Tehnički vjesnik*, 21(4):889–896, 2014.

[232] G. K. Balachandran, V. P. Petkov, T. Mayer, and T. Balslink. A 3-axis gyroscope for electronic stability control with continuous self-test. *IEEE Journal of Solid-State Circuits*, 51(1):177–186, 2015.

[233] S. Colton. The balance filter: a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform. *Chief Delphi white paper*, 1, 2007.

[234] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel. A complementary filter for attitude estimation of a fixed-wing uav. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 340–345. IEEE, 2008.

[235] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[236] D. L. Iverson. Inductive system health monitoring. In *IC-AI*, pages 605–611. Citeseer, 2004.

[237] M. C. Chuah and F. Fu. Ecg anomaly detection via time series analysis. In *International Symposium on Parallel and Distributed Processing and Applications*, pages 123–135. Springer, 2007.

[238] R. El Sibai, Y. Chabchoub, J. Demerjian, Z. Kazi-Aoul, and K. Barbar. Sampling algorithms in data stream environments. In *2016 International Conference on Digital Economy (ICDEc)*, pages 29–36. IEEE, 2016.

[239] Z. Zhang, Y. Asakawa, T. Imamura, and T. Miyake. Experiment design for measuring driver reaction time in driving situation. In *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3699–3703. IEEE Computer Society, 2013.

[240] J. Khashbat, T. Tsevegjav, J. Myagmarjav, I. Bazarragchaa, A. Erdenetuya, and N. Munkhzul. Determining the driver's reaction time in the stationary and real-life environments (comparative study). In *Strategic Technology (IFOST), 2012 7th International Forum on*, pages 1–3. IEEE, 2012.

[241] T. Krotak and M. Simlova. The analysis of the acceleration of the vehicle for assessing the condition of the driver. In *2012 IEEE Intelligent Vehicles Symposium*, pages 571–576. IEEE, 2012.

[242] E. A. Kensinger. Remembering emotional experiences: The contribution of valence and arousal. *Reviews in the Neurosciences*, 15(4):241–252, 2004.

[243] J. F. Guerrero Rázuri, A. Larsson, D. Sundgren, I. Bonet, and A. Moran. Recognition of emotions by the emotional feedback through behavioral human poses. *International Journal of Computer Science Issues*, 12(1):7–17, 2015.

[244] W. Li, J. Hou, and L. Yin. A classifier fusion method based on classifier accuracy. In *2014 International Conference on Mechatronics and Control (ICMC)*, pages 2119–2122. IEEE, 2014.

[245] R. A. Khan, A. Meyer, H. Konik, and S. Bouakaz. Human vision inspired framework for facial expressions recognition. In *2012 19th IEEE international conference on image processing*, pages 2593–2596. IEEE, 2012.

[246] M. M. Donia, A. A. Youssif, and A. Hashad. Spontaneous facial expression recognition based on histogram of oriented gradients descriptor. *Computer and Information Science*, 7(3):31–37, 2014.

[247] H. Alshamsi, H. Meng, and M. Li. Real time facial expression recognition app development on mobile phones. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 1750–1755. IEEE, 2016.

[248] W. Swinkels, L. Claesen, F. Xiao, and H. Shen. Svm point-based real-time emotion detection. In *2017 IEEE Conference on Dependable and Secure Computing*, pages 86–92. IEEE, 2017.

[249] J. S. K. Ooi, S. A. Ahmad, Y. Z. Chong, S. H. M. Ali, G. Ai, and H. Wagatsuma. Driver emotion recognition framework based on electrodermal activity measurements during simulated driving conditions. In *Biomedical Engineering and Sciences (IECBES), 2016 IEEE EMBS Conference on*, pages 365–369. IEEE, 2016.

[250] R. Theagarajan, B. Bhanu, A. Cruz, B. Le, and A. Tambo. Novel representation for driver emotion recognition in motor vehicle videos. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 810–814. IEEE, 2017.

[251] A. Tawari and M. Trivedi. Speech based emotion classification framework for driver assistance system. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 174–178. IEEE, 2010.

[252] C. M. Jones and I.-M. Jonsson. Automatic recognition of affective cues in the speech of car drivers to allow appropriate responses. In *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future*, pages 1–10. Computer-Human Interaction Special Interest Group (CHISIG) of Australia, 2005.

[253] M. Ali, F. Al Machot, A. H. Mosa, and K. Kyamakya. Cnn based subject-independent driver emotion recognition system involving physiological signals for adas. In *Advanced Microsystems for Automotive Applications 2016*, pages 125–138. Springer, 2016.

[254] D. Datcu and L. Rothkrantz. Multimodal recognition of emotions in car environments. *DCI&I 2009*, 2009.

[255] S. Hoch, F. Althoff, G. McGlaun, and G. Rigoll. Bimodal fusion of emotional data in an automotive environment. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2, pages ii–1085. IEEE, 2005.

[256] S. Wilf. Session management over a stateless protocol, December 17 2002. US Patent 6,496,824.

[257] G. Singh. A study of encryption algorithms (rsa, des, 3des and aes) for information security. *International Journal of Computer Applications*, 67(19), 2013.

[258] P. Mahajan and A. Sachdeva. A study of encryption algorithms aes, des and rsa for security. *Global Journal of Computer Science and Technology*, 2013.

[259] L. Heinzmann, S. Shafaei, M. H. Osman, C. Segler, and A. Knoll. A framework for safety violation identification and assessment in autonomous driving. In *AISafety@ IJCAI*, 2019.

[260] S. Nair, S. Shafaei, D. Auge, and A. Knoll. An evaluation of crash prediction networks (cpn) for autonomous driving scenarios in carla simulator. 2021.