

Universidad de Alcalá

PhD. Program in Electronics: Advanced Electronic Systems. Intelligent Systems

Driver Attention based on Deep Learning for a Smart Vehicle to Driver (V2D) Interaction

PhD. Thesis Presented by

Javier Araluce Ruiz

2023



Universidad de Alcalá

PhD. Program in Electronics: Advanced Electronic Systems. Intelligent Systems

Driver Attention based on Deep Learning for a Smart Vehicle to Driver (V2D) Interaction

PhD. Thesis Presented by

Javier Araluce Ruiz

Advisors

Dr. Luis M. Bergasa Pascual

Dr. Manuel Ocaña Miguel

Alcalá de Henares, 23th of March, 2023

A mi abuelo José, por su fuerza incondicional y ejemplo de superación.

Agradecimientos

Esta tesis llega a su fin y no quería terminar sin agradecer a todas las personas que me han ayudado a conseguir este hito en mi carrera profesional. Que ha sido como una montaña rusa, lleno de bajadas y subidas pero que ahora en retrospectiva todo cobra sentido y ves el porqué de las cosas.

En primer lugar, quiero agradecer a Irene, por estar ahí durante este largo camino y ser mi revisora de confianza para todos mis trabajos. Tu criterio me ha hecho mejorar y tu inmensa ayuda en los últimos momentos de agobio han conseguido que el libro cobre forma y sea lo que es hoy en día. Gracias por ayudarme y ser mi apoyo incondicional.

A continuación, quiero agradecer a mis tutores Luis Miguel y Manuel, que me han ayudado en todo momento, con sabios consejos y han conseguido que un chaval que estaba recién salido de la carrera con unas vagas ideas sobre su futuro cambie completamente y descubra un mundo que le apasiona y con un recorrido profesional a la vista bastante prometedor. Siento haber sido tan duro de revisar en ciertos momentos, jejeje.

No me quiero olvidar de los que han hecho que esté donde estoy, mis padres y hermanos. Muchas gracias por no discutir ninguna de mis decisiones y dejarme hacer mi camino a mi manera. Gracias por proporcionarme la tranquilidad que se respiraba en casa.

En estos agradecimientos no pueden faltar mis compañeros de laboratorio, estos cinco años de investigación desde que empecé con el trabajo final de grado hasta ahora dan para muchos compañeros y muchas historias, que, aunque el COVID no ha permitido estar siempre presencial, nos las hemos apañado para apoyarnos y crecer juntos consiguiendo hacer crecer a Robesafe. Por eso gracias, Carlos, Felipe, Rodri, Alex, Javi, Óscar, Edu, Santi, Miguel y muchos más, pero en algún punto tengo que parar. Gracias y a los que continuáis, seguir investigando y descubriendo vuestro camino que cuando miréis para atrás estoy convencido que os veréis mejores profesionales y lo más importante, mejores personas.

A mis amigos, que proporcionaban momentos de respiro sin pensar en el trabajo para poder despejarme y recargar las pilas y poder volver al trabajo con energías renovadas.

A los profesores de Robesafe, que compartían su conocimiento y curiosidades, haciendo crecer mi inquietud, que, desde mi punto de vista, es la mayor cualidad que se puede tener para investigar.

A todos esos revisores anónimos que ayudan a que los investigadores puedan aprender con sus valiosas críticas constructivas. Hablando de anónimos, a todas esas personas que han escrito valiosos comentarios en Internet que me han ayudado a aprender de mis errores.

Por último, a mi abuelo, que era la persona que más ilusión le hacía ver este momento y que por cosas de la vida no lo vamos a poder disfrutar juntos, pero no te preocupes que lo celebraremos en algún momento.

Resumen

La atención del conductor es un tópico interesante dentro del mundo de los vehículos inteligentes para la consecución de tareas que van desde la monitorización del conductor hasta la conducción autónoma. Esta tesis aborda este tópico basándose en algoritmos de aprendizaje profundo para conseguir una interacción inteligente entre el vehículo y el conductor.

La monitorización del conductor requiere una estimación precisa de su mirada en un entorno 3D para conocer el estado de su atención. En esta tesis se aborda este problema usando una única cámara, para que pueda ser utilizada en aplicaciones reales, sin un alto coste y sin molestar al conductor. La herramienta desarrollada ha sido evaluada en una base de datos pública (DADA2000), obteniendo unos resultados similares a los obtenidos mediante un seguidor de ojos caro que no puede ser usado en un vehículo real. Además, ha sido usada en una aplicación que evalúa la atención del conductor en la transición de modo autónomo a manual de forma simulada, proponiendo el uso de una métrica novedosa para conocer el estado de la situación del conductor en base a su atención sobre los diferentes objetos de la escena.

Por otro lado, se ha propuesto un algoritmo de estimación de atención del conductor, utilizando las últimas técnicas de aprendizaje profundo como son las *conditional Generative Adversarial Networks (cGANs)* y el *Multi-Head Self-Attention*. Esto permite enfatizar ciertas zonas de la escena al igual que lo haría un humano. El modelo ha sido entrenado y validado en dos bases de datos públicas (BDD-A y DADA2000) superando a otras propuestas del estado del arte y consiguiendo unos tiempos de inferencia que permiten su uso en aplicaciones reales.

Por último, se ha desarrollado un modelo que aprovecha nuestro algoritmo de atención del conductor para comprender una escena de tráfico obteniendo la decisión tomada por el vehículo y su explicación, en base a las imágenes tomadas por una cámara situada en la parte frontal del vehículo. Ha sido entrenado en una base de datos pública (BDD-OIA) proponiendo un modelo que entiende la secuencia temporal de los eventos usando un *Transformer Encoder*, consiguiendo superar a otras propuestas del estado del arte. Además de su validación en la base de datos, ha sido implementado en una aplicación que interacciona con el conductor aconsejando sobre las decisiones a tomar y sus explicaciones ante diferentes casos de uso en un entorno simulado.

Esta tesis explora y demuestra los beneficios de la atención del conductor para el mundo de los vehículos inteligentes, logrando una interacción vehículo conductor a través de las últimas técnicas de aprendizaje profundo.

Palabras clave: Atención del conductor, Vehículos autónomos, Sistemas avanzados de ayuda a la conducción, Interacción conductor vehículo, Redes neuronales, Transformers, simulador CARLA.

Abstract

Driver attention is a hot topic in the world of intelligent vehicles to achieve tasks ranging from driver monitoring to autonomous driving. This thesis addresses this topic based on deep learning algorithms to achieve an intelligent interaction between the vehicle and the driver.

Driver monitoring requires an accurate estimation of the driver's gaze in a 3D environment in order to know the driver's state of attention. This thesis manages this problem using a single camera that we can use in real applications without a high cost and without disturbing the driver. We evaluate the developed tool in a public dataset (DADA2000), achieving on-par results with an expensive, intrusive and difficult to install in a vehicle eye-tracker. Furthermore, we use it in an application that assesses the drivers' attention in the transition from autonomous to manual mode in a simulated way, proposing a novel metric to know the state of the drivers' situation based on their attention to different objects in the scene.

On the other hand, we propose a driver attention estimation algorithm using the latest deep learning techniques such as conditional Generative Adversarial Network (cGAN) and Multi-Head Self-Attention. It emphasises certain regions of the scene just as a human would. The model has been trained and validated in two public datasets (BDD-A and DADA2000), outperforming other state-of-the-art proposals and achieving inference times that allow its use in real applications.

Moreover, we develop a model that leverages our driver attention algorithm to understand a traffic scene by obtaining the decision taken by the vehicle and its explanation, based on the images taken by a camera located at the front of the car. We have trained it on a public database (BDD-OIA), proposing a model that understands the temporal sequence of events using a Transformer Encoder, managing to outperform other state-of-the-art proposals. Finally, to its validation in the dataset, we have implemented the model in an application that interacts with the driver advising about the decisions to take and their explanations for different use cases in a simulated environment.

This thesis explores and demonstrates the benefits of driver attention for the world of intelligent vehicles by achieving vehicle-to-driver interaction through the latest deep learning techniques.

Keywords: Driver attention, Self-driving vehicles, Advanced Driver Assistance Systems, Vehicle-to-driver (V2D) interaction, Neuronal networks, Transformers, CARLA simulator.

Contents

Agradecimientos vii

Resumen ix

Abstract xi

Contents xiii

List of Figures xix

List of Tables xxiii

1. Introduction 1

1.1. Aim 1

1.2. Background 2

1.2.1. Driver Attention Estimation for Driving Monitoring 3

1.2.2. Driver Awareness Estimation 5

1.2.3. Modelling Visual Attention in the Traffic Scene 6

1.2.4. Attention for Self-Driving Vehicles 7

1.2.5. V2D Applications 9

1.3. Motivation 10

1.4. Contributions of Dissertation 10

1.5. Organization of Dissertation 11

2. Driver Evaluation System 13

2.1. Introduction 13

2.2. Related Works 14

2.3. System Architecture 15

2.3.1. Video Acquisition . . . 16
2.3.2. Gaze Estimation . . . 16
2.3.3. Gaze Focalization Model . . . 17
2.3.3.1. OLS . . . 19
2.3.3.2. ERR . . . 20
2.3.3.3. FROLS . . . 21
2.3.3.3.1. Step 1. Data Collection . . . 21
2.3.3.3.2. Step 2. Defining the Modelling Framework . . . 21
2.3.3.3.3. Step 3. Determination of the Regressor Vector . . . 22
2.3.3.3.4. Step 4. Choosing the First Element . . . 22
2.3.3.3.5. Step 5. Selecting the Next Elements of the Model 23
2.3.3.4. Modelling Process Analysis . . . 24
2.3.3.5. Determination of the Final Model . . . 24
2.3.3.6. Calibration Procedure . . . 25
2.3.4. Heat Map Visualisation . . . 26
2.3.5. Driver Situation Awareness (DSA) . . . 26
2.4. Experimental Results . . . 29
2.4.1. Visual Focalization Model Evaluation . . . 29
2.4.2. Camera Parameters . . . 29
2.4.3. Camera Position . . . 30
2.4.4. Glasses . . . 31
2.4.5. Precision Test . . . 31
2.4.6. DADA2000 Evaluation . . . 32
2.4.6.1. Crash Object Detection in DADA2000 Benchmark . . . 33
2.4.6.2. Heat Focalization Map on DADA2000 . . . 36
2.5. Chapter Summary . . . 37

3. ARAGAN: A dRiver Attention estimation model based on conditional Generative Adversarial Network 39

3.1. Introduction . . . 39
3.2. Related Works . . . 41
3.2.1. Driver Attention . . . 41
3.2.2. Deep Learning Techniques . . . 42

3.3. ARAGAN Architecture 43
3.3.1. Adversarial Learning Procedure 44
3.3.2. Attention Modules 46
3.3.2.1. Soft-Attention 46
3.3.2.2. Hard-Attention 47
3.3.2.3. Convolutional Block Attention Module (CBAM) 47
3.3.2.4. Self-Attention Module 48
3.3.2.5. Multi-Head Self-Attention Module 50
3.3.3. Generator Architecture 51
3.3.4. Discriminator Architecture 54
3.4. Evaluation Metrics 54
3.5. Experimental Results 55
3.5.1. Ablation Study 56
3.5.2. Comparison with Other State-of-the-art Models 57
3.6. Chapter Summary 58

4. Driver's Attention-based Explainable Decision-Making 59
4.1. Introduction 59
4.2. Related Works 61
4.2.1. Datasets 61
4.2.1.1. BDD-OIA 61
4.2.1.2. BDD-X 62
4.2.1.3. DoTA 62
4.2.1.4. CTA 63
4.2.1.5. HDD 63
4.2.1.6. BDD-A Extended 64
4.2.2. State-of-the-art Models Trained in BDD-OIA 64
4.3. Theoretical Background 65
4.3.1. Feature Extractor 66
4.3.2. Attention Mechanisms 67
4.3.2.1. Self-Attention 67
4.3.2.2. Cross-Attention 68
4.3.2.3. Multi-Head Attention 68

4.3.2.4. Transformer Encoder 69
4.4. Dataset BDD-OIA 71
4.5. Driver's Attention-based Explainable Decision-Making Model 73
4.5.1. Feature Extractor 74
4.5.2. Driver's Attention Fusion based on ARAGAN 74
4.5.3. Spatio-Temporal Transformer Encoder 75
4.5.4. Classification Heads 77
4.6. Experimental Setup 77
4.6.1. Loss Functions 77
4.6.2. Metrics 79
4.7. Experimental Results 81
4.7.1. Ablation Study 81
4.7.2. Comparison with State-of-the-art Proposals 83
4.7.3. Discussion 84
4.7.4. Qualitative Results 84
4.8. Chapter summary 84

5. V2D applications **89**
5.1. Autonomous Driving Simulators 89
5.2. CARLA Simulator 90
5.2.1. Simulator Hardware 92
5.3. Communication Interface through ROS 92
5.3.1. Driver Take-Over Request ROS Interface 93
5.3.2. Driver's Attention-based Explainable Decision-Making ROS Interface 93
5.4. Scenarios 94
5.4.1. Driver Take-Over Request Scenario 94
5.4.2. Driver's Attention-based Explainable Decision-Making Scenario . . 95
5.5. Driver Take-Over Request Application 96
5.5.1. Experimental Framework 100
5.5.1.1. Driver's Gaze Focalization (Subsystem 1) 101
5.5.1.2. Simulator (Subsystem 2) 102
5.5.1.3. Take-Over Request Trigger 102
5.5.2. Experiment Setup 104

5.5.3. Experiment Results . . . 105
5.5.3.1. Gaze focalization analysis . . . 105
5.5.3.1.1. Gaze Focalization over the Objects of the Scene . 105
5.5.3.1.2. Gaze Focalization over the Different Screens . . . 107
5.5.3.2. Reaction Times Analysis . . . 109
5.5.3.3. Situation Awareness . . . 110
5.6. Driver's Attention-Based Explainable Decision-Making Application . . . 110
5.6.1. Experiment Framework . . . 111
5.6.2. Experiment Results . . . 112
5.7. Chapter Summary . . . 113

6. Conclusions and future works **119**
6.1. Conclusions . . . 119
6.2. Future Works . . . 120

Bibliography **125**

List of Figures

1.1. Word map of the thesis 2

1.2. Eye-trackers available in the market. 4

1.3. Driver Situation Awareness. Source: [34] 5

1.4. Driver Attention Estimation. Source: [55] 7

1.5. Driver Attention Estimation. Source: [71] 9

2.1. Proposed framework for driver attention estimation based on a monocular camera. 16

2.2. Full framework set-up. A user looks at a 24 inches screen with the camera positioned in front of him. DADA2000 videos are displayed on the screen to evaluate his attention. The attention map represented as a heat map is generated in real-time. 17

2.3. Nonlinear calibration script flow. 26

2.4. Driver Situation Awareness model. It shows the gaze focalization and the detector options to perform the analysis. 27

2.5. ROIs vs semantic segmentation detector as supervisor for the DSA analysis. 27

2.6. Relationships between objects and its attention level. 28

2.7. Calibration results using NARMAX calibration method. Eight points are displayed on the screen (Red circle). The proposed system creates 8 Gaussian around the testing points. 32

2.8. DADA2000 frame. Crash object circle for 60 and 300 pixels (blue). Error linear calibration area (red). Error NARMAX calibration area (green). . . 33

2.9. Temporal partition of the video. Dividing the accident part into three pieces of the same length to evaluate the results by sections. 33

2.10. Obtained Results on the testing using NARMAX calibration. 35

2.11. Attention map with temporal aggregation for the frame t, t-1 s, t-2 s and t-3 s using NARMAX calibration method. 36

3.1. ARAGAN framework. Generator, Discriminator and the flow of the generated information by these two nets are represented. 44
3.2. Convolutional Block Attention Module (CBAM). 47
3.3. Self-Attention module 49
3.4. Multi-Head Self-Attention module. 51
3.5. First approach. U-Net [135] architecture for driver attention estimation. . . 51
3.6. Residual convolutional modules used as an image feature extractor. 52
3.7. Second approach. Encoder based on Convolutional Block Attention Modules (CBAM) for driver attention estimation. 53
3.8. Third approach. Residual modules with Self-Attention mechanism for driver attention estimation. 53
3.9. Fourth and final approach. Residual modules with Multi-Head Self-Attention modules for driver attention estimation. 54
3.10. Discriminator architecture 55
3.11. Qualitative results for the model proposed using the combination of Residual modules + Multi-Head Self-Attention. 56

4.1. Driver's attention-based explainable decision-making framework. 60
4.2. Self-Attention procedure. 68
4.3. Cross-Attention procedure. 69
4.4. Multi-Head Attention mechanism 70
4.5. Transformer Encoder procedure. 70
4.6. Dataset analysis. 71
4.7. Driver attention-based explainable decision-making. 73
4.8. Driver's attention fusion based on ARAGAN. 74
4.9. Spatio-Temporal Transformer Encoder. 75
4.10. Example of positional coding for 49 positions and 100 channels. It is for display only because the presented model has 2560 channels. 76
4.11. Transformer Encoder 76
4.12. Classification heads. 78
4.13. Qualitative results for the driver's attention explainable decision-making in BDD-OIA test set. 87
4.13. Qualitative results for the driver's attention explainable decision-making in BDD-OIA test set. 88

5.1. CARLA simulator communication between the world and clients. 91
5.2. Point of view of the driver during the experiment. 92
5.3. Driver Take-Over Request ROS interface. 93
5.4. Driver's attention-based explainable decision-making ROS interface. 94
5.5. Developed scenarios for the proposed experiment. 95
5.6. Town01 scenario for the driver's attention-based explainable decision-making 96
5.7. Automation Levels, with the level of attention required by the driver. . . . 97
5.8. Experiment scenario take-over process timeline. 100
5.9. Proposed framework to analyse Take-Over Request (TOR). Green boxes determine the first subsystem/computer and blue boxes the second one. . . 101
5.10. Circular uncertainty gaze area with a radius of 30 pixels on the image following a conical projection represented with an orange cone. 103
5.11. CTRV model prediction and IOU between both vehicles to determine TTC. 104
5.12. Gaze focalization analysis with semantic segmentation. 108
5.13. Proposed framework to analyse the driver's attention-based explainable decision-making in CARLA. 111
5.14. Experiment conducted in CARLA to evaluate the driver's attention-based explainable decision-making. 115
5.14. Experiment conducted in CARLA to evaluate the driver's attention-based explainable decision-making. 116

List of Tables

2.1. Testing accuracy vs camera parameters on a 1920x1080 screen using NARMAX calibration method. The test is done by one person looking at eight points on a screen. The test was done 5 times. 30

2.2. Testing accuracy vs camera position on a 1920x1080 screen. The test is done by one person looking at eight points on a screen. The test was done 5 times. 30

2.3. Testing accuracy vs glasses on a 1920x1080 screen. The test is done by one person looking at eight points on a screen. The test was done 5 times using NARMAX calibration 31

2.4. Testing accuracy on a 1920x1080 screen. The test is done by 10 users looking at eight points on a screen using the NARMAX calibration method. 32

2.5. Success rate: percentage of the number of clips that frame ratio for the entire clip is more than half of the frames of the complete accident scene. . 34

2.6. Frame ratio: percentage of frames in each clip section where the attention point is inside the correct detection area. It is measured frame by frame because the crash object changes with it. 34

3.1. Attribute comparison with existing driver attention datasets.. 40

3.2. Ablation study performed using adversarial learning with different generators configuration. All models have been trained in BDD-A and tested in BDD-A and DADA2000. 57

3.3. Comparison with other state of the art models trained in BDD-A and testing in BDD-A and DADA2000 57

4.1. Driving datasets that can be used to develop explanation methods for AVs and the stakeholders that would potentially benefit from such explanations. 61

4.2. Action and explanation categories in the BDD-OIA dataset. Changing lanes to left/right is merged with turn left/right to avoid distribution imbalance. For the turn left/right rows, the upper sub-row presents statistics of changing lanes to the left/right and the lower sub-row those of turning left/right. For these actions, explanations address why the action is not possible. 62

4.3. Traffic anomaly categories in the DoTA dataset 63

4.4. Causality in Traffic Accident (CTA) cause and effects categories. 64

4.5. Ablation study of modules carried out to build the driver's attention-based explainable decision-making. Every model has been trained in the same data, and the results exposed are for the testing set of BDD-OIA. 83

4.6. Action and explanation prediction performance using the driver's attention-based explainable decision-making. It will be compared with other proposals in the literature. Results in blue colour represent the best of the evaluated models. **Bold** results represent the best over all the proposals. 84

4.7. Qualitative results explained for Figure 4.13. Correct predictions Incorrect predictions 85

5.1. Autonomous driving simulators 90

5.2. Results summary agenda 105

5.3. Gaze focalization analysis with semantic segmentation. 107

5.4. Gaze focalization percentage over the three screens during the transition after the Take-Over Request. 107

5.5. Reaction time for the different experiments performed. 109

5.6. Situation awareness ratio for the different experiments performed. 110

5.7. Comparison between the predicted decisions and explanations in some labelled sequences. 113

5.8. Qualitative and quantitative results explained for Figure 5.14. Green options denote correct predictions and red options denote incorrect predictions. 114

Chapter 1

Introduction

"Mi nombre es Máximo Décimo Meridio, comandante de los ejércitos del norte, general de las Legiones Fénix, fiel servidor del verdadero emperador Marco Aurelio. Padre de un hijo asesinado, esposo de una esposa asesinada y juro que me vengaré, en esta vida o en la otra."

Máximo; Gladiator

1.1. Aim

The role of driver attention systems for a smart vehicle-to-driver (V2D) interaction in an intelligent vehicle has not been explored in depth in the literature. This dissertation aims to understand driver attention and the benefits of its use to fulfil the self-driving task. It tackles a bidirectional path over driver attention that goes from calculating driver status to building decision-making imitating driver behaviour. This work is under the sight of the driver to understand their decision during driving and transfer this knowledge to the vehicle.

We base our methods used in this thesis on learning algorithms. Specifically, this dissertation presents the author's evolution from machine learning algorithms to deep learning ones. The former requires a deep understanding of the task by the developer, and the second one to understand some features from data like a human does without developer intervention.

With this proposal, this thesis aims to build a smart V2D interaction for safe driving. Figure 1.1 depicts the goal of this work, with the main topic in the centre (Driver attention) and the different modules developed around it. The first two modules are in charge of driver evaluation. They are a driver monitoring system and driver situation awareness. These two modules have a V2D application consisting of a take-over control to evaluate the driver with these two methods. Furthermore, there are two modules in the self-driving domain. They estimate driver attention and explain decision-making.

They have a V2D application to explain the vehicle's action predictions to a driver. We will further explain these modules in this chapter to introduce the questions that this thesis aims to answer.

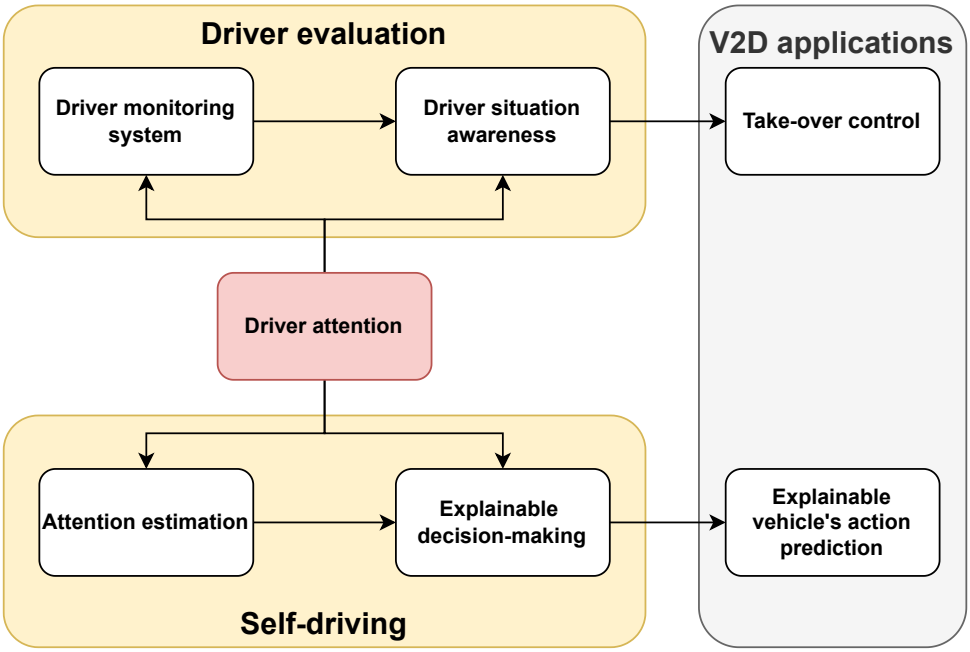


Figure 1.1: Word map of the thesis

1.2. Background

The study of the attention topic to increase road safety for intelligent vehicles follows the lines presented in [1].

Driving demands different activities that have to work together to fulfil the task. In addition to keeping the vehicle on the road, drivers pay attention to other agents and foresee potential hazards. Drivers rely primarily on vision to make decisions [2]. Understanding how drivers attend to the scene and ensuring their availability to drive is crucial for increasing road safety, as authors in [3], [4] proved that distractions increase the probability of an accident.

The automotive industry has made crucial advances in deploying systems for assistive and automated driving. Within these advances are the Advances Driver Assistance Systems (ADAS). Recent studies show that more than 30 % of the vehicles sold in the USA in 2016 had passive ADAS, such as rearview cameras, parking proximity sensors, and blind-spot detection [5]. The category of active sensors includes other ADAS, such as lane displacement detection, emergency braking, and adaptive cruise control. These systems have become standard in more than 200 vehicles manufactured in the last five years [6]. By 2030, 54% of all cars in Europe will have ADAS. These systems can reduce

up to a third of accidents occasioned by vehicles weighing less than 3.5 tonnes on highways [7].

Usually, ADAS act independently of the driver's state. Driver Monitoring Systems (DMS) offers an additional approach to safety by estimating drivers' inattention. Currently, most commercial DMS rely on vehicle measures such as steering or lateral control to assess drivers' state [7]. However, modern DMS are based on in-vehicle cameras to watch the drivers, analyse where they are looking at, and warn them in case of inattentions or critical objects/events on the road that request a take-over manoeuvre from autonomous driving.

Vision-based DMS are really important for automated vehicles, especially from SAE Levels 2-4 [8]. Authors in [9], [10] proved that, when drivers are supervising the driving, they are more tipped to distractions. Switching the control to manual should be preceded by driver assessment to evaluate if they are distracted or fatigued [11], [12]. Therefore, DMS take on a fundamental role in road safety, so much so that Euro NCAP 2025 roadmap to zero road fatalities by 2050 [13] includes it.

In addition to assisting the driver, recent studies have implemented attention mechanisms for perception tasks in self-driving vehicles [14]. They use these systems to improve the perceptual context and the ability to generate an explainable decision-making following human reasoning.

But, what is attention, and why is it needed in this context? Drivers do not take into account the complete scene during the driving. Instead, they pay attention to the most relevant regions/agents. It has a biological answer, as human attention is highest in the centre of the visual field (fovea) and lower in the periphery (parafovea) due to the receptors distribution in the retina [15], [16]. Head and eye movements complement this mechanism to focus on other areas of the scene [17].

The following sections introduce driver attention for the four main modules identified in this work. After that, we introduce V2D applications in a simulation environment to test the proposed methods in safe environments.

1.2.1. Driver Attention Estimation for Driving Monitoring

To estimate driver attention, commonly, the driver gaze is employed. Two attentional control mechanisms classify eye movements: bottom-up and top-down [18]. The saliency of the objects conducts the bottom-up mechanism. Items are attractive to the gaze [19], [20], for example, colour, brightness, etc. The task drives the top-down mechanism [21], for example, the front vehicle during an approaching manoeuvre.

Eye-trackers get the driver's gaze (Figure 1.2). They are used to obtain the foveal vision. Inside this technology, there are different options in the market. Tower-mounted eye-trackers (Figure 1.2a) are the most accurate ones and get more samples per second



(a) Tower-mounted eye-tracker (b) Desktop mounted eye-tracker (c) Head-mounted eye-tracker (d) Glass mounted eye-tracker

Figure 1.2: Eye-trackers available in the market.

[22], but they limit head movements. Desktop-mounted (Figure 1.2b) and head-mounted (Figure 1.2c), allow head movements but have lower precision and are limited to in-lab applications [23], [24]. In-vehicle applications normally use glass-mounted eye-trackers (Figure 1.2d). All the above eye-trackers are costly, intrusive and susceptible to data loss due to calibration problems. Using a camera as an eye-tracker provides an alternative sufficiently accurate system but with lower cost, almost no maintenance and without intrusiveness.

After tracking the eyes, a further step is needed to obtain the driver's attention. There are two approaches to solve this task, first, matching the driver gaze with respect to some coarse Areas Of Interest (AOIs) inside the car, using a classification mechanism. Second, calculating the driver's 3D gaze direction and its intersection with the 3D model of the vehicle interior, using a geometrical method [25], [26]

The most common approach in the literature to obtain the driver gaze is using eye-trackers [27], [28]. On the other hand, cameras get similar features but with lower accuracy. Their price, lack of intrusiveness and no need for maintenance make cameras the best suitable alternative for DMS in real applications.

DMS should meet some requisites such as real-time operating and a short computing chain to avoid adding errors. According to [25], to obtain a good driver's attention model for a DMS, it must apply the following steps:

1. Detect and track the face.
2. Detect eye landmarks (iris and pupil).
3. Eye pose estimation (position and rotation).
4. Head pose estimation (position and rotation).
5. Gaze pose estimation (fusing eye pose and head pose).
6. Projection of the 3D gaze into the 3D vehicle interior.

This work follows this approach and proposes a camera-based DMS that uses a NARMAX calibration to project the 3D gaze over a simulator to evaluate the driver's attention in challenging situations, as will be explained in Chapter 2.

DMS can also obtain driver attention maps for other tasks. There are two trends in how to record datasets for obtaining attention maps in a learning process: in-vehicle or in-lab. In-vehicle datasets record the driver attention maps while they are driving [29]. It offers the most ecologically valid conditions. Moreover, in-lab datasets collect the driver attention from multiple users who watch the same scene [30]. In-lab datasets get attention maps from multiple subjects over the same frame to replicate the parafovea vision using the addition of the fovea vision for different users [31], [32]. The choice of on-road versus in-lab conditions is a trade-off between realism and replicability.

This thesis uses driver attention datasets recorded in-lab since they capture more generalizable attention maps obtained from multiple users. We will explain this decision later in Chapter 3.

1.2.2. Driver Awareness Estimation

Driver attention helps to know what the drivers are aware of analysing whether the areas where they fix their gaze correspond to objects of interest in the scene in a driving manoeuvre. It helps to understand the drivers' behaviour and verify if they are suitable for driving, evaluating the scene and their attention. It is especially important in take-over manoeuvres with automated vehicles, where the driver can do Non-Driving Related Tasks (NDRTs).

Authors in [33] proposed a classification for NDRTs: 1) visual - requires using the gaze, such as checking the air conditioner; 2) cognitive - requires thinking, such as talking to the copilot; 3) manual - requires using the hands, such as using the mobile phone. Figure 1.3 shows a user performing a visual, cognitive and manual NDRT and the objects to be aware of in case of necessity.



Figure 1.3: Driver Situation Awareness. Source: [34]

We stated above that two different methods assess the driver's attention: classification and geometric. Attention systems based on classification do not consider the changing

environment and what the driver is aware of. For this reason, the current task uses the geometric method. Most geometric approaches follow the same procedure:

1. Estimate the driver's 3D gaze.
2. Convert it to the vehicle camera frame.
3. Detect surrounding objects of interest in the scene using the vehicle's camera in a supervised way.
4. Associate the driver's gaze fixations with the detected objects of interest.

With this procedure, [33] gets the attended agents and calculates driver awareness. Matching the drivers' gaze with the objects to obtain their attention is done with different strategies in the literature. Some solutions check if the gaze projection falls inside the object bounding box [35]–[37]. However, estimating the gaze as a point is inaccurate. For this reason, [38], [39] proposed to treat it as a cone to measure driver attention. Authors of [40], conclude that the gaze has an error of 10 pixels due to the eye-tracker error. Using a projection cone instead of a point partly solves the problem [41].

The evaluation of driver awareness is not trivial and is typically carried out with a qualitative assessment based on illustrative scenarios [39], [41]–[44]. Other proposals tested humans in no realistic driving conditions: pedestrian crossing [45], driving around a university campus [35] or in driving simulators [46], [47]. However, qualitative results give little idea of how effective, robust and usable a model might be.

Our approach uses a projection cone from our gaze focalization system based on a camera to evaluate driver awareness. We will employ a quantitative metric used in aviation [48] while the driver is performing NDRTs. In addition, we will use realistic driving scenarios in simulation in the validation process. We will show this evaluation in Chapter 5.5.

1.2.3. Modelling Visual Attention in the Traffic Scene

Recently, modelling driver attention and getting attention maps in traffic scenes has become a hot topic. These algorithms do not predict the driver's gaze. They learn to attend to regions for safe driving. Models used in this task rely on human data to estimate where safe drivers should look at in a specific traffic manoeuvre. The idea is that algorithms for autonomous driving can use mechanisms inspired by human attention to focus on the main objects of the scene to make decisions and make them more explainables. Driver attention is a fusion of two mechanisms: bottom-up (colour or intensity) and top-down (goals or intention) [49], [50]. This fusion enables drivers to focus only on the necessary objects to get sufficient information to make decisions [51], [52]. Modelling this behaviour allows the vehicle to optimise the visual scene

understanding [53], saving computational efforts when it is driving in a complex scenario and improving its interpretability [54].



(a) Input raw image (b) Model prediction

Figure 1.4: Driver Attention Estimation. Source: [55]

There are several approaches to predict driver attention from a camera. Some use optical flow that helps to identify the motion direction and magnitude [29], [56]–[58]. Another trend is to use 3D convolutional layers [59]–[62], which are heavy architectures. Other approaches take advantage of recurrent architectures to encode single frames [55], [63] or temporal ones [64]. Figure 1.4 shows the procedure of these models, where the input is an RGB image, and the output is the predicted attention map, drawn as a heat map.

The evaluation of these models follows the metrics of saliency models [65]. They evaluate how similar both distributions are (predicted and human-recorded) [66]. These models can be part of a V2D interface to assist the drivers with the areas where they should pay more attention while driving.

In this field of research, there are two ways of collecting data, as mentioned before, in-vehicle and in-lab. DR(eye)VE project dataset [29] was collected in-vehicle, and BDD-A [55] and DADA2000 [64] datasets were collected in-lab.

This thesis proposes an efficient driver attention estimation model imitating human behaviour to understand regions of interest in an image for driving. It is based on conditional generative adversarial learning and Multi-Head Self-Attention to obtain a precise attention map. It is trained in an in-lab driver attention dataset (BDD-A) obtained from humans and evaluated in this same dataset and in DADA2000, to test its domain adaptation capability. We will present this work in Chapter 3.

1.2.4. Attention for Self-Driving Vehicles

Self-driving cars are the next milestone to fulfil in the automotive industry. Current self-driving systems can broadly be modular and end-to-end. The former use dedicated modules for the different tasks involved in the navigation of an autonomous vehicle. The latter are unified systems that transform input from sensors directly to control commands

as black boxes. Attention mechanics are especially interesting in end-to-end systems. The role of attention is to identify the objects or areas in the scene that are more relevant in the driving, improving in this way the end-to-end models' performance and their explicability.

Different approaches on how to use attentional mechanisms are in the literature. Spatial attention is the most used in self-driving models because it retains the spatial distribution of features and applies attention weights to track back to the locations in the environment. Weights visualized as a heatmap can interpret the scene for the self-driving model as relevant areas. Spatial attention is usually applied to the intermediate and final layers of the feature extraction step. In [67], authors used multiple attention modules after the intermediate and last layers of the CNN to refine features. In [68], they used an attention module after the convolutional extractor as well as conditional spatial attention weights from the previous steps. In [69], authors implemented spatial attention trained as part of the network. The model extracts features using a pre-trained CNN, groups them and passes them through pooling layers. They implemented an attention block that multiplies its output with the output of the pooling layer.

In the last years, Transformers [70] have shown their effectiveness for many computer vision tasks. Transformer process sequential data using several identical blocks composed of Multi-Head Attention, feed-forward neural network, residual connection and layer normalization. This architecture supposes an alternative to use recurrences such as Long Short Term Memory (LSTM) or Gated Recurrent Unit (GRU) scheme. Stacking different attention blocks within a Multi-Head layer allows learning to focus on distinct input parts. The flexibility of the Multi-Head Attention to adapt to different types of input and the interpretability of their learned attention scores make them a valuable tool for the self-driving domain.

There are recent examples where driver attention has improved the efficiency and interpretability of self-driving architectures. In [72], authors trained a driver attention model that provides visual context to infer the vehicle speed. In [73], a pre-trained driver attention estimation model improves the accuracy of the vehicle control commands obtained during imitation learning. The driver's attention controls the amount and spread of dropout during the training. Authors of [74] proposed an explanation model for the vehicle controller that predicts a textual explanation for the forecasted decision and spatial attention to highlight the most relevant image regions. Kim et al. in [75] predict the control commands and spatial attention maps after inputting the model with textual advices. In [71], authors simultaneously generate the vehicle commands as natural language and textual and visual explanations, as shown in Figure 1.5.

This thesis proposes to take advantage of our driver attention model inspired by Transformers to obtain an explainable decision-making model trained on BDD-OIA [76],

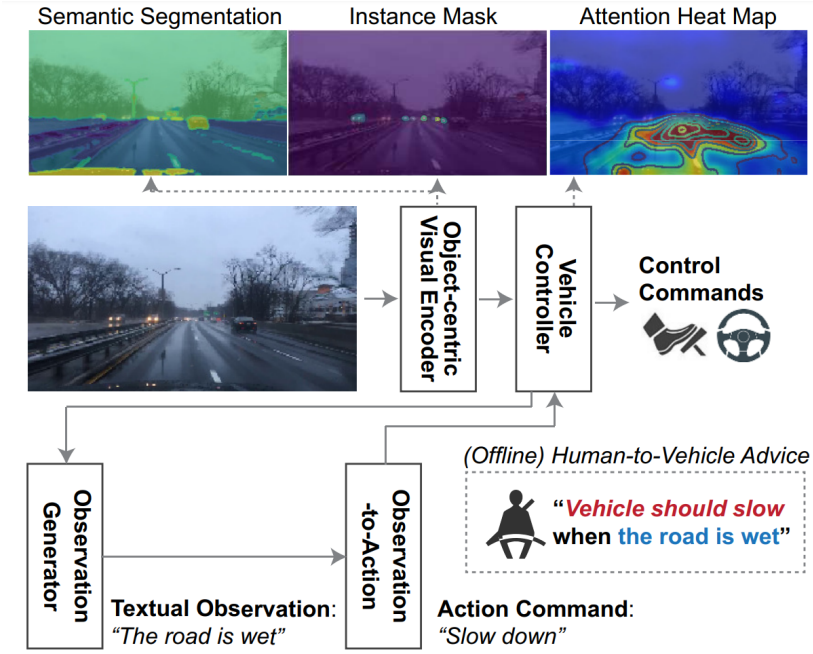


Figure 1.5: Driver Attention Estimation. Source: [71]

which takes as input a simple sequence of images from the vehicle's front camera and provides the action that the vehicle is performing, as well as an explanation for this action. We will expose this work in Chapter 4.

1.2.5. V2D Applications

The use of simulation is crucial in evaluating applications of intelligent vehicles, especially when a V2D interaction is required. Testing these systems in real-world scenarios is expensive, time-consuming, and potentially dangerous. Simulation, on the contrary, provides a controlled environment to test and evaluate these applications without personal and material losses. With simulation, developers can create challenging scenarios that are very difficult to obtain in the real world.

Moreover, simulation allows developers to test the performance of their applications under a wide range of conditions and scenarios, which is not always possible with physical testing. It helps to identify potential problems and allows for refinement and optimization of the systems before deployment in real-world environments.

Another benefit of simulation is that it enables developers to test various types of intelligent vehicle applications, including autonomous driving, vehicle-to-vehicle communication, and Advanced Driver Assistance Systems (ADAS), among others. With simulation, developers can assess the performance of these applications in different scenarios, including emergency situations, and evaluate their effectiveness in preventing accidents and improving safety.

In this work, we will develop and test two different applications based on our driver monitoring and decision-making models in simulated environments. We will explain them in Chapter 5.

1.3. Motivation

This dissertation aims to explore the research field of driver attention for smart V2D interaction, from its use as an ADAS to evaluate the driver status for predicting the executive layer of an autonomous vehicle. We allocate this work within the intelligent vehicles topic, where technology helps to achieve safe driving. Moreover, this dissertation explores the machine learning paradigm, especially the deep learning one. We implement two architectures based on the latest techniques in this field. Additionally, we propose another model based on machine learning.

The principal motivation of the thesis is relative to safe driving, studying driver attention. The first step is to build a DMS based on a camera to obtain a gaze focalization system. It is cheap, non-intrusive and can be easily placed in every vehicle. This system can help to assess driver attention and driver awareness during driving in order to build safe driving.

The next step goes further in the SAE Levels, helping an autonomous vehicle to focus on the most relevant zones in a scene following human behaviours. For this purpose, a trained model predicts attention maps based on recorded human-driving attention maps.

Building an explainable end-to-end autonomous vehicle executive layer using only the information from the front camera placed on the vehicle brings with it a challenge to be solved. For this purpose, and continuing with the proposed line, we leverage our driver attention estimation to make this task. Moreover, the decision has to be explainable with a dual explanation model: visual and "textual".

The final motivation of this dissertation is to assess our V2D proposals in applications in a simulation environment. Simulation provides a safe environment to test our algorithms in challenging situations, validating in this way our DMS and explainable decision-making models in different scenarios that we do not use in their training.

1.4. Contributions of Dissertation

We summarise the contributions of this dissertation in the following points:

- **Contribution 1:** A driver focalization model to asses the driver's attention during critical situations. It is based on a camera and can work on a vehicle. We evaluate the proposal in the DADA2000 dataset obtaining comparable results to those obtained

by the dataset's authors but using a simple camera instead of an expensive desktop-mounted eye-tracker (Chapter 2).

- **Contribution 2:** A novel real-time quantitative Driver Situation Awareness (DSA) metric based on gaze focalization, projection cone and semantic segmentation of the traffic scene (Chapter 2.3.5).
- **Contribution 3:** A real-time driver attention estimation model based on deep learning, which employs conditional Generative Adversarial Networks (cGANs) and Multi-Head Self-Attention, providing an efficient and accurate model. We validate the model in BDD-A and DADA2000 datasets obtaining better results than other state-of-the-art proposals and a good domain adaptation capability (Chapter 3).
- **Contribution 4:** An explainable end-to-end decision-making model leveraging our previous attention model. It is a two-stage model that uses driver attention to comprehend the visual scene, achieving the decision of the self-driving vehicle and its explanation. Validation results obtained in the BDD-OIA dataset give the best performance in explanation and comparable results in action detections that some other competitive strategies, as using a Fast R-CNN with an object selector [76] (Chapter 4).
- **Contribution 6:** Application of the gaze focalization system and DSA metric to evaluate the state of drivers during a simulated Take-Over Request (TOR) manoeuvre in CARLA (Chapter 5.5).
- **Contribution 7:** Application of the explainable decision-making model next to our driver attention contribution to vehicle's action prediction in CARLA simulator showing the domain adaptation of the proposal (Chapter 5.6).

1.5. Organization of Dissertation

We organize the remainder of this dissertation as follows: Related work is discussed separately for each of the three main chapters (Chapter 2, 3 and 4).

- **Chapter 2:** Description of the gaze focalization system based on a camera to capture driver attention developed in this work.
- **Chapter 3:** Presentation of the driver attention estimation model based on cGAN using efficiency Multi-Head Self-Attention.
- **Chapter 4:** Overview of the driver's attention-based explainable decision-making model, which ends the baseline of this work based on driver attention, achieving a smart-to-vehicle interaction.

- **Chapter 5:** Presentation of two V2D applications, the first one to evaluate a take-over request based on our driving monitoring system and the second one to assess an explainable decision-making, both in a simulation environment designed in CARLA.
- **Chapter 6:** Brief chapter including the most relevant conclusions obtained through this dissertation and future lines of work that remain.

Chapter 2

Driver Evaluation System

¿Por qué caemos, Bruce?
Para aprender a levantarnos.

Thomas Wayne; Batman Begins

This dissertation chapter presents a driver attention focalization system and a driver situation awareness metric. The architecture was implemented and evaluated using a driver attention dataset with multiple users. This work was partially published in the following Sensors journal paper [77]: "Gaze focalization system for driving applications using Openface 2.0 toolkit with NARMAX algorithm in accidental scenarios". The research aims to show the possibilities of building a driver's attention monitoring system with a camera and use it in a smart interaction with the vehicle.

2.1. Introduction

The future autonomous vehicles will have shared control between humans and machines. In consequence, ascertaining that the driver is suitable for driving will be a core task for safe driving. One of the moments when the vehicle should evaluate driver attention is during a take-over manoeuvre. The vehicle will request driver assistance in complex scenarios that cannot handle. In these scenarios, where the vehicle will request human help, it is crucial to ensure that the driver is aware of their surroundings (e.g., pedestrians or other cars) just before the take-over request to perform a safety transition.

A way to measure driver attention is to evaluate the driver's visual focalization. These modules are very important for automated vehicles safety, that is why the European Commission introduced a new regulation whereby all vehicles must have an attention warning system in case of driver drowsiness or distraction [78] from 2024. Gaze tracking estimation is the most used approach to evaluate the driver's visual focalization [79].

This chapter introduces a gaze focalization system based on a camera to evaluate driver attention in challenging situations. We study its power and requirements to be installed in a vehicle and present some evaluations performed with it in two distinct ways: with an ablation study to find the best parameter to be applied and in an accidental dataset. Additionally, we present a novel driver awareness situation metric.

2.2. Related Works

There are different approaches in the literature regarding gaze tracking. They are mainly based on head-mounted eye-trackers [80]–[82], active desktop-mounted eye-trackers [31] and passive desktop-mounted cameras [25], [79], [83]–[85]. The first ones provide accurate gaze information at the expense of being intrusive and costly. The latter are low-cost, no maintenance needed and non-intrusive but less accurate.

In [86], the authors reviewed the different methods for eye-tracking and gaze tracking, which are two different research fields. As the main contribution of this chapter is to implement a gaze tracking system for driver-to-vehicle interaction, this section will be focused only on this research field. Gaze tracking systems based on cameras use two different approaches: model-based techniques and regression-based methods. The formers use a geometric model of the eye to calculate the gaze direction. They calculate the gaze focalization by the intersection of the gaze vector with the object [87]–[91]. The later ones, [92]–[96], map the image features to the gaze coordinates using neural networks, as in [97], or polynomials, as in [98]. Most of these works use IR lights to obtain a stable gaze, whose prolonged usage causes dryness and fatigue in the eyes [99].

This thesis proposes using a regression-based system, but employing the parameters extracted from an open source face tracker, namely OpenFace 2.0 [100], to obtain the gaze focalization over a screen in order to develop a visual attention system.

OpenFace 2.0 is a facial behaviour analysis toolkit. In [100], authors made a broad comparison with other state-of-the-art systems. They evaluate the system in different tasks such as landmark detection, head pose estimation, gaze estimation and action unit recognition. As the objective is to obtain a gaze focalization estimation, the tool will be evaluated only in this task.

Some others state-of-the-art gaze estimators are: WebGazer.js [101], EyeTab [102] and OKAO [103]. This last tool is not free to use, which does not follow our premise of Open Source. Regarding the former two systems, they do not include head position, which is used in our proposed algorithm to estimate the gaze based on the correlation between both variables (gaze and head pose). Moreover, OpenFace 2.0 provides a C++ code and a simplified model that allows real-time operation, which is a key factor in this proposal.

Usually, simulations with simple scenes are used to validate visual attention systems in the literature [104]. However, the simulation of driver attention in complex driving

scenarios is rather challenging and highly subjective [32]. These systems aim to automatically estimate the area where the driver is focusing their attention to. For this reason, the acquisition systems and computer vision techniques are one step ahead of traditional human factor approaches focused on driving behaviour understanding using manual tools. To help the development of automatic models, different datasets have collected drivers' attention in the state-of-the-art. The first public dataset based on driving attention maps was the project DR(eye)VE [29], which collected 555,000 frames with those attention maps generated by the actual driver. After that, Berkeley DeepDrive Attention (BDDA) dataset [31] was published. It contains 1,232 videos that produced a braking event, and they recorded the attention map later in-lab. This approach not only presented a critical situation where attention is significant but also showed a work where the attention maps were generated by more than one user. However, these scenes are not challenging enough to test the attention because these situations do not cause accidents. Therefore, our method has been evaluated on the recent DADA2000 dataset [32], which is the largest and most diverse driving attention dataset composed of accidental scenarios. It contains 2,000 video clips of fairly complex accidental scenarios in diverse weather and lighting conditions.

Our proposal explained here, uses a non-linear model based on NARMAX (Nonlinear AutoRegressive Moving Average model with eXogenous inputs) to achieve a cheap and non-intrusive gaze focalization system. This is our first step into the machine learning paradigm, which will be deepened in following chapters. It can get accurate results on par of the most sophisticated systems over complex accidental scenarios. With this objective, it has been validated with the challenging public dataset DADA2000 [32]. The authors collected real accidental scenarios annotated with the accident localization on the scene and the attention map generated by some users who watched the clip in their laboratory using an expensive and active desktop-mounted eye-tracker. A comparison with their gaze focalization system is proposed. The comparison has been done in similar conditions to the one reported by the authors. Up to our knowledge, this is the first time that DADA2000 has been used to evaluate a camera-based vehicle-mounted driver attention system.

2.3. System Architecture

The proposed system has four modules that work together. The combination of all of them allows us to obtain the gaze focalization over the scene. These nodules are: video acquisition, gaze estimation, gaze focalization model and driver attention. Figure 2.1 shows the architecture.

All of these modules communicate with each other using Robot Operating System (ROS) [105]. It is a framework that allows the standardisation, communication and

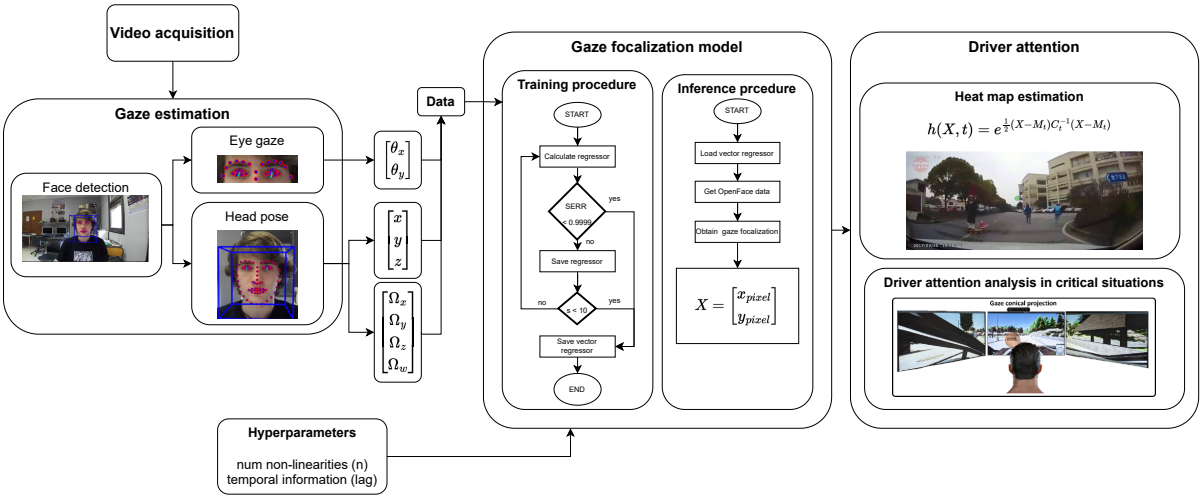


Figure 2.1: Proposed framework for driver attention estimation based on a monocular camera.

modularity facilities for all the modules. It offers the possibility to modularize the complete system and even to work on distributed systems.

2.3.1. Video Acquisition

The user image is obtained by a ZED camera from Stereolabs (Figure 2.2). It is a stereo camera with two lenses. However, for this application, it will be used as a monocular camera. Furthermore, this camera has been chosen due to its ROS compatibility and great specs in terms of frame rate and resolution. Sterolabs provides a wrapper that allows the publication of all the camera data using ROS topics. Furthermore, the camera can work at 100 fps at VGA, which is much higher than what could be obtained by a standard webcam. Nevertheless, this application is not limited to this kind of camera. It could work with a standard webcam, but it obtains worse results.

Evaluating all the camera possibilities for this application is an ablation study. The camera has been tested at different resolutions and frame rates. In addition, the relative position of the user in relation to the camera is an interesting issue to evaluate in terms of the possibilities offered by such systems in a vehicle. An ablation study will be carried out in the following sections.

Figure 2.2 shows the experimental set up to emulate the same environment as the authors of the DADA2000 dataset in order to have a fair comparison.

2.3.2. Gaze Estimation

The main goal of this chapter is to contribute to the scientific field with a gaze focalization system based on a camera. Our proposal is a non-intrusive and cheap system that does not generate rejection or fatigue to the driver. Furthermore, it can be mounted on a vehicle without disturbing the user. This premise makes than

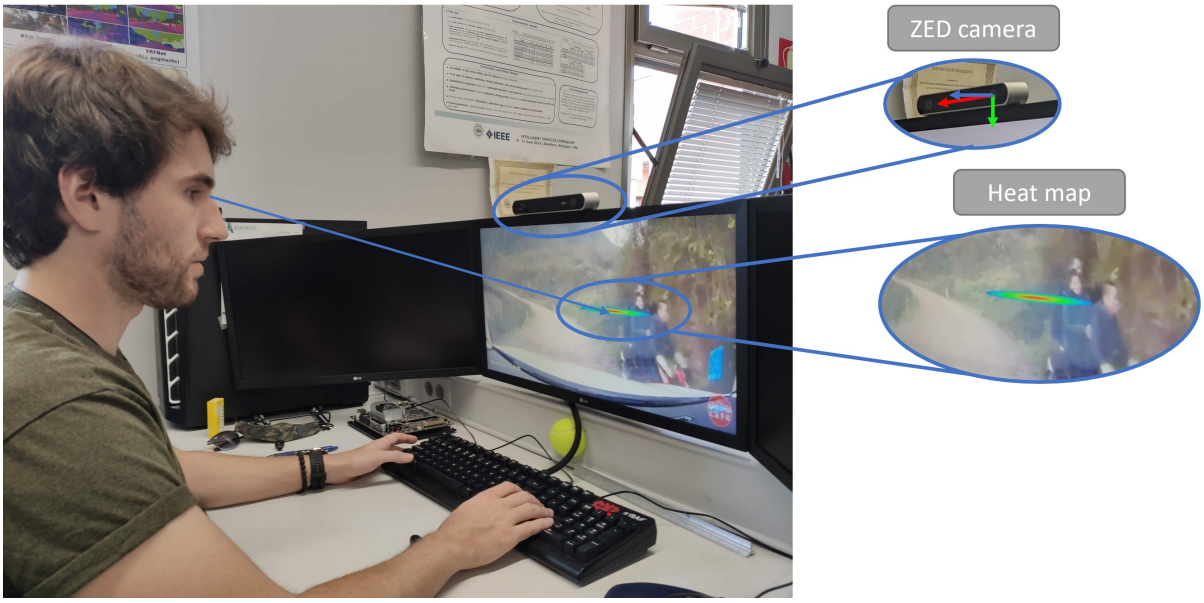


Figure 2.2: Full framework set-up. A user looks at a 24 inches screen with the camera positioned in front of him. DADA2000 videos are displayed on the screen to evaluate his attention. The attention map represented as a heat map is generated in real-time.

desktop-mounted eye-trackers and head-mounted eye-trackers are refused to be used in this application [25], [79], [83]–[85].

Face detection is one of the most challenging modules. OpenFace is one of the most popular open source facial analysis tools because of its robustness and performance. As this study is out of the scope of this thesis, we propose to use OpenFace 2.0 toolkit to provide face detection and facial landmarks. With some of these landmarks, it estimates the gaze direction, as shown in Figure 2.1.

The tool uses a Conditional Local Neural Field (CLNF) to predict facial landmarks. These facial landmarks include: eyelids, iris and pupils. Having the eyes detected and located in the space, it computes the gaze vector of each eye using the eye region. It returns three vectors, one for each eye and a third one for the fusion of the other two. This last vector next to some other facial landmarks will be used in this work. The OpenFace 2.0 authors tested the tool performance on the challenging MPIIGaze dataset [106]. They achieved a mean absolute error of 9.1 degrees per frame. It is a satisfactory result for this benchmark and good enough for the proposed application. The tool predicts an amount of 68 facial landmarks. The used ones will be explained in the following sections.

2.3.3. Gaze Focalization Model

Variation of the gaze angle is not linear, but it can be assumed to be linear when the gaze range is small. To implement a system as general as possible, we improve our previous gaze focalization model based on a linear method [107]. This new proposal is based on

a Nonlinear AutoRegressive Moving Average model with eXogenous inputs (NARMAX model).

The NARMAX model [108] includes a wide range of nonlinear systems:

- Past signal terms (AR).
- Past noise terms (MA).
- Delayed or not signal (X).

Relationships among these signals are nonlinear. The modelled signal is discrete and can be defined as:

$$\begin{aligned} y(k) = F[&y(k-1), y(k-2), \ldots, y(k-n_y), \\ &u(k-d), u(k-d-1), \ldots, u(k-d-n_u), \\ &e(k-1), e(k-2), \ldots, e(k-n_e)] + e(k) \end{aligned} \tag{2.1}$$

where $y(k)$, $u(k)$ and $e(k)$ are the system output, input, and noise sequence, respectively; n_y, n_u, and n_e are the maximum lags for the output, input and noise; F is an unknown nonlinear function and d is a time delay, typically set to $d = 1$.

The NARMAX method is based on five steps that answer the following questions:

1. Structure detection: ”What parts are in the model?”;
2. Parameter estimation: ”Which are the values of the model coefficients?”;
3. Model validation: ”Is the model correct and unencumbered?”;
4. Prediction: "What does the modelled signal look like in the future?”.

Some methods to answer these questions, are FROLS or MFROLS [109]. These algorithms try to find the simplest model that describes the system structure. FROLS will be tracked hereafter because is the method used in this work.

The concern with these approaches is that the developer has to suggest possible types of nonlinearity because the amount of nonlinearities is high for this type of application. These can be: Polynomial degree, Term degree, Logarithm, Neuronal network, and Superposition of the above methods.

For this reason, modelling a nonlinear system brings the possibility of unknowingly making a fatal error. If the system under test has a nonlinearity that was not assumed, the model will not fit the data. In NARMAX methods, the only limits are the assumptions made by the modeller. After this is exposed, some nonlinear modelling systems are presented.

2.3.3.1. OLS

This work proposes using the Orthogonal Least Squares (OLS) algorithm to establish the correspondence between the facial features based on the camera and the coordinates projected by the gaze on the screen. This approach is used on nonlinear systems because OLS will search through all possible candidate models to select the best one.

The main idea of the OLS estimator is to introduce an auxiliary model whose elements are orthogonal to the modelled signal, as in the Wiener Series [110]. Then, subsequent components of the orthogonal model in turn can be determined, and then the parameters of the searched model can be calculated based on them. An iterative repetition of the steps of this method allows not only finding unbiased model estimators but also showing the contribution of each of them to the final modelling result.

Considerations should begin with the assumption of the general model:

$$y(k) = \sum_{i=1}^{M} \theta_i p_i(k) + e(k) \tag{2.2}$$

where $y(k)$ is the modelled system response for $k = 1, 2, \ldots, M$; where M is the number of regressors; θ are the model parameters ($\theta \in R$) associated to the regressors $p_i(k)$ and $e(k)$ is the external noise or error at the moment $k = 1, 2, \ldots, M$.

Regressors $p_i(k)$ are defined as the combination of delayed signal values or delayed external signals. In the general case, the function of the delayed components of the model can take any nonlinear form. For the model, it is assumed that each regressor $p_i(k)$ is independent of the model parameters θ, therefore:

$$\frac{\partial p_i}{\partial \theta_j} = 0 \text{ for } i = 1, 2, \ldots, M \text{ and } j = 1, 2, \ldots, M \tag{2.3}$$

The goal of the estimator is to transform the model specified in Equation (2.2) into an auxiliary model whose elements are orthogonal to each other. This type of model has the form:

$$y(k) = \sum_{i=1}^{M} g_i(k) q_i(k) + e(k) \tag{2.4}$$

where $g_i(k)$ are the parameters of the orthogonal model and $q_i(k)$ ($i = 1, 2, \ldots M$) are the orthogonal components of the model. The orthogonality condition is presented as the following Equation:

$$\sum_{k=1}^{M} g_i(k) q_i(k) = \begin{cases} d_i = \sum_{k=1}^{M} q_i^2(k) \neq 0, & i = j \\ 0, & i \text{!=} j \end{cases} \tag{2.5}$$

The orthogonalization procedure for the model from Equation (2.2) can be summarized as:

$$\begin{cases} q_1(k) & = p_1(k) \\ q_2(k) & = p_2(k) - a_{1,2}p_1(k) \\ q_3(k) & = p_3(k) - a_{1,3}p_1(k) - a_{2,3}p_2(k) \\ & \vdots \\ q_m(k) & = p_m(k) - \sum_{r=1}^{m-1} a_{r,m}q_r(k),\ m = 2, 3, \ldots, M \end{cases} \tag{2.6}$$

Where parameter $a_{r,m}$ associates the components of the output model from Equation (2.3) with the orthogonal model according to the following form:

$$a_{r,m} = \frac{\sum_{k=1}^{N} p_m(k)q_r(k)}{\sum_{k=1}^{N} q_r^2(k)}, 1 \leq r \leq m-1 \tag{2.7}$$

On the base:

$$g_i = \frac{\sum_{k=1}^{N} y(k)q_i(k)}{\sum_{k=1}^{N} q_i^2(k)}, i = 1, 2, \ldots, M \tag{2.8}$$

Therefore:

$$\begin{cases} \theta_M & = g_M \\ \theta_{M-1} & = g_{M-1} - a_{M-1,M}\theta_M \\ \theta_{M-2} & = g_{M-2} - a_{M-2,M-1}\theta_{M-1} - a_{M-2,M}\theta_M \\ & \vdots \\ \theta_m & = g_m - \sum_{j=m+1}^{M} a_{m,j}\theta_j,\ m = M-1, M-2; \ldots, 1 \end{cases} \tag{2.9}$$

Based on the previously established regressors, a basic model is built to obtain the desired output.

2.3.3.2. ERR

The difficulty in carrying out OLS estimation is the criterion for selecting subsequent regressors for the orthogonal model. The Error Reduction Ratio (ERR) is employed to measure the significance of each model parameter. The model also has noise e. The energy (or variance) of the system is represented as:

$$\frac{1}{N}y^T y = \frac{1}{N}\sum_{i=1}^{M} g_i^2 q_i^T q_i + \frac{1}{N}e^T e \tag{2.10}$$

Where y is the modelled signal vector, g_i are the coefficients standing next to elements of the orthogonal model, q_i are the vector elements of the orthogonal model, e is the vector of noise samples, and N is the number of signal samples. It can be stated that:

$$1 = \frac{\sum_{i=1}^{M} g_i^2 q_i^2 q_i}{y^T y} + \frac{e^T e}{y^T y} = \sum_{i=1}^{M} ERR_i + ESR \tag{2.11}$$

where ERR_i is the Error Reduction Ratio and ESR is the Error to Signal Ratio. (ERR) is employed to measure the significance of each model parameter. It indicates the system improvement as a variable from 0 to 1. It can be used to measure the contribution of each model parameter. This ability is significant in order to obtain the best model without making it too complex. ERR allows the model to get only the best parameters to achieve the best model performance without a high training time and without complicating the model.

When the sum of the ERR subsequent values tends to one means that the modelling process can stop ($\sum_{i=1}^{M} ERR_i \to 1$). The comparison of individual ERR values for various elements of the model shows which of them are the most influenced ones by the signal of the designated model.

2.3.3.3. FROLS

The joint use of OLS estimator and ERR coefficient gives the FROLS (Forward Regression with Orthogonal Least Squares) nonlinear modelling algorithm, which is a forward regression algorithm using the orthogonal sum of the least squares. It is assumed that the selection of subsequent regressors for the OLS estimator is dependent on the highest ERR value for a given regressor. It allows to choose the model component that reduces the modelling error to the greatest extent and stops the modelling when ESR accomplishes a determinate value. In the following sections, the modelling steps will be presented according to the FROLS algorithm.

2.3.3.3.1. Step 1. Data Collection In order to perform the modelling and acquisition of the test signal waveforms, $y(k)$, (X_{pixel}, Y_{pixel}), as well as external input signals, $x(k)$ (OpenFace parameters) are required, which affect the system under study. The signal should be collected as discrete or transformed into such a form. In addition, the recorded waveform should have as little external noise as possible and not be subjected to filtering, which may disturb the identification process. According to the author of the method [108], if the noise present in the signal is zero-mean white noise, it does not affect the ERR value and the identification process.

2.3.3.3.2. Step 2. Defining the Modelling Framework Due to the fact that the number of possible nonlinearities is very high, an a priori assumption is required to define the search method. The nonlinearities requirements are formulated by answering the following questions:

- Output signal: Which is the maximum delay of the AR term (n_y)?
- Input signal: Which is the maximum delay of external signals (n_x)?
- Which are the predicted nonlinearities, and which is their maximum degree (l)?

After the developer has provided the answers to these questions, the following parts of the model are known: $y(k-1), y(k-2), \ldots, y(k-n_y), u_i(k-1), u_i(k-2), \ldots, u_i(k-n_{x_i})$. Where y is the signal under test, u_i is one of the external input signals, n_y is the maximum delay of the signal under test and n_{x_i} is the maximum delay of one of the external input signals. The nonlinearity that binds these components is also known, which allows the next step to be taken. When the nonlinearity is a polynomial, its maximum power will be determined by the maximum nonlinearity degree.

2.3.3.3.3. Step 3. Determination of the Regressor Vector Knowing the modelling framework defined in the previous step, a vector containing all possible regressors for a given signal must be determined. For example, for a system with one output (y) and one input (u) with $n_y = n_x = 2$, where nonlinearity is modelled as a polynomial up to the second degree, $l = 2$. The regressor vector will consist of a combination of signals: $y(k-1),\ y(k-2),\ u(k-1),\ u(k-2)$ according to the maximum polynomial degree. Including the constant component marked as c, regressor vector marked as D is defined as:

$$\begin{aligned} D = \{&c, y(k-1), y(k-2), u(k-1), u(k-2)\} \cup \\ \{&y^2(k-1), y(k-1)y(k-2), y(k-1)u(k-1), y(k-1)u(k-2)\} \cup \\ \{&y^2(k-2), y(k-2)u(k-1), y(k-2)u(k-2)\} \cup \\ \{&u^2(k-1), u(k-1)u(k-2)\} \cup \\ \{&u^2(k-2)\} \end{aligned} \tag{2.12}$$

2.3.3.3.4. Step 4. Choosing the First Element Once the regressor vector has been specified, the dataset of the created model is known. Choosing the first element requires determining the value of ERR for each element of the vector $D = p_1, p_2, \ldots, p_m$, where m is the number of potential regressors.

$$g_m = \frac{y^T p_m}{p_m^T p_m} \tag{2.13}$$

$$ERR[m] = \frac{g_m^2 p_m^T p_m}{\sigma} \tag{2.14}$$

$$l_1 = arg \max_{1 \le m \le M} ERR[m] \tag{2.15}$$

where l_1 corresponds to the highest regressor and it is accepted as the first element of the orthogonal model. The parameters g and ERR are saved.

$$q_1 = p_{l_1} \qquad g_1 = g_{l_1} \qquad err[1] = ERR[l_1] \qquad D_1 = D - p_{l_1} \tag{2.16}$$

After saving the found regressor and its associated values, the search is restarted to find the next member of the orthogonal model.

2.3.3.3.5. Step 5. Selecting the Next Elements of the Model The following step is analogously performed to the first step, with the difference that the orthogonal model already has a certain number of parameters, depending on the number of iterations previously performed. The calculation of the ERR values for new potential model members must be newly done with the current model shape. Therefore, the regressor index is searched (marked as l_s) about the highest ERR for the current model form, where s is the current step. The regressor vector is marked as D_{s-1} and does not contain the members selected in the previous steps. Then, for $m = 1, 2, \ldots, M - (s-1)$ and $m \neq l_1 \neq l_2 \neq \ldots \neq l_{s-1}$, can be expressed as:

$$q_m = p_m - \sum_{r=1}^{s-1} \frac{p_m^T q_r}{q_r^T q_r} q_r, p_m \in D_{s-1} \tag{2.17}$$

$$g_m = \frac{y^T q_m}{q_m^T q_m} \tag{2.18}$$

$$ERR[m] = \frac{g_m^2 q_m^T q_m}{\sigma} \tag{2.19}$$

$$l_s = arg \max_{1 \leq m \leq M-(s+1)} ERR[m] \tag{2.20}$$

After finding l_s, which corresponds to the highest regressor ERR, the orthogonal model is extended by another member, q_s, along with the corresponding parameter g_s. The ERR value for the selected element is saved, and the selected regressor is deleted from the vector D. In addition, parameter values $a_{r,s}$ are determined according to the following equations:

$$q_s = q_{l_s} \qquad g_s = g_{l_s} \qquad err[s] = ERR[l_s] \qquad D_s = D_{s-1} - p_{l_s} \tag{2.21}$$

$$a_{r,s} = \frac{q_r^T p_{l_s}}{q_r^T q_r}, r = 1, 2, \ldots, s-1 \tag{2.22}$$

The model is iterated over this step until the ESR value is satisfactory:

$$ESR = 1 - \sum_{s=1}^{M_0} err[s] \tag{2.23}$$

where M_0 is the number of selected regressors. A threshold is necessary for ESR, which will decide the end of the modelling.

2.3.3.4. Modelling Process Analysis

An important decision when performing modelling with the NARMAX method is when to stop adding more regressors to the model. The right moment does not have a strict definition but is determined by a set of factors whose superposition is a premise to take into account. In the case of NARMAX, one of the basic parameters is the ERR sum for the model, called $SERR$:

$$SERR = \sum ERR \tag{2.24}$$

It is assumed that a well-fitted model has a $SERR$ higher than 0.8 [111]. we will use a more restrictive threshold in this work.

2.3.3.5. Determination of the Final Model

When M_0 regressors are selected, it is necessary to determine their parameters. Assuming a low value of ESR, the general and orthogonalized models are equal to:

$$y(k) = \sum_{i=1}^{M_0} \theta_i p_i(k) + e(k) = \sum_{i=1}^{M_0} g_i q_i(k) + e(k) \tag{2.25}$$

In Equation 2.25, the only unknowns are θ_i. Regressors g_i were selected during modelling. The orthogonal model elements $q_i(k)$, along with their parameters were appointed on an ongoing basis during the process. Therefore, converting from an orthogonal model to the initial model by performing an inverse process to the previous orthogonalization. Knowing the values of the elements $a_{r,s}$ for $1 \leq r < s \leq M_0$, the following matrix is specified:

$$A = \begin{pmatrix} 1 & a_{12} & a_{13} & \dots & a_{1M_0} \\ 0 & 1 & a_{21} & \dots & a_{2M_0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & a_{M_0-1,M_0} \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \tag{2.26}$$

Which can be included in the Equation:

$$A\theta = g \tag{2.27}$$

Where $g = [g_1, g_2, \dots, g_{M_0}]$ and $\theta = [\theta_1, \theta_2, \dots, \theta_{M_0}]$. The solution of this equation gives the model parameters, thus determining the full output model.

The output has a two coordinates vector form (X_{pixel}, Y_{pixel}). Therefore, two models should be calculated to obtain the system output as required.

2.3.3.6. Calibration Procedure

Three main parts compose the framework used in this calibration procedure, which define how the subsystems are communicated among them in order to achieve the best result. The first one, is in charge of getting data to calibrate. The second one, calculates the regressors for the model. Finally, the last one uses these regressors and sends the topic to the ROS environment.

In order to achieve this calibration method, a training method is necessary. At the beginning of any test, the user has to look at eight points around the screen. These points are displayed for 8 s, but in order to obtain a more accurate model, the data of the first two seconds are removed. During the first 2 seconds, the eyes are in transition between fixations. These points are displayed in red over a black image using OpenCV. They have a 30-pixel radio and are placed on the following positions in pixels: $[X_1, Y_1] = [200, 200]$, $[X_2, Y_2] = [700, 200]$, $[X_3, Y_3] = [1100, 200]$, $[X_4, Y_4] = [1700, 200]$, $[X_5, Y_5] = [200, 700]$, $[X_6, Y_6] = [700, 700]$, $[X_7, Y_7] = [1100, 700]$ and $[X_7, Y_8] = [1700, 700]$. After training, these points will be evaluated during a testing procedure.

Data is composed of the point where the user is looking at (X_{pixel}, Y_{pixel}) and the data provided by OpenFace for the gaze and the head for this position, which is formed by: Gaze angle (X (θ_x) and Y (θ_y)); Head position (X (x), Y (y), and Z (z)); and Head rotation (X (Ω_x), Y (Ω_y), Z (Ω_z)).

Data is referenced to the camera and saved on a *.txt* file with the following structure: Gaze angle X; Gaze angle Y; Head position X; Head position Y; Head position Z; Head rotation X; Head rotation Y; Head rotation Z; Head rotation W; X_{pixel}; and Y_{pixel}. These are the chosen parameters because gaze focalization systems take as input: eyeball orientation and head pose (head orientation and position) [87]. Instead of eyeball orientation, a gaze vector provided by OpenFace 2.0 is employed to obtain a more accurate gaze focalization system.

The created file with the recorded data for each test will be loaded on the next step of the model, where a training procedure is carried out to calculate the parameters (see Figure 2.3). The data feed the model and follow the steps described earlier to obtain the regressors and their parameters for the two models.

The NARMAX model is fitted with these parameters. The model will use the necessary parameters to fit the model to the output as best as possible.

After some experiments, the best model is obtained when SERR is higher than 0.9999 or when the model has ten regressors. More regressors complicate the model with a slight improvement in accuracy and overfit the model.

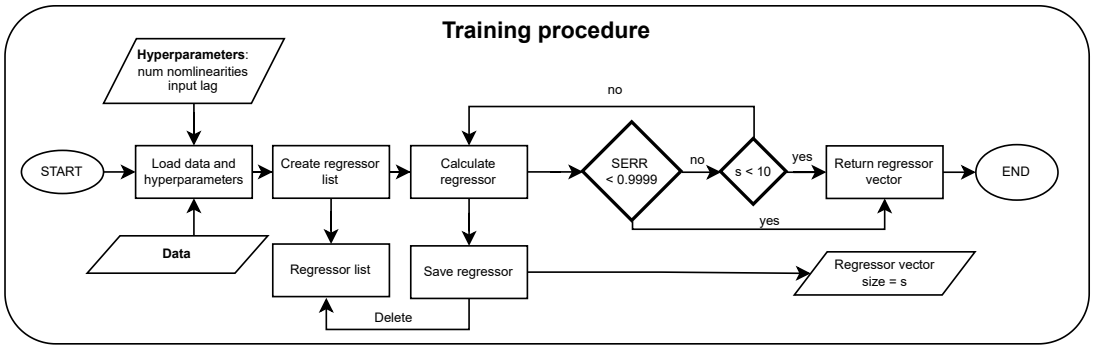


Figure 2.3: Nonlinear calibration script flow.

2.3.4. Heat Map Visualisation

The Heat map is one of the most common visualization approaches to represent a spatial distribution produced by a gaze focalization system on the screen, which represents the driver attention in our context. The spatial distribution over time is obtained using a temporal window $[t-w,t]$, where t is the current frame, and w is the width, which has been set to 1 s as done by [112]. In one second there are 30 samples to generate the spatial distribution that is represented as a heat map. The distribution is fitted on a Gaussian model. It represents the different focalized areas in the image using a colour gradient. $M_t = [m_x(t), m_y(t)]$ is the centre of the Gaussian, C_t is the covariance matrix and $h(X,t)$ is the intensity of the heat map for the pixel ($X = [x,y]$). For this work, ellipsoidal level curves following the Gaussian model are drawn applying Equation 2.28.

$$h(X,t) = e^{-\frac{1}{2}(X-M_t)^T C_t^{-1}(X-M_t)} \tag{2.28}$$

2.3.5. Driver Situation Awareness (DSA)

Using gaze focalization as an awareness metric is valuable because eyes off the road for a time results in a reduced attention level, which is highly decreased when the driver is performing some NDRTs. However, this is just a rough estimation of the driver's attention and cannot distinguish between the perception of relevant and irrelevant objects inside the driver's field of view [48]. Active driver attention is still an open challenge due to the high variability of the traffic scene data. To do that, it is necessary to match the driver's gaze fixations with the objects of interest in the traffic scene established by a supervisor, as can be depicted in Figure 2.4.

Because of this, a more accurate metric is required to measure the DSA to determine the driver's state, especially during the take-over request, as will be evaluated later in Chapter 5.5.

Several studies in the literature assess DSA [39], [113], but they do not work in real-time. In [48], authors propose an online method inspired by aviation using Region-of-

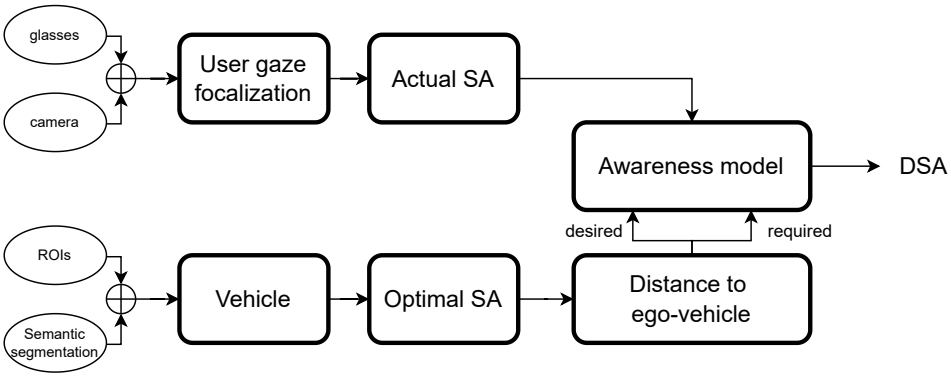


Figure 2.4: Driver Situation Awareness model. It shows the gaze focalization and the detector options to perform the analysis.

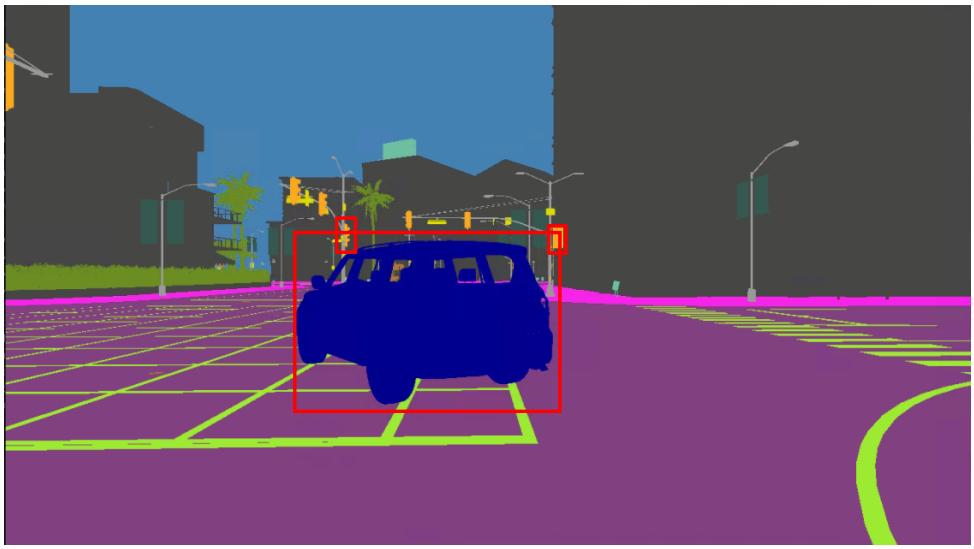


Figure 2.5: ROIs vs semantic segmentation detector as supervisor for the DSA analysis.

Interest (ROI) prediction and eye-tracking. The comparison of the actual awareness with the target awareness calculates the DSA. Driver's gaze focalization areas on the objects are the actual awareness. The objects, where a supervisor thinks the drivers should have focalized their gaze at, are the target awareness, which are based on the distance to the camera (Figure 2.4). However, this method has only been validated with pure observation, not in challenging take-over scenarios, as will be evaluated later. Moreover, ROI detections enclose more information as the object of interest (Figure 2.5), which can be misleading. To solve this problem we propose to use semantic segmentation instead of ROIs to determine the target awareness and cone projection for the actual awareness as well as validate this metric in an urban stressful transition scenario in real-time (Chapter 5.5). Semantic segmentation produces pixel-wise detections, which enclose only the object of interest, which is perfect for detecting whether the gaze has focused on the object.

Most driver's take-over behaviour metrics in the state-of-the-art do not consider the environment and what the driver is aware of on the surrounding elements. However, a better understanding of the driver's behaviour and context is fundamental for these

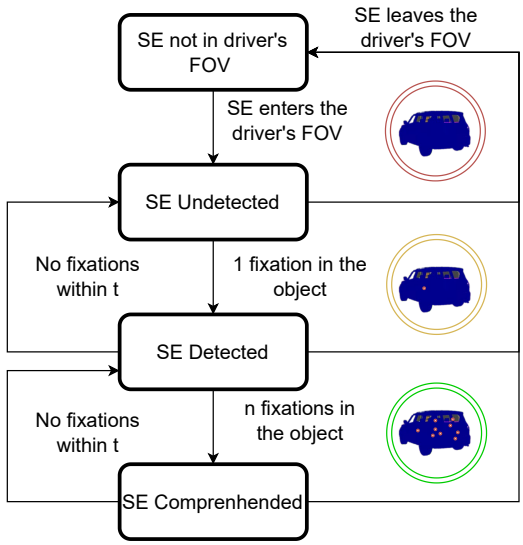
systems to verify if the drivers are aware of relevant objects before passing them the control. Methods used in aviation to test pilots' awareness inspired this use. Authors in [114] defined the SA score as the ratio of the actual SA and the optimal SA. Equation 2.29 shows that score, where SA_{actual} is the actual situation awareness and $SA_{optimal}$ is the target awareness.

$$SA_{ratio}(t) = \frac{SA_{actual}(t)}{SA_{optimal}(t)} \tag{2.29}$$

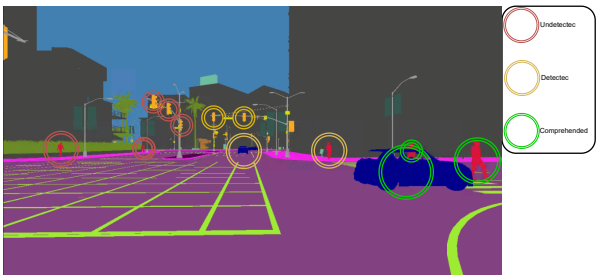
The optimal SA defines awareness as the situation where the driver perceives and comprehends all Situation Elements (SEs) as the weighted sum of required ($SE_{opt,r}$) elements and desired ($SE_{opt,d}$) elements. In our case, the SEs are obtained automatically and online through semantic segmentation, taking the relevant elements for driving (vehicles, pedestrians, cyclists, traffic lights, etc.) of the scene.

$$SA_{optimal} = 2\sum_{r=1}^{m} SE_{opt,r} + \sum_{d=1}^{n} SE_{opt,d} \tag{2.30}$$

Equation 2.30 shows the optimal SA. The difference between the required and desired elements is a function of their Euclidean distance to the ego-vehicle.



(a) Workflow of the proposed driver situation awareness model. It shows the object process along the framework.



(b) Visualization of the detections with circular ROIs (optimal SEs) over the semantic segmentation and their perception level. The undetected SE is visualized in red, the detected SE in yellow and the comprehended SE in green.

Figure 2.6: Relationships between objects and its attention level.

The actual SA is calculated following the Equation 2.31, where p_{drt} and p_{irt} are 0 for an undetected SE, 0.5 for a detected SE, and 1 for a comprehended SE to differentiate among these detection levels (Figure 2.6). A time window (t) has been implemented to measure the gaze over the different objects. Nevertheless, in contrast to the original work, semantic segmentation is used instead of ROIs over the detections because it gives more accuracy than a ROI due to the former technique using pixel-wise labelling.

$$SA_{actual} = 2\sum_{r=1}^{m} SE_{act,r} + \sum_{d=1}^{n} SE_{act,d} = 2\sum_{r=1}^{m} p_{irt} + \sum_{d=1}^{n} p_{drt} \tag{2.31}$$

This metric will be evaluated in Chapter 5.5 to perform a take-over request study with some users with the gaze focalization system presented here.

2.4. Experimental Results

In this section, the results obtained for our driver monitoring system are presented. The evaluation is divided into three subsections. Firstly, a precision experiment is done to test the accuracy of the system. Secondly, an ablation study is carried out. Finally, a comparison is conducted with other state-of-the-art systems in the challenging dataset DADA2000 as was carried out in a previous author's work [107].

2.4.1. Visual Focalization Model Evaluation

A precision test for each user has been built to evaluate the accuracy of the driver focalization system. After the calibration and training steps, the users have to focalize their gaze on eight points for eight seconds per point. The points are displayed in different locations on the screen. The first two seconds of each point are despised because the gaze is not stable when transitioning to the new position. This test has been conducted previously to the ablation study to test the best camera parameters, the best camera position and the loss of accuracy that the use of glasses may cause to the system. After that, the test has been carried out with multiple users to evaluate the system. The system's accuracy is evaluated using the Root Mean Square Error (RMSE) between the gaze projection on the screen and the position of the points. Accuracy is shown per coordinate $[X, Y]$ and in total. The results are shown in screen percentage, pixels and millimetres.

2.4.2. Camera Parameters

The ablation study starts with the analysis of the camera parameters, to find the best ones that suit the system. For this experiment, different resolutions and frame rates have been employed to test the precision of each parameter. The used camera is the ZED from Sterolabs, which allows four different resolutions with different frame rates, as shown in Table 2.1. In the performance column (Hz column), the ability of OpenFace to process frames is shown. The user repeats the test five times to reach a better conclusion on the best camera parameters to be used in the following sections.

This experiment shows that the X axis gets the best accuracy using 2k resolution at 15 frames per second. Instead, the Y axis gets the best accuracy using HD720 at 60 frames per second. However, taking into account the two axes' results, the best accuracy is

obtained for HD1080, which is better to the linear calibration results obtained in previous work [107], where the best camera parameters were 720p at 60 frames per second. In conclusion, HD1080 at 30 frames per second will be the camera parameters used for the next tests performed with the NARMAX calibration.

Results show a considerable improvement from the linear calibration method, which obtained an error of 60.5 pixels, 3.5 times higher than the one obtained with this nonlinear method.

Table 2.1: Testing accuracy vs camera parameters on a 1920x1080 screen using NARMAX calibration method. The test is done by one person looking at eight points on a screen. The test was done 5 times.

Method	**Camera parameters**		**Acc_x**			**Acc_y**			**Acc_{total}**			**Hz**
	Resolution	**fps**	**%**	**pix**	**mm**	**%**	**pix**	**mm**	**%**	**pix**	**mm**	
NARMAX	HD2K	15	**0.7**	12.5	3.9	1.3	14.5	4.5	1.1	20.3	6.3	14.8
NARMAX	HD1080	30	0.7	12.6	3.9	1.1	11.8	3.7	**0.9**	17.3	5.4	29.47
NARMAX	HD720	60	1.6	29.9	9.3	**0.9**	10.2	3.2	1.3	24.7	7.7	53.45
NARMAX	VGA	100	0.9	18.2	5.7	1.1	12.4	3.9	1.1	20.2	6.3	76.16
Linear	HD720	60	1.9	36.1	11.3	4.0	43.7	13.6	3.2	60.5	18.9	50.74

2.4.3. Camera Position

Camera position is crucial for the system to work. For this purpose, the camera is placed in front of the user at different heights (10, 48 and 78 cm). These heights are measured from the table where the screen is located.

Table 2.2 shows the results obtained for this test. The best accuracy is obtained when the camera is just in front of the user, 48 cm above the table. Nevertheless, this localization is not possible in a real car, as it would hide the road scene. Placing the camera on the top localization is not allowed because most of the landmarks of the face are occluded, and OpenFace 2.0 does not work well. The model does not solve this shortage because it cannot work with that noisy landmarks. Therefore, the bottom position should be chosen to place the camera in real vehicles because it achieves satisfactory results and allows naturalistic driving.

Even though the bottom position is the most suitable for an application on the vehicle, the middle one is used for the following experiments due to the fact that this position is the best one for the in-lab application and does not occlude the simulator view.

Table 2.2: Testing accuracy vs camera position on a 1920x1080 screen. The test is done by one person looking at eight points on a screen. The test was done 5 times.

Position	**Acc_x**			**Acc_y**			**Acc_{total}**		
	%	**pix**	**mm**	**%**	**pix**	**mm**	**%**	**pix**	**mm**
Top	-	-	-	-	-	-	-	-	-
Mid	**0.7**	12.6	3.9	**1.1**	11.8	3.7	**0.9**	17.3	5.4
Bot	1.0	18.6	5.8	1.5	16.1	5.0	1.3	24.1	7.5
Mid linear	1.9	36.1	11.3	4.0	43.7	13.6	3.2	60.5	18.9
Bot linear	2.5	48.0	15.0	10.6	114.5	35.8	7.7	147.9	46.2

2.4.4. Glasses

Once the ablation study has been carried out to find the best camera parameters, it is time to test the performance when the user wears glasses. Some drivers wear glasses while driving because of medical restrictions, whereas others wear sunglasses for convenience. The results previously obtained with the linear calibration method showed that it does not work with a user wearing sunglasses, which limits the usability of the system.

NARMAX calibration method shows that performance is even better using glasses than without them. This result may be due to the fact that the glasses are rigid elements where OpenFace can find robust visual features. Even so, the differences between both results are less than a pixel, which is not representative. Both tests were conducted using uniform lighting conditions, which will differ from a vehicle experiment where the sun may cause reflexes on the glasses. The results are depicted in Table 2.3. With this calibration method, it is possible to track the gaze even with sunglasses as it bears face parameters. These parameters are an addition of taking only the gaze features, which are occluded by the glasses.

Table 2.3: Testing accuracy vs glasses on a 1920x1080 screen. The test is done by one person looking at eight points on a screen. The test was done 5 times using NARMAX calibration

Eyes-glasses	**Acc_x**			**Acc_y**			**Acc_{total}**		
	%	**pix**	**mm**	**%**	**pix**	**mm**	**%**	**pix**	**mm**
Free	0.7	12.6	3.9	1.1	11.8	3.7	0.9	17.3	5.4
Glasses	**0.6**	11.6	3.6	**1.0**	11.3	3.5	**0.9**	16.4	5.1
Sunglasses	0.9	17.5	5.5	1.3	13.6	4.2	1.1	21.1	6.6
Glasses linear	2.5	48.7	15.2	4.6	50.0	15.6	3.7	71.7	22.4

2.4.5. Precision Test

Following the ablation study and the opportunity to work with any user, regardless of whether they wear glasses, an evaluation has been performed with multiple users to get statistical results. The test was done by ten users who calibrated the system following the procedure explained before to obtain a comparison with the method presented in [107] in terms of precision. Results can be seen in Figure 2.7.

The presented method achieves about 1.02% error on the X axis and 1.04% error on the Y axis, which is an error three times less than the obtained by the linear calibration as can be seen in Table 2.4 The proposed system achieves similar accuracy in both axes. It is a significant improvement over previous methods because it predicts a circular focalization map instead of an ellipsoidal one, which is the correct approximation for a focalization error. Figure 2.8 shows the error overprinted on an image with a crash object to compare scales for both tested systems. Furthermore, the linear calibration error is displayed in order to check the difference between both methods. This new model offers a better

approximation to obtain the focalization map generated by a driver than our previously linear version.

Table 2.4: Testing accuracy on a 1920x1080 screen. The test is done by 10 users looking at eight points on a screen using the NARMAX calibration method.

	$\mathbf{Acc_x}$			$\mathbf{Acc_y}$			$\mathbf{Acc_{total}}$		
	%	**pix**	**mm**	**%**	**pix**	**mm**	**%**	**pix**	**mm**
10 people	**1.02**	19.5	10.4	**1.04**	11.2	6.0	**1.03**	19.7	6.2
25 people linear [107]	1.9	36.1	11.3	4.0	43.7	13.6	3.2	60.5	18.9

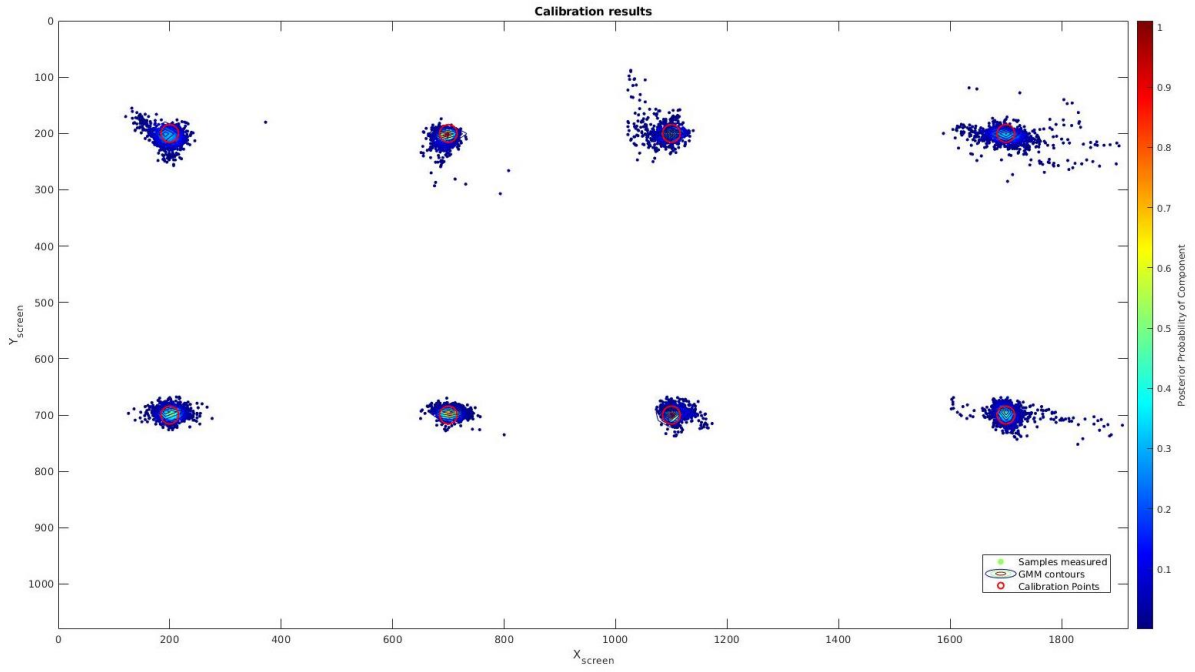


Figure 2.7: Calibration results using NARMAX calibration method. Eight points are displayed on the screen (Red circle). The proposed system creates 8 Gaussian around the testing points.

2.4.6. DADA2000 Evaluation

The presented system based on a NARMAX model shows the potential to have an adequate focalization method to be used for driving applications. To evaluate its potential, it will be tested on a challenging video benchmarking composed of accident videos that include driver attention and annotated accident localizations (DADA2000). With this dataset, the system can be evaluated to estimate accidents in video sequences.

The results are compared with those found by the dataset's authors using an expensive active desktop-mounted eye-tracker. Some users watched the same videos in similar conditions.

This dataset is provided with three annotated pieces of information. Firstly, it offers the crash-object position (accident localization in pixel per frame). Secondly, a focalization map captured with a desktop-mounted eye-tracker for some users is annotated. Lastly,

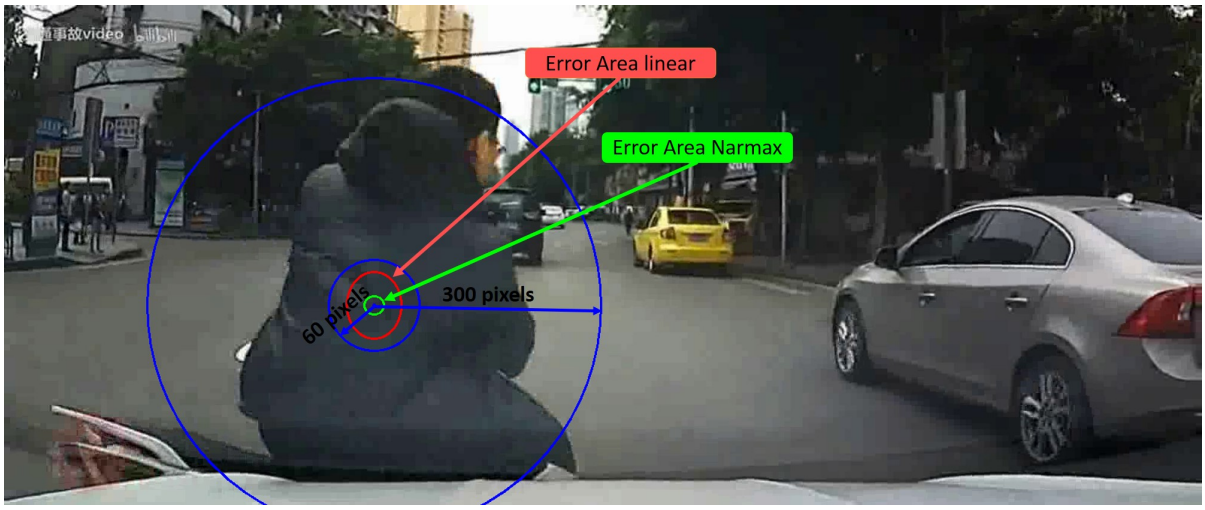


Figure 2.8: DADA2000 frame. Crash object circle for 60 and 300 pixels (blue). Error linear calibration area (red). Error NARMAX calibration area (green).

the temporal window that involves the accident is exposed. The videos provided are partitioned into three main clips: before, during and after the accident, as it is depicted in Figure 2.9. The accident clip is also divided into three sub-clips (Start, Mid and End) for a better analysis.

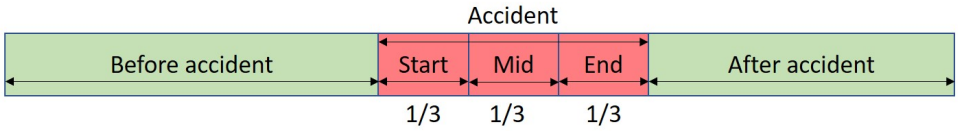


Figure 2.9: Temporal partition of the video. Dividing the accident part into three pieces of the same length to evaluate the results by sections.

2.4.6.1. Crash Object Detection in DADA2000 Benchmark

Crash object detection has been tested with four users. They were invited to watch the videos to evaluate the system in this dataset. Between videos, users made pauses of 3 s to accommodate the gaze for the following video. The dataset was resized from 1584 $\times$ 660 to 1920 $\times$ 1080 to match it with the screen resolution used in the experiment by the original authors to widen the working field of vision.

The clips are displayed at 30 fps to have a fair comparison with the results achieved by the authors of DADA2000. The attention map is recorded without temporal aggregation, obtaining one measure per frame because it works at 30 frames per second, which gives one focalization measure per frame.

As the main proposal of this work is to study a gaze focalization system, the goal of this test is to localize driver attention on a screen. For this purpose, the same evaluation methods are applied as in the original work. They measured the distance between the crash object centre, which is manually annotated, and the attention point obtained for

each frame. After that, they used some thresholds (Th) in pixel to establish if the detection is below that Th or not. With these premises, two indicators are obtained (frame ratio and success rate).

Results are compared with the ones obtained by DADA2000 authors using a Senso Motoric Instrument (SMI) RED250 desktop-mounted infrared eye-tracker for different detection area sizes as is shown in Table 2.5. The achieved success rate is higher than the one obtained by the DADA200 authors for all the sections except for start clips at a threshold between 160 and 300. However, the obtained results are for four users per video instead of the five they claim to use. Nevertheless, the authors claim that the videos were seen by at least two users, so they were not all seen by the five they claim to employ.

For a 300 pixels threshold, the success rate is higher than a 90% for the mid and end sections of the accident. This test shows worse outcomes in the start section but is close to the results claimed by the authors of the dataset. Mid and end sections show better results than those claimed by the authors.

Results show that early accident detection is worse than for the middle and end partitions because users take time to react and focus their gaze on the accident area.

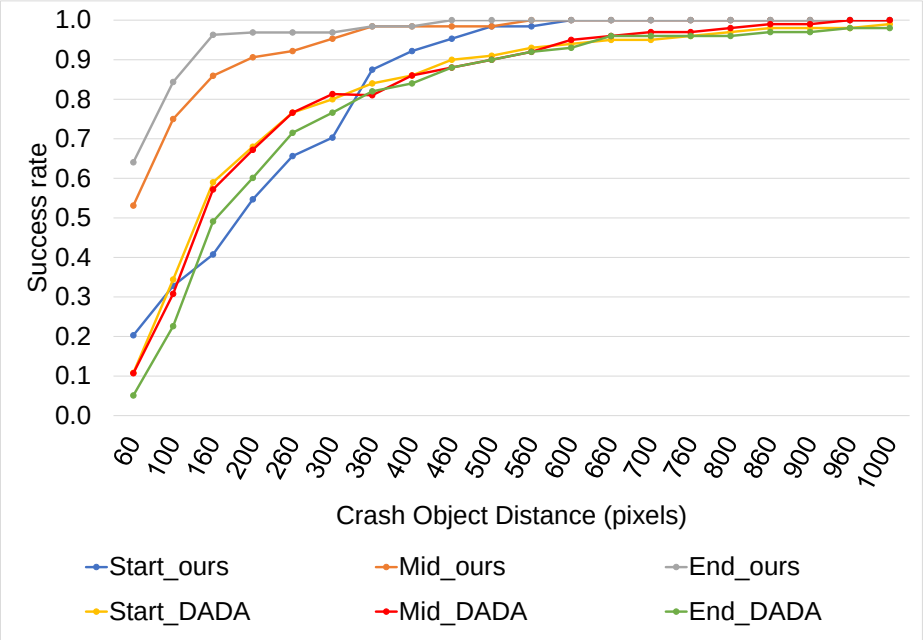
Table 2.6 shows the Frame ratio, which evaluates the distance between the focalization and the accident frame by frame. The frame ratio is higher for all the thresholds.

Table 2.5: Success rate: percentage of the number of clips that frame ratio for the entire clip is more than half of the frames of the complete accident scene.

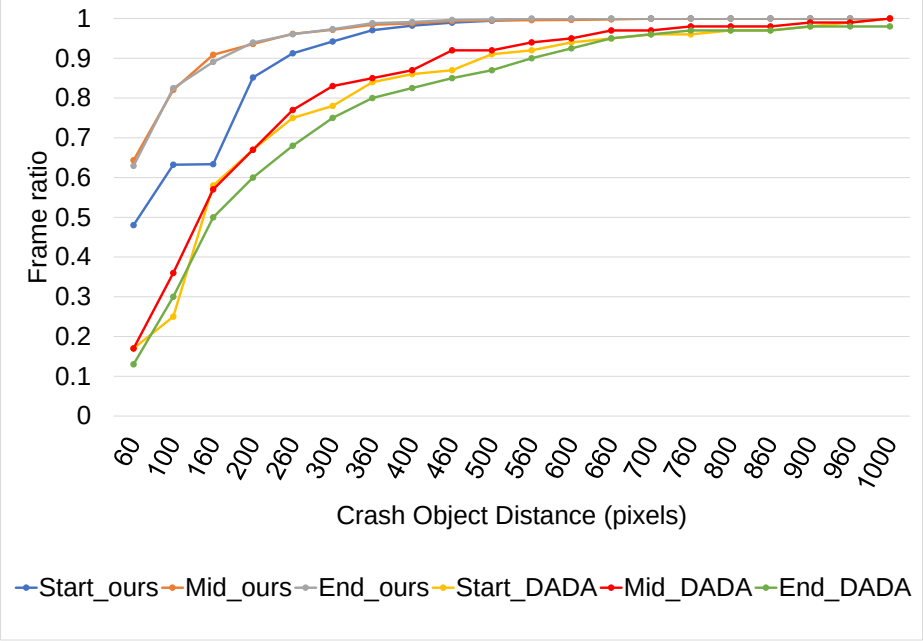
System	**Our gaze focalization system**			**DADA2000**		
Th (pixels)	**Start**	**Mid**	**End**	**Start**	**Mid**	**End**
60	20.3%	53.1%	64.1%	10.8%	10.7%	5.1%
100	32.8%	75.0%	84.4%	34.4%	30.8%	22.6%
160	40.7%	85.9%	96.3%	59.0%	57.2%	49.1%
200	54.7%	90.6%	96.9%	68.0%	67.2%	60.1%
260	65.6%	92.2%	96.9%	76.6%	76.6%	71.5%
300	70.3%	95.3%	96.9%	80.0%	81.3%	76.6%
360	87.5%	98.4%	98.4%	84.0%	81.0%	82.0%
400	92.2%	98.4%	98.4%	86.0%	86.0%	84.0%
460	95.3%	98.4%	100%	90.0%	88.0%	88.0%

Table 2.6: Frame ratio: percentage of frames in each clip section where the attention point is inside the correct detection area. It is measured frame by frame because the crash object changes with it.

System	**Our gaze focalization system**			**DADA2000**		
Th (pixels)	**Start**	**Mid**	**End**	**Start**	**Mid**	**End**
60	48.0%	64.3%	62.9%	17.0%	17.0%	13.0%
100	63.2%	82.0%	82.5%	25.0%	36.0%	30.0%
160	63.4%	90.9%	89.1%	58.0%	57.0%	50.0%
200	85.2%	93.6%	94.0%	67.0%	67.0%	60.0%
260	91.3%	96.2%	96.1%	75.0%	77.0%	68.0%
300	94.2%	97.2%	97.3%	78.0%	83.0%	75.0%
360	97.1%	98.4%	98.8%	84.0%	85.0%	80.0%
400	98.2%	98.8%	99.1%	86.0%	87.0%	82.5%
460	99.0%	99.4%	99.6%	87.0%	92.0%	85.0%



(a) Success rate obtained by the system proposed using NARMAX calibration.



(b) Frame ratio obtained by the system proposed using NARMAX calibration.

Figure 2.10: Obtained Results on the testing using NARMAX calibration.

The success rate and frame ratio curves are shown in Figure 2.10. It shows the curves obtained for the different Th (60 to 1000 pixels) and divided by the three sub-clips in which the accident is split, as done by the original author in [32]. The achieved results outperform the obtained by the baseline for all the analysed sub-sections except for the success rate start section, which has a lower curve in some ranges of the curve (160–400 pixels).

2.4.6.2. Heat Focalization Map on DADA2000



Figure 2.11: Attention map with temporal aggregation for the frame t, t-1 s, t-2 s and t-3 s using NARMAX calibration method.

A heat focalization map is the visualization tool used to show qualitative results for this method. For this purpose, it is necessary to establish a temporal window, which has been set to 1 s, as done in the DR(eye)VE project. As this method works at 30 fps, the window is composed of 30 measures.

The experimental procedure is shown in Figure 2.2, where the user under test is looking at the accident while the system is evaluating their gaze focalization over the accident displayed and is recording results in real-time. The generated attention map by the samples is also shown. As can be seen, the obtained focalization map is included in

the baseline map, showing the usability of the proposed system but using a cheaper and non-intrusive sensor than the authors of the dataset.

Figure 2.11 shows how temporal aggregation can help to predict where the accident will happen in some sequences taken from DADA2000. Displayed accidents involved other vehicles where the ego-vehicle is an observer of the accident. This figure shows the heat map created by four users. The figure shows two accident situations with their corresponding focalization maps in the image where the accident takes place as a heat map. Images are displayed with a temporal difference of 1 s. As can be seen, the heat map can correctly predict all the crash object positions. Generating driver attention maps is fundamental to this work to get where drivers look at how they are driving, as a way to predict their driving behaviours.

2.5. Chapter Summary

This chapter presents a new low-cost and non-intrusive method to obtain visual focalization maps using the NARMAX method as an improvement over the work proposed on [107].

The system is composed of a passive camera that can be mounted on every vehicle. Moreover, it is satisfactory enough to be an alternative to active desktop-mounted eye-trackers (baseline) or head-mounted eye-tracker, which are intrusive, costly and difficult to be installed in a vehicle. The system has been validated in the challenging public dataset DADA2000 [32], showing on-par results with the baseline in similar conditions. It has been shown that the proposed system could be a competitive technique to monitor driver focalization over other state-of-the-art proposals. It provides a safety supplement to the vehicle to make driving safer for all users. In addition, a novel real-time quantitative Driver Situation Awareness (DSA) metric has been proposed, which will be validated in simulated scenarios in Chapter 5.5.

Chapter 3

ARAGAN: A dRiver Attention estimation model based on conditional Generative Adversarial Network

"Necesito creer que algo extraordinario es posible."

Alicia Larde Nash; Una mente maravillosa

This chapter tackles the model implemented to predict the driver's attention. The idea is to model the "eyes" of the vehicle, so that they watch as a human does. The work in this chapter was partially published in the conference IV 2022 [115]: "ARAGAN: A dRiver Attention estimation model based on conditional Generative Adversarial Network", where it was accepted as a regular paper and required to be presented in the workshop on "Human Factors in Intelligent Vehicles". This chapter is structured following the section distribution of the paper (Introduction, Related works, ARAGAN architecture, Evaluation metrics, Experiments and Chapter summary).

3.1. Introduction

Intelligent vehicles have made multiple advances in recent years intending to achieve the fully autonomous driving architecture, defined as the fifth level (L5) according to the J3016 SAE International Standard [116]. L5 architectures are divided into different tasks to fulfil the outstanding goal of autonomous driving. This chapter is focused on visual scene understanding, which includes tasks such as pedestrian/vehicle detection and tracking, signs/traffic lights recognition or semantic segmentation. But do autonomous

Table 3.1: Attribute comparison with existing driver attention datasets..

Dataset	Duration	Collection Method	Field of View	Scenario(s)
DR(eye)VE [29]	6.1 hrs	in-car	wide	highway, countryside, downtown
BDD-A [31]	3.5 hrs	in-lab	narrow	downtown
DADA-2000 [32]	6.1 hrs	in-lab	N/A	accidents
EyeCar [118]	3.5 hrs	in-lab	wide	accidents

vehicles need to detect all the objects on the scene or only the ones that affect them? Do humans recognize the complete scene when they are driving or just a few objects?

Humans keep their visual attention on the most salient objects to focus their attention on a potential hazard, be aware of the traffic situation and make some decisions based on it. Human attention is a fusion of two mechanisms: bottom-up (colour or intensity) and top-down (goals or intention) [49], [50]. This fusion enables drivers to focus only on the necessary objects to get sufficient information to make decisions [51], [52]. This concept, transferred to the task proposed here, optimises the visual scene understanding [53], saving computational efforts when the vehicle is driving in a complex scenario [54].

Traditionally, driver gaze focalization has been used to obtain the driver's attention on the scene. Recently some datasets have been published to explore this field in intelligent vehicles applications. The first one to come out was DR(eye)VE dataset project [29], which recorded and annotated 550,000 frames of driving sequences in different traffic and weather conditions. Every frame provides the driver's gaze obtained through an accurate eye tracker (SMI ETG 2w Eye Tracking Glasses) while driving. After that, Berkeley DeepDrive Attention (BDD-A) [31] dataset was launched, claiming that it is nearly impossible to collect enough driver attention data for critical events with a conventional in-car data collection protocol, like DR(eye)VE project did. They claimed that this protocol only captures a single focus and false positive gazes, which will confuse the attention model during training. To overcome this concern, they proposed an in-lab recorded dataset after being looked at different driving scenes on a screen by several experienced drivers with an EyeLink 1000 desktop-mounted infrared eye tracker. With this new protocol, DADA-2000 dataset [32] came out, which enabled annotating critical accidental situations with this mechanism. It collected 658,476 frames from accidents and annotated them using an in-lab protocol with experienced drivers' gaze to obtain their attention with Senso Motoric Instruments (SMI) RED250 desktop-mounted infrared eye tracker. Up to the best of our knowledge, the referenced datasets are the only ones that tackle driver attention and are public nowadays. Many works have been focused on obtaining the driver's attention through deep learning models trained on these datasets to substitute a gaze focalization system. More competitive proposals are based on conditional Generative Adversarial Nets (cGANs) [117] because they can generate distributions without a big dataset. Table 3.1 summarises the main driver attention datasets in the literature.

The release of Transformers [70] in 2017 supposed a change in the deep learning paradigm regarding attention mechanism as it effectively allowed Multi-Head Attention. Firstly, this technique was used to outperform Natural Language Processing (NLP) models. Further, it has recently been implemented in computer vision applications. Several works have overcome state-of-the-art results. Like [119] that presents a Transformer architecture tested on different benchmarks, including image-level classification [120], region-level object detection [121] and pixel-level semantic segmentation [122].

This dissertation chapter presents a driver attention estimation model based on a cGAN [123]. It is a network that learns to map an image of distribution (b) from another of distribution (a) mapping a loss function during training. cGANs differ from GANs [124] in the fact that the Generator and the Discriminator are fed with a condition instead of with only random noise. In this case, they are fed with an RGB image extracted from a traffic scene obtaining an attention image as output. Besides, the cGAN Generator uses the encoder of a Transformer, providing the architecture with Multi-Head Attention capacity. Up to the best of our knowledge, this is the first time that these two techniques (cGAN and Transformer) have been used together to predict driver attention.

The main contribution of this chapter is the novel and efficient architecture proposed, which could be used in real-time applications, due to its high performance, outperforming other state-of-the-art proposals on open-source datasets.

3.2. Related Works

In recent years, prior efforts have been made in the literature to get driver attention models using computer vision techniques. Moreover, some new Deep Learning (DL) techniques have been used in intelligent vehicle applications. For this reason, this section splits into two subsections the literature review. The first one will expose the driver attention paradigm in the literature, with the datasets and some proposed models. Secondly, this work presents novel deep learning techniques, which need to be understood in the literature review.

3.2.1. Driver Attention

Research has been conducted in the field of fixation models based on computer vision for some years using DL techniques [125]–[127]. These are the first-ranked models on the MIT300 benchmark [128]. The works in [129], [130] are the first-ranked models on the CAT2000 benchmark [131]. However, it was not until the release of the DR(eye)VE dataset [29] that these models started to be used in intelligent vehicle scenarios.

With this release, several DL models were trained on this dataset, including a proposal by the dataset authors in [29], based on a multi-branch architecture. Three branches (RGB, Flow estimation and Semantic Segmentation) composed the model and were fused to obtain the attention prediction. Other trained models were proposed on [49], [62] and [132].

After this dataset, Berkeley DeepDrive Laboratory launched its Berkeley DeepDrive Attention dataset (BDD-A) [31], which not only proposed a new in-lab protocol to estimate the driver's attention but obtained the peripheral vision, through which humans can fixate their gaze on an object while attending to others. In consequence, multiple fixations must be identified in an image. To include this functionality, authors proposed using more than one observer to capture the attention map. They presented their dataset and a model trained on it. The proposed model used a feature extractor based on AlexNet and pre-trained in ImageNet [133]. The input was based on six consecutive frames with a temporal processing unit in the form of 2D convolutions and Convolutional LSTM (Conv2D-LSTM).

With this new in-lab protocol, another dataset came out, DADA2000 [32], which not only proposed a new dataset based on accidents but also proposed an architecture to predict driver's attention [64]. Their proposed model is based on a multi-branch architecture with a semantic-guided attentive fusion to predict the attention map from the RGB image and the semantic segmentation. Like its predecessors, this model used a sequence of images as input with Convolutional LSTM (Conv2D-LSTM).

In recent years, other works aimed to be different from the state-of-the-art models in this field. Authors in [118] proposed a Maximum Entropy Deep Inverse Reinforcement Learning model to predict visual attention in driving scenarios. They also announced a dataset to be tested by the community, but it is not fully public. They predict the eyes' movement, which is the action of their inverse reinforcement learning model.

3.2.2. Deep Learning Techniques

In order to build the application described earlier, some DL techniques have been proposed. Adversarial learning was proposed in [124]. It learns to generate a data distribution (Generator), while a discriminative model (Discriminator) learns to estimate if a sample comes from the ground truth or the Generator. This training procedure corresponds to a minimax two-net game. The image-to-image generation has been a topic developed in the DL field for a long time, but it was not until [134] came out that it was implemented with conditional adversarial networks. They proposed a procedure to solve different image processing problems. Their Generator was based on the U-Net [135]. Their Discriminator was based on a Markovian discriminator denominated PatchGan. Driver's attention through adversarial learning has recently been presented in [117], which was based on the architecture described in [134].

Apart from adversarial learning, another disruptive trend is changing the DL paradigm, Multi-Head Attention. Because not all stimuli are equal, focus attention has enabled the human species to direct attention to objects of interest, such as prey and predators have been doing in the complex visual environment throughout history. Finding the saliency regions of a traffic scene making the most of Transformers [70], which allows efficient Multi-Head Attention capability, is the crucial aspect of this chapter.

Before this architecture, some others estimated attention. The most commonly used ones were additive attention [136] and dot-product (multiplicative) attention [137]. Different approaches have implemented Transformers for computer vision tasks despite they were first designed to be used in NLP tasks. In [138], they used an attention module based on Channel Attention 'what to pay attention to' and Spatial Attention 'where to pay attention to'. This implementation was based on the work proposed in [139], which implements channel attention differently. These works show progress in using attention mechanisms in computer vision tasks. But they did not end up using the Transformer. In [140], Self-Attention is employed to model an adversarial learning evaluated on the ImageNet-1K [120] dataset. Self-Attention is the base of the work proposed on [70] but still needs some modifications to explore the Transformers' capability. Finally, [119] presented a work where the Transformer was used for computer vision tasks. They tested and got state-of-the-art results on different benchmarks, such as, ImageNet-1K image classification (V1 and V2) [141], [142], COCO object detection [121], ADE20K semantic segmentation [143] and Kinetics-400 video action classification [144].

There is a recent work [145] that has presented a driver attention model using the fusion of Transformers and convolutions. This model has been trained on BDD-A but obtains worse performance results than the one obtained with our proposal in the same test dataset.

After this literature review and up to the best of our knowledge, it can be claimed that the architecture proposed here is the first that fuses the novel DL techniques cGANs and Transformers to predict driver's attention for real-time intelligent vehicles applications.

3.3. ARAGAN Architecture

Thanks to the advances on novel state-of-the-art DL techniques (cGANs and Transformers), ARAGAN: A dRiver Attention estimation model based on conditional Generative Adversarial Network is proposed. ARAGAN improves state-of-the-art results, opening the possibility to perform real-time applications due to the efficiency of its architecture. Figure 3.1 shows the framework of the proposal, where the Generator, Discriminator and the information flow generated by these two nets are represented. Besides the networks, loss function calculation is included graphically.

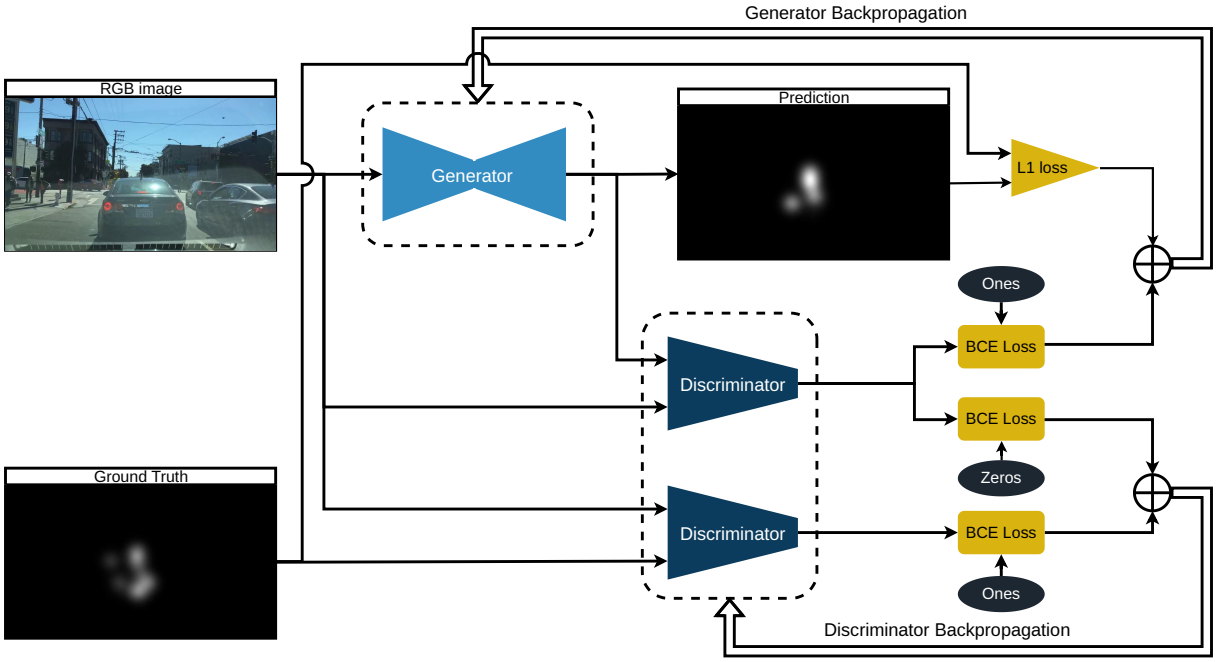


Figure 3.1: ARAGAN framework. Generator, Discriminator and the flow of the generated information by these two nets are represented.

Using as baseline the Pix2Pix architecture [134], which presented a cGAN that learns from an image to obtain another one, this work proposes to use conditional adversarial learning for driver attention prediction. The network (Generator) is fed with an RGB image of the traffic scene (condition), expecting to obtain a predicted attention map over the image. It predicts "fake" attention maps that should be distinguished from the "real" ones by the Discriminator, which is trained adversarially to identify these synthetic attention maps. Both nets apply the gradients at every batch.

Hereafter, the learning strategy applied in this work, the different attention modules used in the Generator design and the final architectures proposed for the Generator and the Discriminator are presented.

3.3.1. Adversarial Learning Procedure

Before starting the adversarial learning procedure, contextual information about the loss function will be explained. In the context of an optimization algorithm, a function to evaluate the model is necessary. The optimization algorithm usually seeks to minimize that function. This function is called the cost function, loss function or error function.

Within these functions, there are different solutions for each problem. For the proposed application, it is required to delve into the Binary Crossentropy loss. This loss function works for multi-class and multi-label classification. It measures how far the prediction is from the actual value (0 or 1). This function follows the Equation 3.1, where x is the prediction and y is the ground truth. The penalty is logarithmic, offering a small score for small differences (0.1 or 0.2) and an high score for a sizeable difference (0.9 or 1.0).

$$\mathcal{L}_{BCE}(y, x) = \frac{1}{N}\sum_{i=0}^{N}[y \cdot log(x) + (1 - y) \cdot log(1 - x)] \tag{3.1}$$

With that exposed, it is time to introduce the loss function inside the proposal model. It follows the function previously described but with some modifications to train an adversarial learning model. The loss function of a cGAN is shown in Equation 3.2, where the Generator (G) tries to minimize it, and the Discriminator (D) aims to maximize it. While x is the RGB image, y is the driver attention from the ground truth, and z is random noise. This logic follows what authors claimed in [134]. The added random noise is removed from the input image as the Generator learns to avoid the noise as presented in [146]. Namely, the net is provided with Dropout that adds noise to training without jeopardizing the input.

$$\mathcal{L}_{cGAN} = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(G(x, z))] \tag{3.2}$$

To fulfil that requirement (Equation 3.2), where one model needs to minimize the function and the other one to maximize it, the loss function has been split into two losses, one for each model. The Generator model follows the Equation 3.3, where J_n is an all ones matrix with the output shape of the Discriminator; and $D(G(x, z), y)$ is the output of the Discriminator fed with the RGB image and the attention map from the ground truth.

$$\mathcal{L}_{Generator} = \mathcal{L}_{BCE}(J_n, D(G(x, z), y)) \tag{3.3}$$

Besides, an L1 distance is added to the loss function, represented in Equation 3.4. It is based on the work presented in [147], where they claimed that adding a traditional loss to the Generator benefits its performance.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[y - G(x, z)||_1] \tag{3.4}$$

With this said, the final loss function for the Generator is shown in Equation 3.5.

$$\mathcal{L}_{Generator} = \mathcal{L}_{Generator} + \lambda\mathcal{L}_{L1}(G) \tag{3.5}$$

As this proposal is an adversarial learning procedure, the Discriminator loss function requires to be configured. It has two steps, the first one is the loss of the real output (Equation 3.6) and the second one is the loss of the generated output (Equation 3.7). The combination and final function follows the Equation 3.8, which is the sum of both intermediates functions.

$$\mathcal{L}_{disc_real} = \mathcal{L}_{BCE}(J_n, D(x, y)) \tag{3.6}$$

In Equation 3.6, J_n denotes an all ones matrix with the output shape of the Discriminator and $D(x, y)$ is the output of the Discriminator fed with the RGB image and the attention map from the ground truth.

$$\mathcal{L}_{disc_fake} = \mathcal{L}_{BCE}(0_n, D(G(x, z), y)) \tag{3.7}$$

In Equation 3.7, 0_n denotes an all zeros matrix with the output shape of the Discriminator and $D(G(x, z), y)$ is the output of the Discriminator fed with the RGB image and the generated attention map.

$$\mathcal{L}_{disc_total} = \mathcal{L}_{disc_real} + \mathcal{L}_{disc_fake} \tag{3.8}$$

3.3.2. Attention Modules

In the following section, some attention mechanisms are exposed to give a theoretical background about them. Attention mechanisms in deep learning are a way to selectively focus on certain parts of the input, allowing the model to make more informed decisions. The main idea behind attention is to weigh the importance of different input sections so that the model can focus on the most relevant information. Attention mechanisms are particularly useful in tasks where the input is large and complex, such as natural language processing or computer vision. The attention mechanism allows the model to focus on specific parts of the input, which improves the model's performance and reduces the amount of computation required. Attention mechanisms have been applied to various tasks such as image captioning, machine translation, and language modelling.

This subsection introduces some attention modules used in the Generator design. These architectures are explained from the point of view of convolutional use. All of them are used with convolutional layers, which efficiently compute spatial information to understand their relationships.

Chapter 4 will introduce other attention mechanisms and explain how to use them with linear layers instead of convolutional ones.

3.3.2.1. Soft-Attention

One popular type of attention mechanism is called "Soft-Attention". This type of attention uses a neural network to learn a probability distribution over the input, which weighs the importance of different input parts. It is calculated as a weighted sum of the encoder's hidden states (n). It was introduced in [148]. The attention weight σ_i for the input element x_i is computed as follows:

$$\sigma_i(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \quad for\ i = 1, 2, \ldots, K \tag{3.9}$$

The attended output z is then computed as:

$$z = \sum_{i=1}^{n} \sigma_i(x_i) x_i \tag{3.10}$$

3.3.2.2. Hard-Attention

Another type of attention mechanism is "Hard-Attention". This type of attention uses a deterministic function to select a specific part of the input rather than a probability distribution [149]. Unlike the previous method, attention scores are used to select a single hidden state instead of the weighted average of all hidden states. The attention weight a_i for the input element x_i is computed as:

$$a_i = \text{argmax}(x_i) \tag{3.11}$$

Where argmax returns the argument or arguments for the target function that returns the maximum value from the target function. The attended output z is computed as:

$$z = x_i a_i \tag{3.12}$$

The problem with this attention mechanism is that it is not differentiable because the argmax function is not. This means that backpropagation cannot be done through this module.

3.3.2.3. Convolutional Block Attention Module (CBAM)

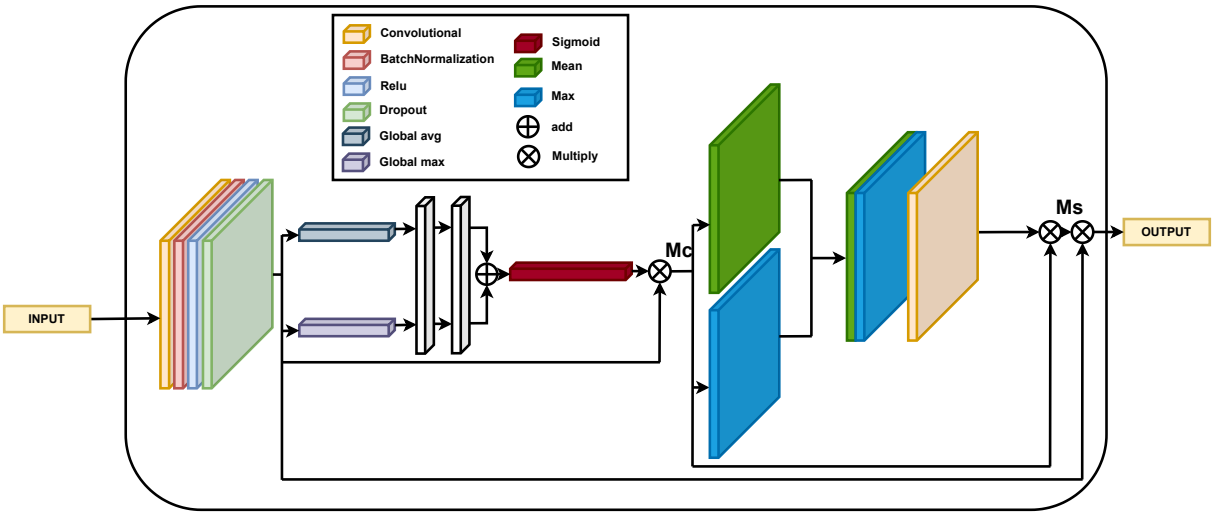


Figure 3.2: Convolutional Block Attention Module (CBAM).

This module was proposed in [138]. With an intermediate feature map, this architecture infers attention in two ways: channel ("what to pay attention to") and

spatial ("where to pay attention to"). Figure 3.2 shows the architecture of this module. The Channel attention (M_c) is done as shown in Equation 3.13 and the Spatial attention (M_s) as exposed on the Equation 3.15. The input to this module is a feature map, and the output is an attended feature map that can have stride or not.

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{3.13}$$

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \tag{3.14}$$

In Equation 3.13, σ denotes the sigmoid function (represented in Equation 3.14), and MLP is a shared multi-layer perceptron of one hidden layer with an activation size set to $\mathbb{R}^{\frac{C}{r}x1x1}$, where r is the reduction ratio.

$$M_s(F) = \sigma(f^{7x7}([AvgPool(F);\ MaxPool(F)])) \tag{3.15}$$

In Equation 3.15, σ denotes the sigmoid function, and f^{7x7} is a convolutional layer with a kernel size of 7x7.

3.3.2.4. Self-Attention Module

Self-Attention module [140] has been claimed in the literature as one of the best attention procedures. A well-known example of this type of attention is the Transformer [70]. In this architecture, the input pays attention to itself. Figure 3.3 represents this module. The input to this module is a previous feature map, and the output is an attended feature map of the same dimension. The input feature map undergoes three transformations. The first two are query (q) and key (k), which are used to calculate the attention, where $q(x) = f_q^{1x1}x$ and $k(x) = f_k^{1x1}x$. With these two feature maps, Equation 3.16 learns to attend to the ith location (pixel) when synthesizing the jth region, building the attention map of the module.

$$\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} exp(s_{ij})},\ where\ s_{ij} = q(x_i)^T\, k(x_j) \tag{3.16}$$

The output layer of the attention (Equation 3.17) has the form of $o = (o_1, o_2, ..., o_j, ..., o_N) \in \mathbb{R}^{CxN}$, where C is the number of channels of the input feature map and N is the number of feature locations from the previous hidden layer. This output is calculated after a matrix product of the attention layer ($\beta_{j,i}$) and the last transformation experienced by the input, the values (v).

$$o_j = f_h^{1x1}(\sum_{i=1}^{N} \beta_{j,i} v(x_i)),\ where\ v(x) = f_v^{1x1}x \tag{3.17}$$

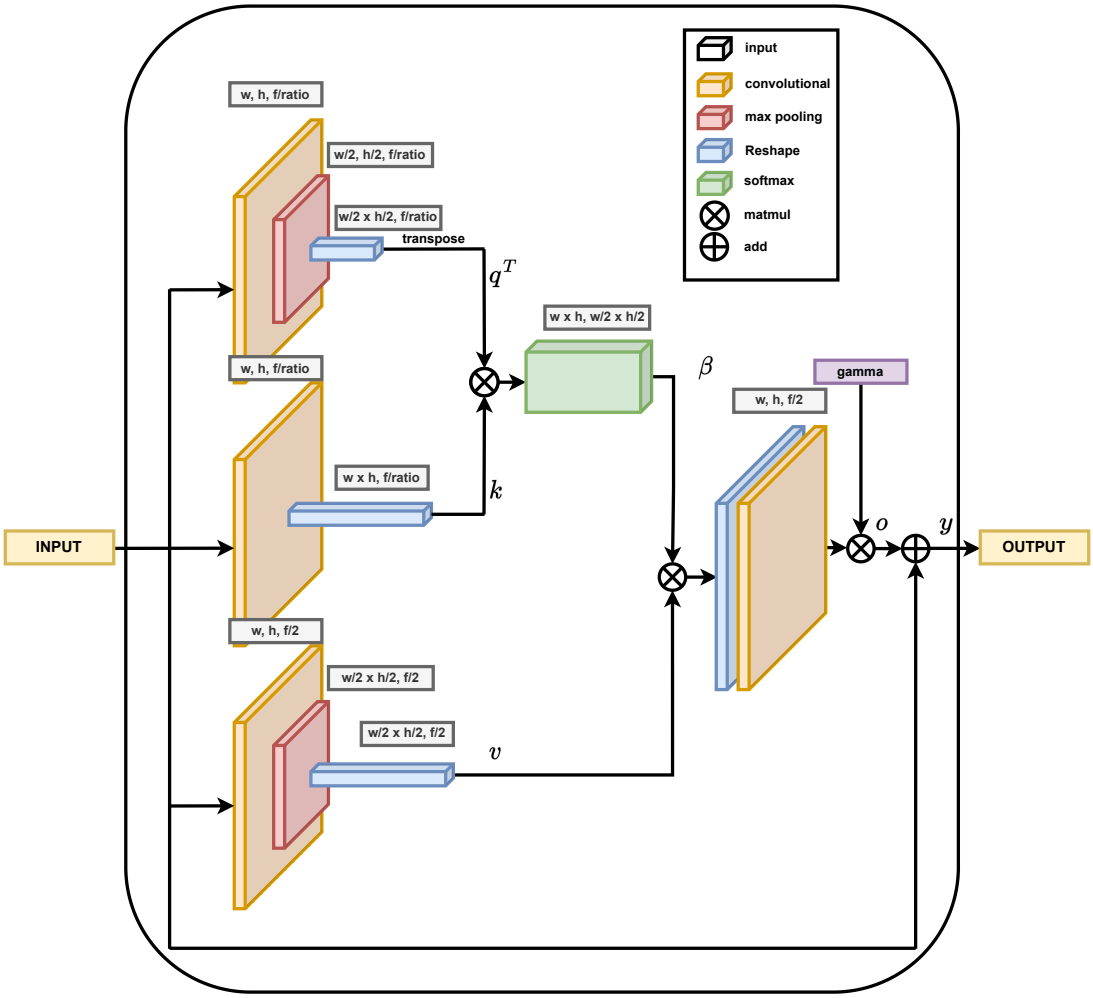


Figure 3.3: Self-Attention module

Convolution operations, with a kernel size of 1, are done to extract the feature maps. Using convolutions instead of linear layers allows having a spatial representation of the feature map. These operations are the following ones: f_q^{1x1}, f_k^{1x1} and $f_v^{1x1} \in \mathbb{R}^{C/rxC}$, where r is a decreasing factor to increase efficiency without compromising performance, and $f_h^{1x1} \in \mathbb{R}^{CxC}$.

$$y_i = \gamma o_i + x_i \tag{3.18}$$

Additionally, the output is multiplied by a learnable scalar γ, and the feature input is added as a residual connection, as shown in Equation 3.18. This scalar is initialized to 0 to provide the ability to the network to first rely on the local neighbourhood and then assign more weight to the non-local evidence. Residual connections mainly help mitigate the vanishing gradient problem.

3.3.2.5. Multi-Head Self-Attention Module

Another type of attention mechanism is Multi-Head Attention, which implements previous attention layers concatenated in parallel. It allows parallel computing and provides the module with the ability to pay attention to different aspects in the same feed. Multi-Head Attention has been widely used in deep learning, particularly in Natural Language Processing (NLP), where it is used in every new model that comes out. The main idea behind Multi-Head Attention is to use several weight sets to attend to different parts of the input vector rather than a single section. It allows the model to attend to various aspects and to make more informed decisions.

It was introduced in the Transformer model [70]. This proposal uses the attention mechanism to calculate a weighted sum of the input items. The weights are computed based on the similarity between the input items and three representations (query, key and value). These different weight sets are concatenated and passed through a feed-forward neural network to obtain the final attention weights. The use of Multi-Head Attention has been shown to improve the performance of the Transformer model on several NLP tasks, such as machine translation and language modelling. The ability to attend to different aspects of the input is valuable for tasks that involve understanding the relationship between distinct parts of the input, such as language comprehension.

Moreover, it can be used in other deep learning tasks, such as image captioning [150] and computer vision. For example, it has been used by the Vision Transformer (ViT) [151]. ViT uses Multi-Head Attention to process the image in a patch-based manner, allowing the model to learn about the spatial relationships between the different pieces of the image.

In conclusion, Multi-Head Attention is a powerful variation of the attention mechanism that allows the model to attend to different aspects of the input. It has been widely used in NLP tasks and has been shown to improve the model performance. It has also been used in other applications, such as computer vision.

Its architecture can be seen in Figure 3.4. The input to this module is a feature map, and the output is an attended feature map of the same dimension. To build the Multi-Head Self-Attention mechanism, M Self-Attention modules, as described above, are concatenated, which act as heads in this new architecture. The Self-Attention module repeats its computations multiple times in parallel. The concatenated feature maps are fused through a convolutional layer f^{1x1}. They constitute the final output of the Multi-Head Self-Attention mechanism, which is a final Attention score. It gives the architecture greater power to encode multiple relationships and nuances for each feature. After that, the output followed by a residual connection and a Layer Normalization [152]. Unlike Batch Normalization, Layer Normalization directly estimates the normalization statistics from the summed inputs to the neurons within a hidden layer so the normalization does not

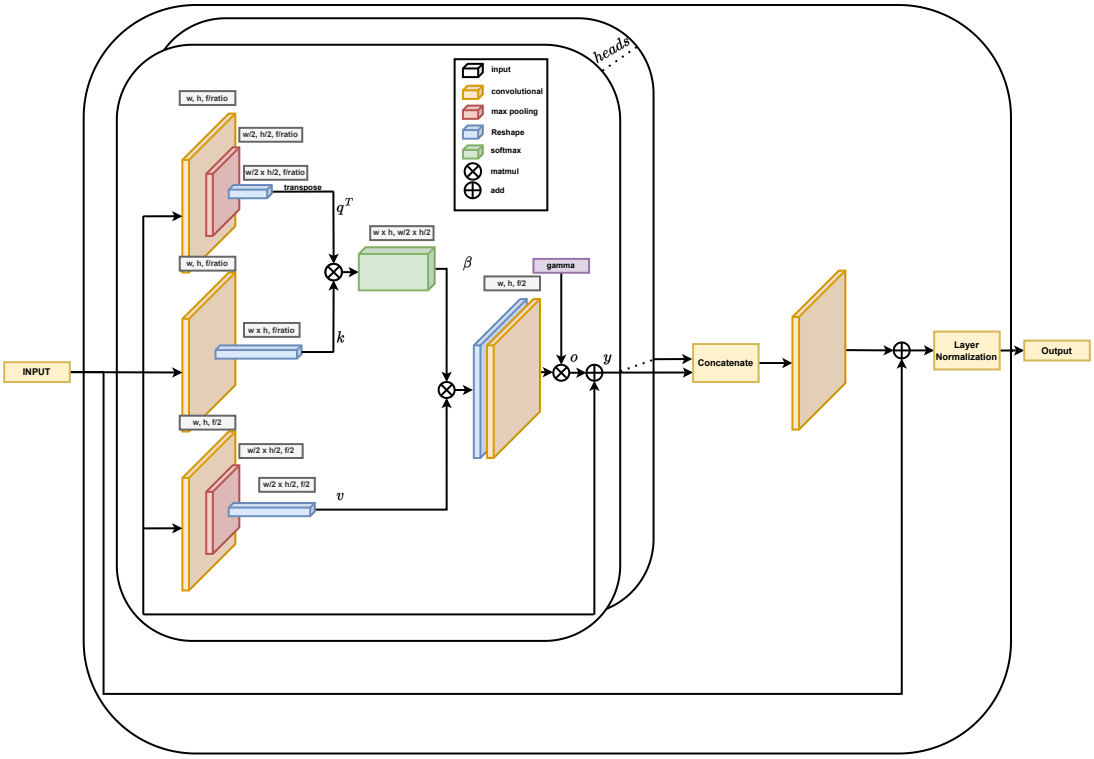


Figure 3.4: Multi-Head Self-Attention module.

introduce any new dependencies between training cases. Layer Normalization performs the same computation at training and test times.

This mechanism allows the network to learn different aspects of the "meaning" of each feature map. This mechanism is shown in Equation 3.19.

$$MultiHead(x_i) = f_o^{1x1}(Concat(head_1, head_2, \ldots, head_M)) \\ where\ head_i = \gamma o_i + x_i \tag{3.19}$$

3.3.3. Generator Architecture

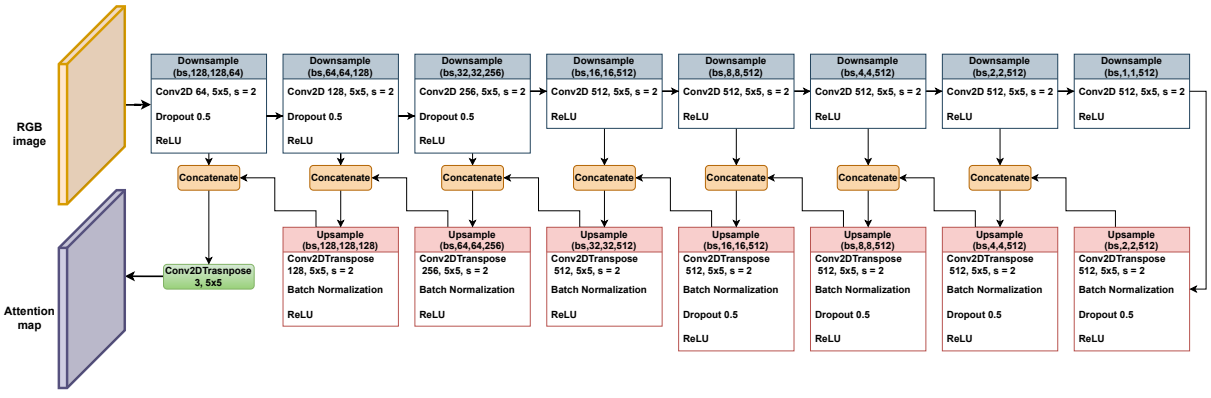
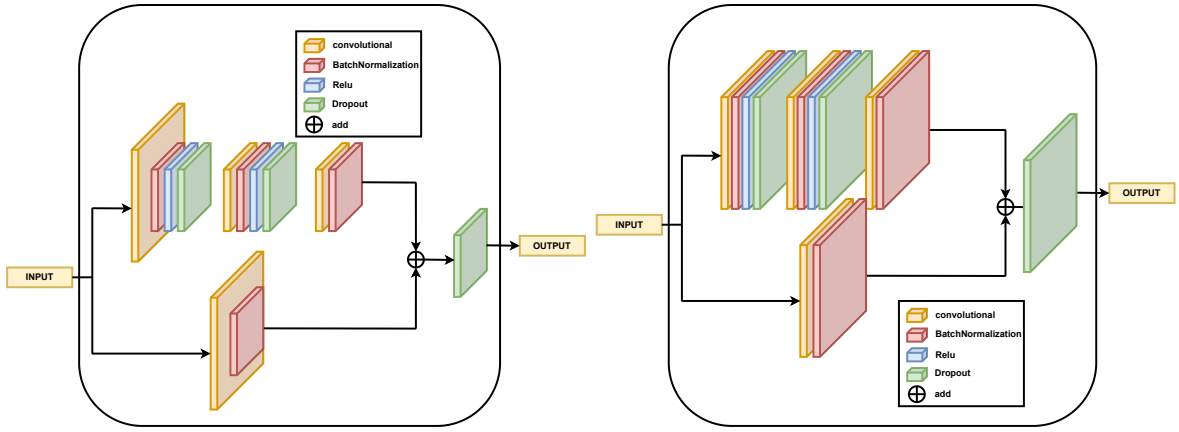


Figure 3.5: First approach. U-Net [135] architecture for driver attention estimation.



(a) Residual convolutional module (Resblock) with stride = 2 for dimension reduction.

(b) Residual convolutional module (Resblock) with stride = 1

Figure 3.6: Residual convolutional modules used as an image feature extractor.

Using the different attention modules and the learning procedure explained above, as well as a Residual convolutional module with and without stride as presented in Figure 3.6, the Generator design is ready to be tackle.

Firstly, a Generator based on [134], which follows the architecture of the U-Net [135], is implemented (Figure 3.5). It is an encoder-decoder architecture built by convolutions with skip connections between mirrored layers of the encoder and the decoder. This architecture is a powerful option for many image-to-image problems. The information about the input is shared with the output, and this functionality is helpful for some tasks. For example, in semantic segmentation, the output has the same object shapes as the input, so this information could be passed in earlier steps of the network instead of vanishing it with a deeper network. This information is not so important to the proposed application, as the shapes are not maintained. The encoder is built with convolutional modules made up of a Convolutional Layer $f^{3\times3}$, a Dropout of 0.5 and a ReLU activation function. The decoder is built with upsample modules. They are composed of a Convolution Transpose, Batch Normalization, a Dropout of 0.5 and a ReLU activation function. The input is a 256×256 RGB image, and the output is a 256×256 attention map. These sizes are the same in all implementations. With that said, this is the first Generator of this proposal. It will be treated as a baseline and proof of what can be achieved in driver attention estimation with adversarial learning.

As second proposal, the convolutional section of the previous Generator is changed for an encoder based on CBAM (Figure 3.7), which implements the first attention module proposed and achieves a better understanding of the features. The proposed Generator is composed of a first-stage module with a convolutional layer ($f^{7\times7}$), Batch Normalization, a ReLU activation function and a Max Pooling ($f^{3\times3}$) with stride = 2. After that, fifteen CBAM modules, as explained before (Figure 3.2), are used. The odd ones have stride = 1, and pairs have stride = 2. Afterwards, a decoder is implemented to upsample the feature maps obtained to get the final output as a 256×256 attention map.

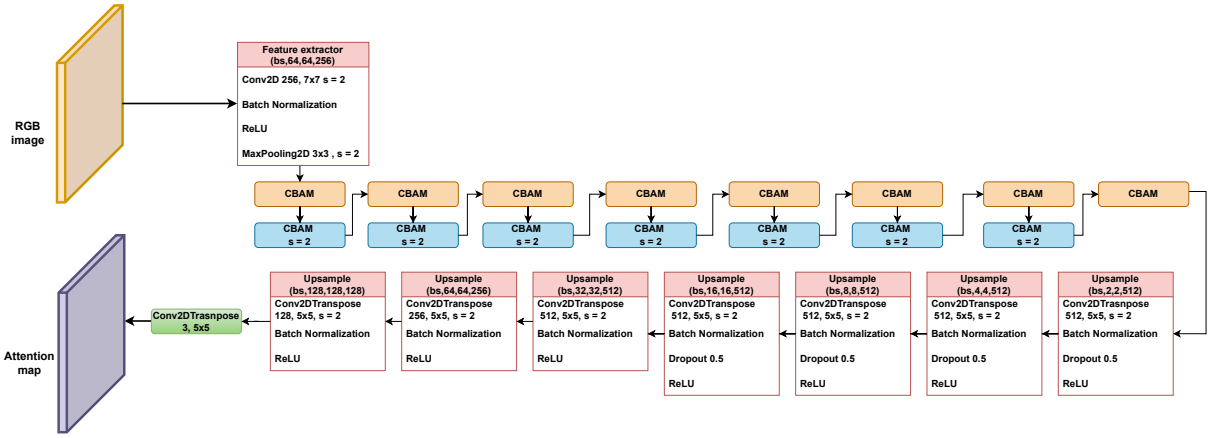


Figure 3.7: Second approach. Encoder based on Convolutional Block Attention Modules (CBAM) for driver attention estimation.

The third approach is based on the Self-Attention mechanism (Figure 3.3). This generator configuration is a fusion of Residual convolutional downsample modules (Figure 3.6) with a Self-Attention module. It takes advantage of the Residual convolutional downsample modules to reduce the dimension of the features and the Self-Attention to attend to that features smartly. It has the same stride configuration as the CBAM-based Generator (Figure 3.8). After downsamples and feature extraction are conducted, a Self-Attention module is implemented to extract the most interesting characteristics of these feature maps. It is made to feed the decoder with this crucial information to obtain the final output as a 256×256 attention map.

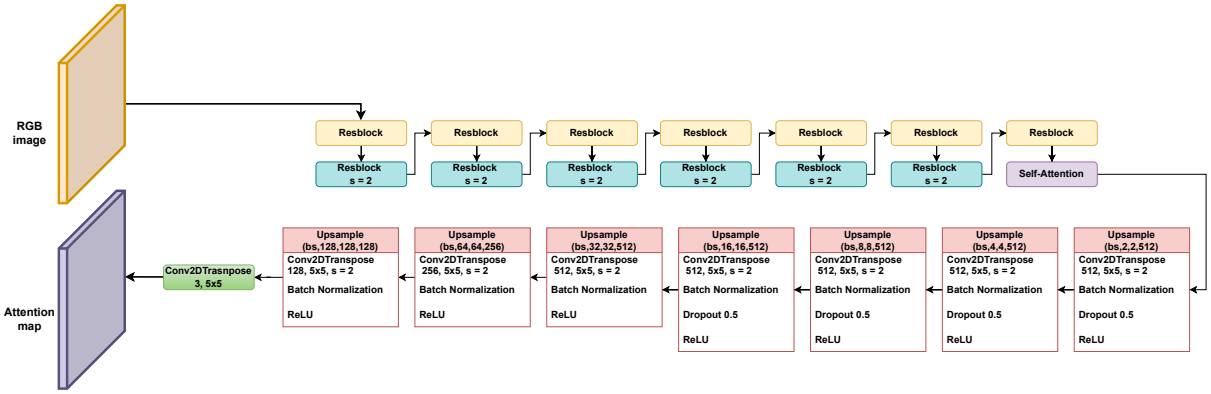


Figure 3.8: Third approach. Residual modules with Self-Attention mechanism for driver attention estimation.

The final proposal takes advantage of a Transformer Encoder to provide the architecture with Multi-Head Self-Attention capability in an efficient way (Figure 3.9). It has convolutional steps based on Resblocks downsample (Figure 3.6a). The deconvolutional steps follow the same architecture as the previous models tested. It uses the Multi-Head Self-Attention modules at different steps of the downsample procedure to take advantage of the benefits of this architecture. It provides parallelism where each head can attend to different parts of the features input. It can learn a more

comprehensive representation of the input data. It is a robust architecture to input variations. It is able to capture complex relationships between data.

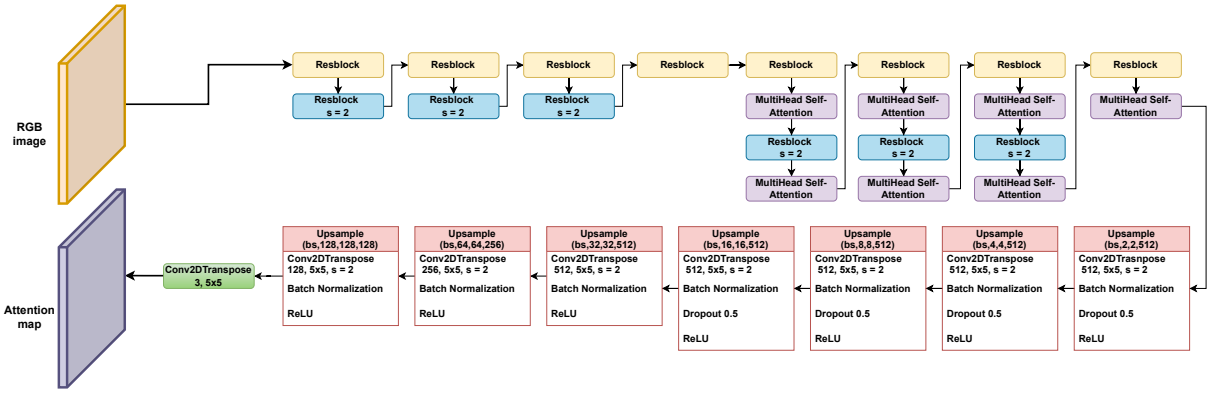


Figure 3.9: Fourth and final approach. Residual modules with Multi-Head Self-Attention modules for driver attention estimation.

3.3.4. Discriminator Architecture

Due to the adversarial training, an additional net is required to complement the Generator. This net has been built following the implementation explained in [134]. In this work, they proposed using a convolutional PatchGAN classifier, previously introduced in [153]. This classifier, instead of classifying between "real" or "fake" for the complete image, classifies $N \times N$ patches, where N is much smaller than the image size. It only penalizes the structure at the scale of patches. The authors proved that N can be much smaller than the image and produces similar results. Among its benefits, it allows using a Discriminator with fewer parameters, which trains faster and uses larger images.

This network experiences two forward passes. Firstly, the network receives the generated attention map and the RGB image. Its output is a patch of $33x33$ that is compared with an all-ones matrix for the Generator loss and an all-zeros matrix for the Discriminator loss, as previously explained. The same that before but in this case the output patch is compared with an all-ones matrix for the Discriminator loss.

The concatenated images are passed through some downsample modules based con Conv2D, Batch Normalization and Leaky ReLU activations to end up with the patch classification. The architecture is represented in Figure 3.10, showing the data processing during the forward pass.

3.4. Evaluation Metrics

The metrics applied in saliency models are used to evaluate the proposed models. These metrics are collected and explained in [65]. From the cited metrics, the following

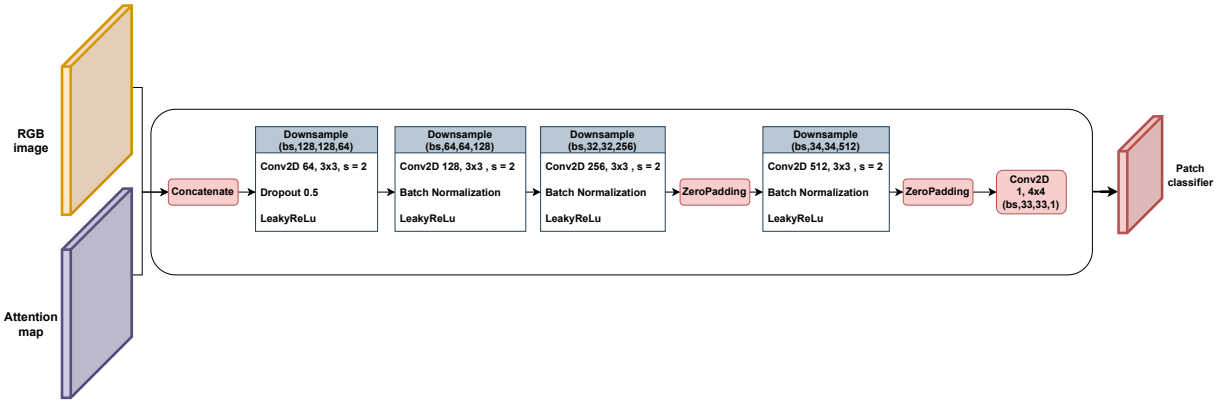


Figure 3.10: Discriminator architecture

ones will be used: Kullback-Leibler Divergence (KLD), Pearson's Correlation Coefficient (CC) and shuffled Area Under the ROC curve (s-AUC). Hereafter, each of them will be briefly explained.

Kullback-Leibler Divergence (KLD) shows the difference between two probability distributions. Equation 3.20 shows the KLD metric. The distributions are P and Q^D, and ϵ is a regularization constant. In the proposal explained here, the two comparing distributions correspond to the generated output and the Ground Truth attention map.

$$D_{KL}(P, Q^D) = \sum_i Q_i^D log\left(\epsilon + \frac{Q_i^D}{\epsilon + P_i}\right) \tag{3.20}$$

Pearson's Correlation Coefficient, CC, also called Linear Correlation Coefficient, is a statistical method generally used to measure how correlated or dependent two variables are. CC can be used to consider saliency and fixation maps, P and Q^D, as random variables to measure their linear relationship. Equation 3.21 shows how it is calculated, where $\sigma(P, Q^D)$ is the covariance of P and Q^D.

$$CC(P, Q^D)) = \frac{\sigma(P, Q^D)}{\sigma(P) \times \sigma(Q^D)} \tag{3.21}$$

Shuffled Area Under the ROC Curve (s-AUC) is a modification of the Area Under the ROC Curve (AUC), which is one of the most used metrics in saliency literature. The s-AUC assumes that the centre bias has not been modelled and penalizes models where this happened. This metric penalizes when the model only predicts the attention map on the centre of the image due to the fact that it is the common area to pay attention to.

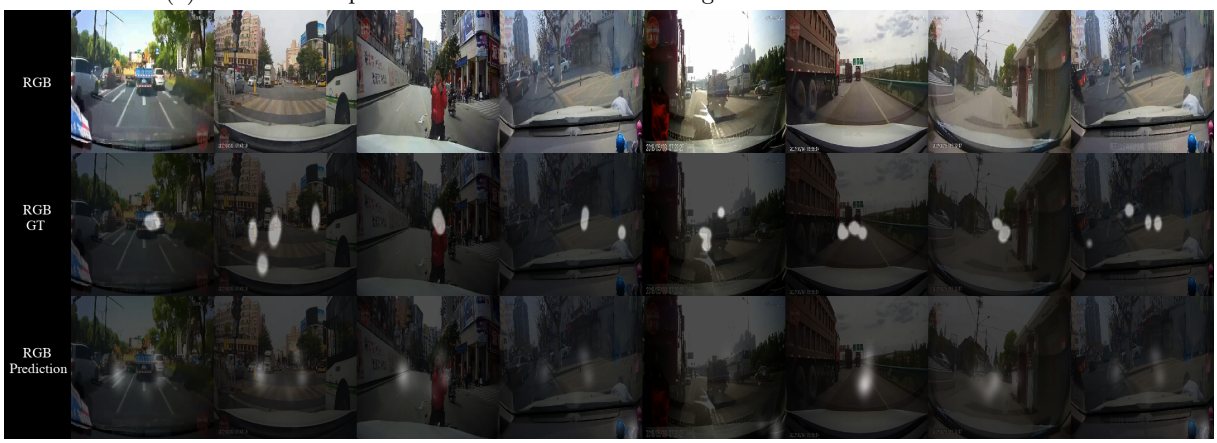
3.5. Experimental Results

This section presents the results obtained for the different proposals. It is divided into two main experiments. Firstly, an ablation study is carried out to test the cGAN

architecture with different Generator models. Secondly, the best model is compared with other state-of-the-art proposals in order to evaluate its contribution. Architectures were trained using Adam optimizer with a declining learning rate and parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. A batch size of 8 is used. The GPU used for all the experiments described below is the NVIDIA 2080 Ti with 11GB of VRAM. Both experiments have been quantitatively analysed with the metrics exposed above.



(a) Attention maps obtained in the BDD-A testing set. Model trained on BDD-A.



(b) Attention maps obtained in the DADA2000 testing set. Model trained on BDD-A.

Figure 3.11: Qualitative results for the model proposed using the combination of Residual modules + Multi-Head Self-Attention.

3.5.1. Ablation Study

All the architectures are trained for 15 epochs on the BDD-A training set (98,984 images). Images and attention maps are resized to 256×256. After the training procedure, a testing step is carried out. This testing phase is done in two different datasets. Firstly, the BDD-A testing set is used (32,253 images). After that, it continues with the DADA-2000 testing set composed of accidental scenarios (64,255 images). This procedure has been done to prove the domain adaptation of the presented work.

Table 3.2 shows two different metrics. It is divided into accuracy metrics (KLD, CC and s-AUC) and performance metrics (Training time and inference frequency). Performance

Table 3.2: Ablation study performed using adversarial learning with different generators configuration. All models have been trained in BDD-A and tested in BDD-A and DADA2000.

Generator	Accuracy metrics						Performance metrics	
	BDD-A dataset			DADA dataset				
	KLD ↓	CC ↑	s-AUC ↑	KLD ↓	CC ↑	s-AUC ↑	Training time	Inference (Hz)
U-Net	0.44	0.91	0.54	0.43	0.94	0.55	10h 45m 25s	7.70
CBAM	0.27	**0.92**	0.68	0.27	0.94	0.69	6h 12m 39s	7.67
Resblocks + SA	0.13	0.87	**0.71**	0.22	0.96	**0.70**	**5h 26m 24s**	8.70
Resblocks + MHSA	**0.05**	**0.92**	0.66	**0.10**	**0.97**	0.65	6h 30m 52s	**10.53**

metrics are not commonly evaluated in the literature, but for us, it is a crucial concept that needs to be assessed, as it proves the ability of the model to work during inference in real time. For that, training time and inference frequency are presented.

Different proposals for the downsample section of the Generator are evaluated: U-Net (baseline), CBAM, Self-Attention (SA) and Multi-Head Self-Attention (MHSA).

Results show that the best model for the downsample section of the Generator is a combination of the residual modules with the Multi-Head Self-Attention module. This achieves the best performance in both datasets for all metrics except for s-AUC and training time, where using the Self-Attention module gives better values. Moreover, the use of attention modules decreases the training time, which is an important fact in terms of research purposes.

3.5.2. Comparison with Other State-of-the-art Models

The best model obtained in the ablation study is compared with some other popular driver attention estimation implementations. Results are shown in Table 3.3. This proposal obtains better performance results than the referenced architectures in the two challenging datasets for KLD and CC metrics. And it is on par with the best architectures evaluated in s-AUC. Moreover, unlike most proposals of the literature, this model shows inference times that can be used in a real-time application at the level of decision-making.

Table 3.3: Comparison with other state of the art models trained in BDD-A and testing in BDD-A and DADA2000

Architecture	Accuracy metrics					
	BDD-A dataset			DADA dataset		
	KLD ↓	CC ↑	s-AUC ↑	KLD ↓	CC ↑	s-AUC ↑
DR(eye)VE [29]	1.95	0.5	-	3.37	0.29	0.63
BDD-A [31]	1.24	0.59	-	2.5	0.4	**0.66**
MEDIRL [118]	2.51	0.74	**0.67**	2.93	0.63	0.61
ACT-Net [145]	1.11	0.63	-	-	-	-
Resblocks + MHSA (ours)	**0.05**	**0.90**	0.66	**0.10**	**0.97**	0.65

Figure 3.11a shows the qualitative result for the proposed model. The images are displayed in the following order: the first row is composed of the RGB images, the second one of the ground truth (humans fixations) overlaid with the RGB images and the third one shows the ARAGAN's predicted attention map over the RGB images. The results show a high performance in predicting driver attention. They display different driving scenes with more or less complexity due to the number of agents in the scenario. They also show good performance in night situations.

Figure 3.11b shows the qualitative results of a model trained in BDD-A but tested in DADA-2000. The displayed images show different accidental situations and the predictions over them. The proposed model shows accurate results in accident scenarios, representing the most crucial areas to pay attention to under critical circumstances.

3.6. Chapter Summary

ARAGAN has been presented as a novel architecture that combines adversarial learning with Multi-Head Self-Attention, two of the most advanced techniques in DL, to obtain driver attention estimation.

This proposal outperforms state-of-the-art results in a comparison experiment with other popular models in two distinct challenging datasets (BDD-A and DADA2000). An extensive study of different architectures for the Generator design and the results obtained in both datasets have been presented. ARAGAN enables using the driver's attention estimation in real driving applications due to its low inference time and may help the development of autonomous vehicles.

In the next chapter, the Generator will feed another network in charge of a decision-making task taking advantage of the ability of seeing as a human does provided by this module.

Chapter 4

Driver's Attention-based Explainable Decision-Making

"La humanidad nació en la Tierra. Nunca tuvo intención de morir aquí"

Joseph Cooper; Interstellar

This chapter presents an explainable decision-making system based on driver attention and computer vision model. It is the final contribution to the driver attention paradigm of this work, closing the loop with the human, to help the building of the executive layer of a vehicle using the driver's scene understanding. The proposed work uses the driver's attention predicted by the ARAGAN Generator to obtain a decision and an explanation.

4.1. Introduction

Vehicle intelligence can be defined as decision-making [154]. It is the executive layer within the vehicle and is in charge of the final decision that concludes with a command to the vehicle's actuators. Within the options offered in the literature to create the decision-making layer, there are modular and end-to-end approaches. Furthermore, they can be done with classical methods or learning-based methods [155]. The modular approaches separate the complete AV procedure into modules, building a pipeline that constitutes the architecture [156], [157]. There is another approach that is based on end-to-end proposals. They work from the raw data extracted from the sensors to get directly the AV decision [158], [159]. This second approach is mainly based on deep learning techniques due to the abstraction and complexity of the task.

The explainability of the before approaches can be summarised in two ideas. The former has the advantage that the different sub-modules that compose the system can explain the decision-making. However, they have the problem of transmitting the error to

the following pipeline modules, which means that the error is cumulative. These modules are separated into layers: perception, localization, mapping, planning, decision-making and control.

On the other hand, end-to-end proposals lack the problem of error propagation. However, they are not interpretable, and an error is difficult to explain. Explaining end-to-end proposals is a complex task and has recently been addressed within the Explainable AI (XAI) paradigm. It has methods such as Shapley values [160] and saliency maps [68] to quantify the contribution of each input feature to the model output. This paper wants to differentiate itself from this tendency because it will not explain the features.

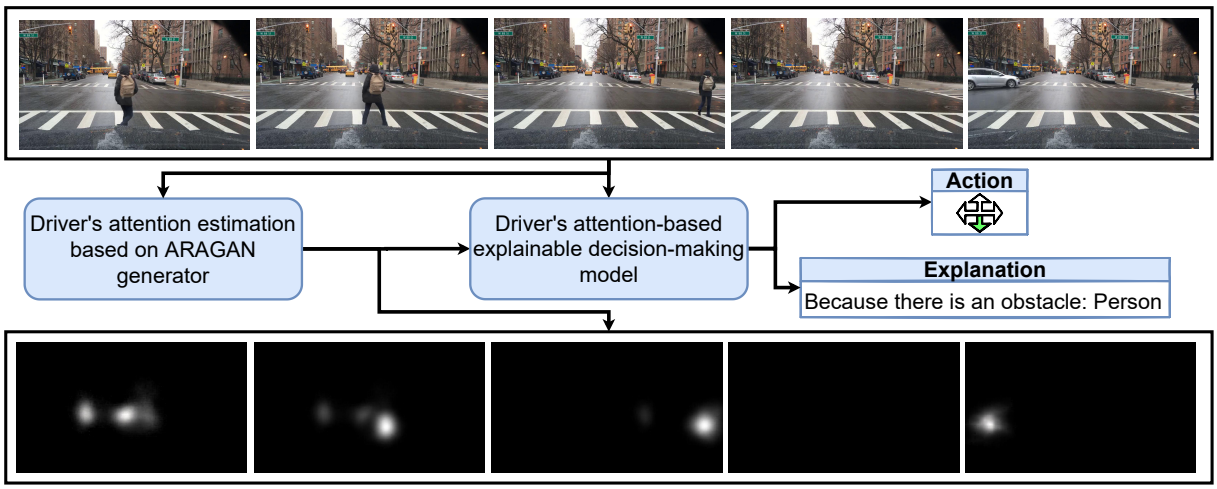


Figure 4.1: Driver's attention-based explainable decision-making framework.

For this purpose, this chapter proposes a driver's attention-based explainable decision-making model to predict the vehicle's action and the explanation of that motion. Figure 4.1 shows the framework of this proposal, where the ARAGAN Generator [115] predicts the driver's attention for the input sequence. The image sequence, the driver's attention sequence and the connection between modules to obtain the final output can be seen in Figure 4.1. It shows the importance of feeding the module with a sequence due to the achievement of events that took place during that time. As can be seen a pedestrian cross the street in the 3 first frames but in the last two frames is out of the field of view of the camera. The vehicle should brake in this situation. Moreover, the action and explanation for this behaviour can not be taken using only the last frame because temporal analysis is needed.

This module is the final contribution of this dissertation. It proposes a model based on driver attention for vehicle-to-driver interaction using deep learning. It predicts the decision and the explanation of a video sequence taken from the in front camera of a vehicle. This information could serve to assist the driver about the manoeuvre to be manually performed or to guide the imitation or reinforcement learning of an autonomous vehicle.

4.2. Related Works

The literature review section is split into subsections to summarise the explainability of autonomous driving using deep learning. The first part consists of a study of the open-source datasets within this topic to show the available data to accomplish this task. The following section describes approaches trained in the same dataset as this proposal to show their strengths and weaknesses. The last part exposes a literature review of some innovative deep learning techniques used in our model design.

4.2.1. Datasets

This work is based on supervised learning. Training deep learning models in a supervised way requires a huge amount of data, as models learn to generalize from seen data. For this purpose, there are many datasets that are open-source to reach different tasks. In the context of autonomous vehicles, object detection, semantic segmentation, and driver attention are usual tasks that have been accomplished using deep learning. These AVs tasks have their own datasets that allow training supervised deep learning models.

Another task that needs its own dataset is the development of an explainable decision-making model. There are different open-source datasets in the literature to learn models for this task. Table 4.1 summarises them with their sizes and the annotation method.

Table 4.1: Driving datasets that can be used to develop explanation methods for AVs and the stakeholders that would potentially benefit from such explanations.

Dataset	Size	Annotation & Explanation
BDD-OIA [76]	23K × 5s	Actions and *Why explanation*
BDD-X [74]	7K × 40s	Textual *Why explanation* associated with videos segments with heatmaps
DoTA [161]	4,677 videos (73,193s)	*What explanation* (Temporal and spatial anomaly identification with bounding boxes)
CTA [162]	1,935 × 17.7s	Why explanation for accidents with cause and effects
HDD [163]	374,400s	*What explanations* for driver actions
BDD-A Extended [164]	1,103 × 10s	Human gaze inciting *why and/or what explanation*, explanation necessity score

4.2.1.1. BDD-OIA

Authors of the BDD-OIA [76] claimed that in real-world driving situations, the decision-making is most of the time "move forward". But it is in short periods of time when the driver has to decide between some possible actions. For this reason, they have selected a subset of BDD100K [165] video clips containing at least five pedestrians or bicycle riders and more than five vehicles. They have chosen this procedure to have a dataset with complex scenarios and many objects to analyse to complicate the decision-making.

The selected videos have various weather conditions and were recorded at different times of the day. Their dataset is composed of 22,924 5-second video clips annotated for 4 actions (Move forward, Stop/Slow down, Turn left and Turn right) and 21 explanations. Table 4.2 shows the label distribution and the connections between them. This way of annotation allows the definition of a multi-label classification problem, which is easier to train because the outputs are bounded. However, dealing with this dataset is complicated because the same video has more than one possible output.

Table 4.2: Action and explanation categories in the BDD-OIA dataset. Changing lanes to left/right is merged with turn left/right to avoid distribution imbalance. For the turn left/right rows, the upper sub-row presents statistics of changing lanes to the left/right and the lower sub-row those of turning left/right. For these actions, explanations address why the action is not possible.

Action Category	Number	Explanations	Number
Move forward	12491	Traffic light is green	7805
		Follow traffic	3489
		Road is clear	4838
Stop/Slow down	10432	Traffic light	5381
		Traffic sign	1539
		Obstacle: car	233
		Obstacle: person	163
		Obstacle: rider	5255
		Obstacle: others	455
Turn left	838	No lane on the left	150
		Obstacles on the left lane	666
		Solid line on the left	316
	5064	On the left-turn lane	154
		Traffic light allows	885
		Front car turning left	365
Turn right	1071	No lane on the right	4503
		Obstacles on the right lane	4514
		Solid line on the right	3660
	5470	On the right-turn lane	6081
		Traffic light allows	4022
		Front car turning right	2161

4.2.1.2. BDD-X

The following dataset is BDD-X [74], that unlike the previous one, is not annotated as a classification problem. They annotated a subset of the Berkeley Deep Drive (BDD) dataset [166] with textual justifications. The sequences are 40 seconds long and were captured with a dashcam mounted inside the vehicle placed behind the rear-view mirror.

To annotate the dataset, the authors asked the annotator to imagine himself being a driving instructor. The annotation has to describe what the drivers were doing and its reason.

The dataset is composed of over 77 hours of driving, divided into 6,984 videos. They were recorded in diverse driving conditions, e.g., day/night, highway/city/countryside, summer/winter, etc. They have an average length of 40 seconds and around 3-4 actions. The dataset contains 8.4M frames with over 26k activities.

4.2.1.3. DoTA

Detection of Traffic Anomaly (DoTA) [161] is composed of 4,677 video clips collected from YouTube and dash camera videos from accidents. It is annotated in three ways:

- Temporal annotations: anomaly start and end times. Each video is separated into three sections: precursor (before the anomaly), the anomaly window and post-anomaly (after it).
- Spatial Annotations: the agents in the scene are labelled with unique IDs and bounding boxes.
- Anomaly Categories: each video is assigned an anomaly category. Table 4.3 shows the 9 categories that they use to label. However, they doubled these categories in order to differentiate between ego-involved and non-ego involved.

Table 4.3: Traffic anomaly categories in the DoTA dataset

ID	Anomaly Categories
1	Collision with another vehicle which starts, stops, or is stationary
2	Collision with another vehicle moving ahead or waiting
3	Collision with another vehicle moving laterally in the same direction
4	Collision with another oncoming vehicle
5	Collision with another vehicle which turns into or crosses a road
6	Collision between vehicle and pedestrian
7	Collision with an obstacle in the roadway
8	Out-of-control and leaving the roadway to the left or right
9	Unknown

4.2.1.4. CTA

Causality in Traffic Accident (CTA) [162] collected 1,935 videos of traffic accidents from the web. They annotated the semantic labels of the cause and effect and their temporal intervals at each accident video. The classification used can be seen in Table 4.4. It represents the source of the accidents and their results.

4.2.1.5. HDD

Honda Research Institute Driving Dataset (HDD) [163] collected 104 hours of driving around the San Francisco Bay Area. They proposed a 4-layer to represent diver behaviour: Goal-oriented action, Stimulus-driven action, Cause and Attention. The dataset has an average video duration of 45 minutes. Annotations include the following fields:

- Goal-oriented action: driver's action in a navigation context, such as right turn, left turn, branch and merge.
- Stimulus-driven action: actions caused by traffic participants or obstacles, such as stopping and deviating.
- Cause: explanation of the previous action. For example, a stopped car in front of us is an immediate cause for a stop.
- Attention: the traffic participant that is attended by the driver.

Table 4.4: Causality in Traffic Accident (CTA) cause and effects categories.

	Categories
Cause labels	Left Turn Across Path at Non-Signalized Junctions Control Loss Vehicle Changing Lanes - Same Direction Vehicle Turning - Same Direction Road Edge Departure Running Red Light Left Turn Across Path at Signalized Junctions Straight Crossing Paths at Non-Signalized Junctions Lead Vehicle Decelerating Lead Vehicle Stopped Evasive Action Backing Up Into Another Vehicle Vehicle Making a Maneuver - Opposite Direction Vehicle Not Making a Maneuver - Opposite Direction Object Crash Vehicle Turning at Non-Signalized Junctions Following Vehicle Making a Maneuver Other
Effect labels	Collision w/ Vehicle Collision w/ Road Obstacle Out Of Road Collision w/ Multiple Vehicle Rollover Stopped Collision w/ Object

4.2.1.6. BDD-A Extended

BDD-A Extended [164] collected 1,103 driving video clips from the BDD-A dataset [55], which is an attention dataset of braking events. They employed five annotators, who had to register the explanation of each sequence. For this purpose, they had to write a textual expression and the necessity of it with a score.

Concluding this point, this work is going to follow the BDD-OIA dataset [76] as it provides a sequence with the actions and the explanations that the driver did during a complex driving manoeuvre. Moreover, it is simplified to a double classification problem instead of a textual task. The prediction of an explanation in text form, as done in some datasets exposed above, is within the Natural Language Processing (NLP) paradigm, which is out of the scope of the proposed work that is focused on computer vision techniques.

4.2.2. State-of-the-art Models Trained in BDD-OIA

Some models in the literature have been trained on the BDD-OIA dataset. The following section describes these architectures used to solve the explainability problem in autonomous vehicles.

The first proposals [76] introduced the prediction of object-induced actions and their explanations. They used an object detection model and a second step called the Action-Inducting Object Selection module. The object detection module is the Faster R-CNN [167], which has been previously trained in BDD100K [165] to perform this task. After

this training, it was frozen to continue with the second step of the model. The local features of each object (bbox) were fed to the second module that fuses them with the global features (full image) extracted by the Faster R-CNN to select the best detections to complete the task.

$$\mathcal{L} = \mathcal{L}_{BCEaction} + \lambda \cdot \mathcal{L}_{BCEexplanation} \tag{4.1}$$

The authors proposed a multi-task loss to train the model. It follows Equation 4.1, which is based on a combination of the Binary Cross Entropy (BCE) (Equation 4.2) and a hyperparameter λ that controls the influence of the explanation. They set λ to 1 after an ablation study to determine the best option.

$$\mathcal{L}_{BCE}(y, x) = \frac{1}{N}\sum_{i=0}^{N}[y \cdot log(x) + (1 - y) \cdot log(1 - x)] \tag{4.2}$$

They fed the model with the last frame of the recorded sequence to obtain the action and the explanation for the complete sequence. They compared their result with other state-of-the-art works (ResNet-101 [168] and the network of [169]), outperforming them with their proposal. Their code is also public to replicate their experiments.

The second proposal [170] claimed that the first model [76] has two shortcomings: 1) It can "only" have a clear look into those cropped-out regions. These regions correspond to the detected object. 2) The proposed attention mechanism depends on the Faster R-CNN. Therefore, the error will be propagated to the rest of the modules.

They proposed a global soft attention (GSA) mechanism to overcome these problems. It allows the model to search for correlations more deeply. To complete this proposal, they proposed a model based on two feature extractors (Resnet50 [168] and Mobile Net [171]), which feeds a Multi-Head Self Attention. Therefore, GSA attends to the "raw" features instead of only a few ones (ROIs). They used the same loss as in the previous proposal based on Equation 4.1. Their code is not public to replicate their results.

4.3. Theoretical Background

This section introduces the theory to explain how the proposed model works and the concepts inside the presented modules. It is divided in these: Firstly, the different modules are explained in a theoretical way. These modules are the feature extractor, attention mechanisms, transformer encoder and final classification heads. Secondly, the section will continue with the theory within the proposed application, exposing the possible losses, the metrics used and how a multi-label classification works.

4.3.1. Feature Extractor

A widely used approach in the literature to solve computer vision problems is to utilise a pre-trained backbone in another dataset and fine-tune it for a target application.

The reason for this procedure is that these models have been trained on large datasets over the course of long periods of time, achieving good results in encoding images. After this training, this model can be used for other computer vision tasks changing only the end heads of the model or adding other modules to it.

This technique has been widely used in the literature [172]. The most used dataset for this task is ImageNet, composed of 1,281,167 training images, 50,000 validation images and 100,000 test images distributed among its 1,000 classes. Lately, Google has been using a private dataset called JFT-300M [173]. Tensorflow has pre-trained models in ImageNet that can be used for computer vision applications with the only restriction of using the same input as the original authors in terms of size and standardisation. Here is a list of some popular backbones that can be used in TensorFlow for computer vision, along with their associated papers or models, and a brief description of each one:

- VGG16: Very Deep Convolutional Networks for Large-Scale Image Recognition [174]. VGG16 is a convolutional neural network architecture that uses small filters (3×3) and deep networks (16 layers) to the time it was published. It achieved good performance on large-scale image recognition tasks.
- ResNet: Deep Residual Learning for Image Recognition [168]. ResNet is a convolutional neural network architecture that uses residual connections that allow training deep networks (up to 152 layers) without suffering from the vanishing gradient problem.
- Inception: Going Deeper with Convolutions [175]. Inception is a convolutional neural network architecture that uses a combination of different filter sizes ($1\times1, 3\times3, 5\times5$) in the same layer to increase the representational capacity of the network.
- MobileNet: Efficient Convolutional Neural Networks for Mobile Vision Applications [176]. MobileNet is a convolutional neural network architecture designed for mobile and embedded devices. It uses depth-wise separable convolutions to reduce the computational cost of the network.
- DenseNet: Densely Connected Convolutional Networks [177]. DenseNet is a convolutional neural network architecture that connects each layer to every other layer in a feed-forward fashion allowing feature reuse and a more efficient training.
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks [178]. EfficientNet is a convolutional neural network architecture that scales up CNNs with a compound scaling method that balances the accuracy and computational cost of the network.

These are just a few examples of many other backbones available in TensorFlow. These are the most popular and widely used ones in the computer vision field.

4.3.2. Attention Mechanisms

The previous chapter presented some attention mechanisms that were explored using convolutional layers. This section explores other mechanisms and some of the above but using linear layers.

Although attention models can be used with different types of layers, linear layers are generally used instead of convolutional layers. It is because convolutional layers are inherently sensitive to the position and locality of features in an input, which makes it more difficult for the model to attend to specific parts of the input. Linear layers achieve a total connection between features independently of their position.

Instead, linear layers allow the model to assign weights to any input section, regardless of its position or locality. It makes it easier for the model to attend to specific input features using attention mechanisms. Positional encoding is employed to preserve location information. This module will be explained later in the model section.

4.3.2.1. Self-Attention

Self-Attention was previously explored in Chapter 3, using convolutional layers. Here, it will be explored with linear layers, and in another of its variants, the scaled-dot product Self-Attention [70]. The attention weights a_i is defined in Equation 4.3 to the input feature vector x_i, which is graphically described in Figure 4.2.

$$a_i = \sigma\left(\frac{Q(x_i)K(x_i)^T}{\sqrt{d_k}}\right) \tag{4.3}$$

Where σ is the softmax function; Q is the query; K is the key; and d_k is the scaling factor. With that said, the attended output z is computed as (also described in Figure 4.2):

$$z = a_i V(x_i) \tag{4.4}$$

It is an attention mechanism where the dot products are scaled down by $\sqrt{d_k}$. If we assume that Q and K are d_k-dimensional vectors whose components are independent random variables with mean 0 and variance 1, then their dot product, $Q \cdot K = \sum_{i=1}^{d_k} u_i v_i$, has mean 0 and variance d_k. Since we would prefer these values to have variance 1, we divide by $\sqrt{d_k}$.

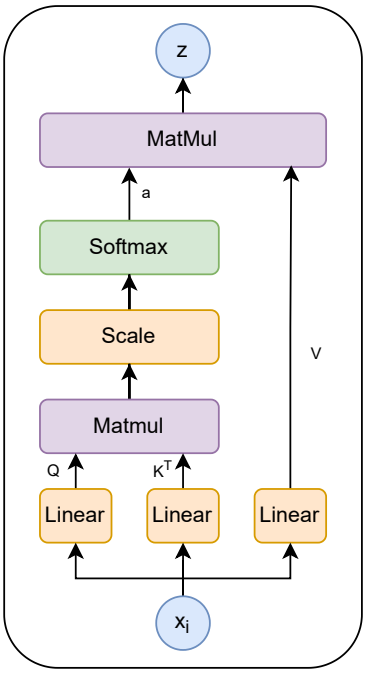


Figure 4.2: Self-Attention procedure.

4.3.2.2. Cross-Attention

Cross-Attention (Figure 4.3), which combines asymmetrically two separate embedding sequences of same dimension. Generally, in Cross-Attention, the key (K) and the value (V) are computed from one input and the query (Q) from another. A well-known example of this type of attention is Transformer [70], which implements this technique in the Decoder section. In this case, the attention weight a_i for the input feature vectors x_i and y_i is computed as follows:

$$a_i = \sigma\left(\frac{Q(y_i)K(x_i)^T}{\sqrt{d_k}}\right) \tag{4.5}$$

Where σ is the softmax function; Q is the query; K is the key; and d_k is the scaling factor. Note that in this method, the query is calculated from a source other than the key and value. The attended output z is computed as:

$$z = a_i V(x_i) \tag{4.6}$$

4.3.2.3. Multi-Head Attention

Multi-Head Attention was previously explored in Chapter 3, which implemented convolutional Self-Attention modules in parallel. It allows parallel computing and provides with the ability to pay attention to different aspects in the same feed.

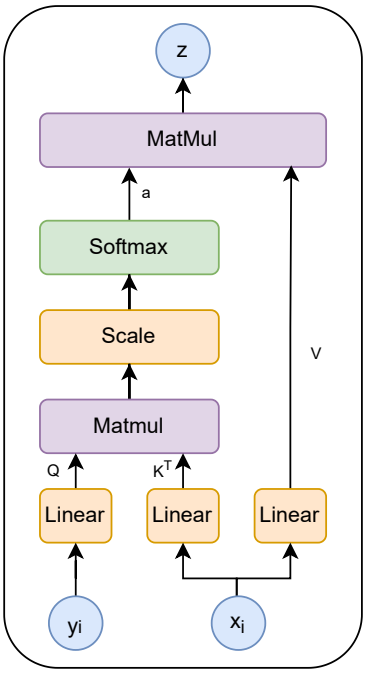


Figure 4.3: Cross-Attention procedure.

As explained before, it is a powerful architecture for understanding features and their relationships. This Chapter explores it in two variants: build with Self-Attention modules and Cross-Attention modules. Figure 4.4a shows this method applied with Self-Attention, following the Equation 4.7. It can also be configured with Cross-Attention, as could be seen in Figure 4.4b. In this case, the attention weight a_i for each input element x_i is computed as follows:

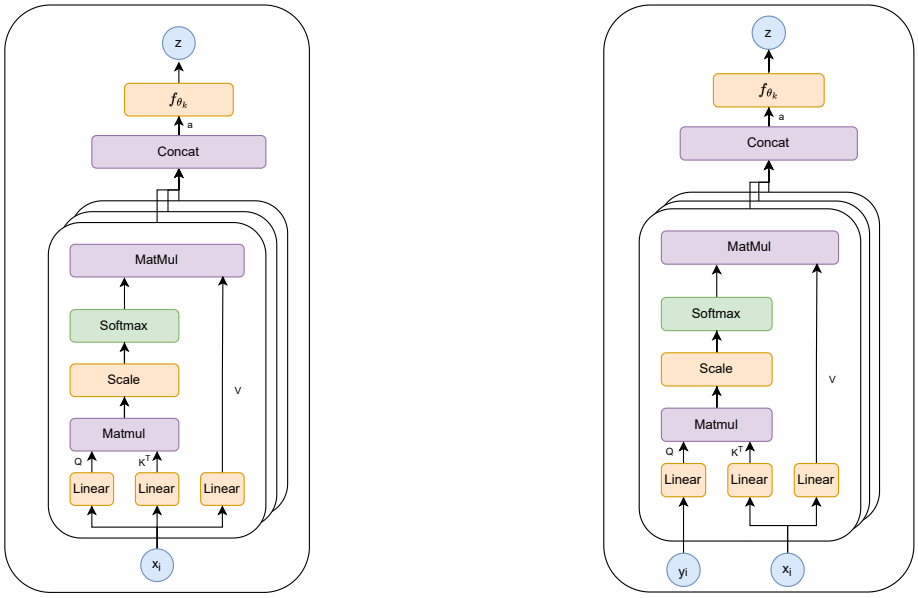
$$a_i = Concat(head_1, head_2, \ldots, head_M) \quad where \quad head_i = \sigma\left(\frac{Q(x_i)K(x_i)^T}{\sqrt{d_k}}\right)V(x_i) \quad (4.7)$$

Where M is the number of heads, and z is computed by a feed-forward neural network with parameters θ_k as:

$$z = f_{\theta_k}(a_i) \quad (4.8)$$

4.3.2.4. Transformer Encoder

This section explains the Transformer Encoder (Figure 4.5), which is one of the components of the complete Transformer (Encoder and Decoder) presented by [70]. The Transformer Encoder is a neural network architecture that uses attention mechanisms to process sequential input data. The Transformer Encoder was introduced as a replacement to Recurrent Neural Networks (RNNs) in tasks such as machine translation, language modelling and other natural language processing tasks.



(a) Multi-Head Self-Attention procedure.

(b) Multi-Head Cross-Attention procedure.

Figure 4.4: Multi-Head Attention mechanism

The Transformer Encoder consists of a stack of layers, each with two sub-layers: a Multi-Head Self-Attention mechanism and a feed-forward neural network. The Multi-Head Self-Attention mechanism allows the model to attend to different parts of the input and to compute a weighted sum of the input elements, where the weights are calculated based on the similarity between the input elements. The feed-forward neural network then processes the output of the Multi-Head Self-Attention mechanism to produce the final attended feature vector.

The Transformer Encoder also has an additional mechanism called the positional encoding, which allows the model to take into account the order of the input elements. In this way, the model can understand the spatial context of the input without fading this information through its layers.

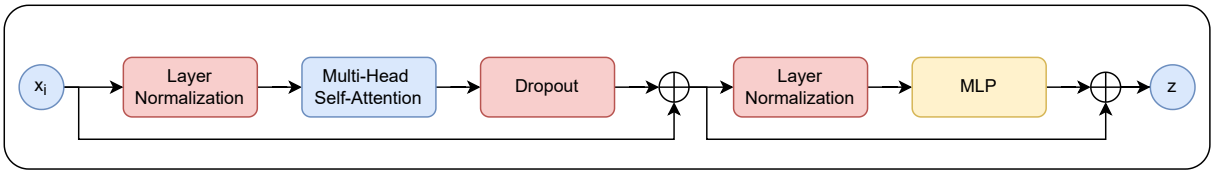
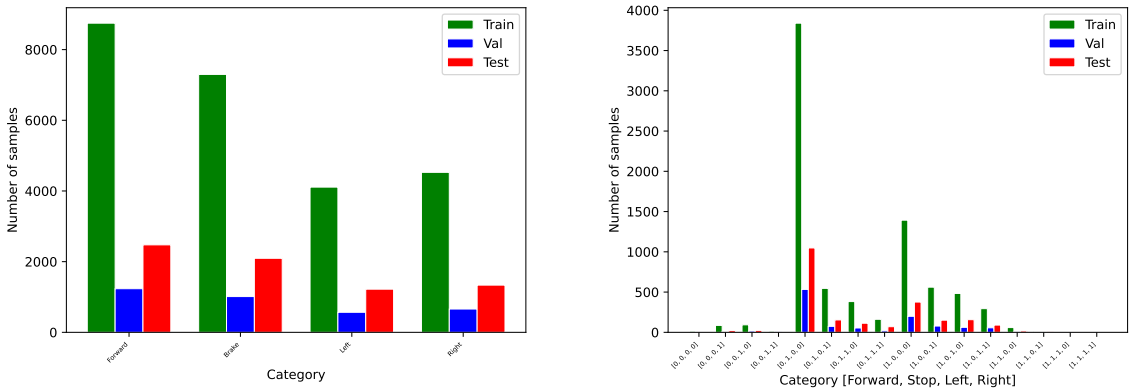


Figure 4.5: Transformer Encoder procedure.

The Transformer Encoder has shown to be very effective in NLP tasks, outperforming traditional RNN-based models. The main advantage of the Transformer Encoder is its ability to handle long-term dependencies in the input data, which is particularly useful in tasks such as machine translation and language modelling. Additionally, the Transformer Encoder is also able to parallelize the computations, which makes it more efficient than traditional RNN-based models. The problem is that it takes many GPUs to calculate these relationships from raw data.



(a) Dataset classification per categories treated as multi-label classification.

(b) Dataset classification per categories treated as categorical classification.

Figure 4.6: Dataset analysis.

Concluding this section, attention mechanisms have been shown to be very useful in many deep learning tasks, such as language modelling, image captioning, and machine translation. They enable the model to focus on the most relevant parts of the input, which improves the model's performance and reduces the amount of computation required for the final sections of the model.

Keep in mind that these are just a few examples, and there are other types of attention mechanisms. Also, attention mechanisms are still an active area of research, and new attention mechanisms are being proposed regularly. It is also worth noting that different types of attention mechanisms may be more appropriate for distinct tasks. A specific implementation of an attention mechanism may vary depending on it and the available data. Additionally, it is possible to combine different types of attention mechanisms in a single model to take advantage of the strengths of each mechanism.

4.4. Dataset BDD-OIA

From the above exposed datasets, BDD-OIA [76] has been chosen as it provides decision-making annotation and their explanation for complex sequences in urban environments. This dataset has 22,924 5-second video clips annotated for 4 actions (Move forward, Stop/Slow down, Turn left and Turn right) and 21 explanations. The classification of these categories was exposed before in Table 4.2.

In this section, the dataset is analysed to get statistics. The multi-label classification problem is exposed. Finally, some of the issues encountered in the dataset are discussed.

Figure 4.6 shows the dataset analysed in two ways for the action label. Firstly, as was done by the original authors, Figure 4.6a shows the dataset split into four categories, one for each possible action label (Forward, Stop/Slow down, Turn left and Turn right). Moreover, the dataset was labelled as a multi-label classification problem. In this regard,

Figure 4.6b shows the dataset split into 16 categories as. It corresponds to the 4 possible actions, which can be True or False, resulting in $2^4 = 16$ different categories.

Multi-label classification is a machine learning task where an instance/sample can be assigned to multiple labels. It is an extension of the traditional single-label classification, where an instance can only be assigned to one class.

The goal of multi-label classification is to predict the most relevant labels for a given input sample. It is commonly used in tasks such as image classification, where an image can have multiple objects or features that have their own label. Other examples include text classification, where a text document can have various topics, or music genre classification, where a song can belong to multiple genres.

There are different approaches to solve multi-label classification problems. It can be handled as a multiple-binary classification problem, where each label is considered a separate binary class. This approach is known as Binary Relevance. Another approach is to use a multi-output classifier to predict all the labels simultaneously. Another strategy is to utilise neural networks with a specific architecture called multi-output neural network, which handles multi-label classification tasks. This architecture can include multiple output layers or multiple heads in a single output layer to deal with the multi-label prediction. Another technique is to use several times fewer representative instances of the dataset in order to give more weight to the final model.

Backing to the subject under analysis, the dataset split into 4 categories is more or less balanced, with a little biased in the Turn options (Left and Right). Whereas, with the multi-label classification, the dataset is unbalanced. The most representative class is $[0, 1, 0, 0]$, which is "Break". The reason for this behaviour is that the dataset is extracted from BDD100K [165], which collected 100K driving videos (40 seconds each) from many cities and regions in the populated areas of the US. In cities, the most common manoeuvre is to break due to traffic jams and crowded environments. This conclusion makes the dataset challenging for label distribution, but the scenes are complex due to the urban scenario, which increases the diversity of the dataset. In opposition, a dataset mainly conducted on highways will cause the opposite behaviour $[1, 0, 0, 0]$, where the most representative class would be "Forward" but with a less complex scenario.

This dataset consists of videos that have to be sampled in order to train the model. The sampling rate of the videos is 10 Hz because it is a standard frequency used in the literature for perception systems [179]. This strategy produces enough data to build a model that can be processed at that frame rate. This contribution is different from what has been done in the state-of-the-art because they used only the last frame, which misses the temporal information. However, it was found that the number of videos does not match with the number of last frames used by other works, so the comparison is not exact with the other proposals of the state of the art.

4.5. Driver's Attention-based Explainable Decision-Making Model

The following section explains the main contribution of this chapter, the Driver attention-based explainable decision-making model built to assist the driver or to help the learning of autonomous vehicles. It is a deep learning model, which predicts the decision-making of the ego-vehicle, giving an explanation based on the attention module previously explained in this dissertation.

The input data of this model is an image sequence extracted from the BDD-OIA dataset [76]. The model predicts an action expressed as a four-dimension vector and an explanation as a 21-dimension vector. This section will expose the model architecture, which main modules have been previously explained.

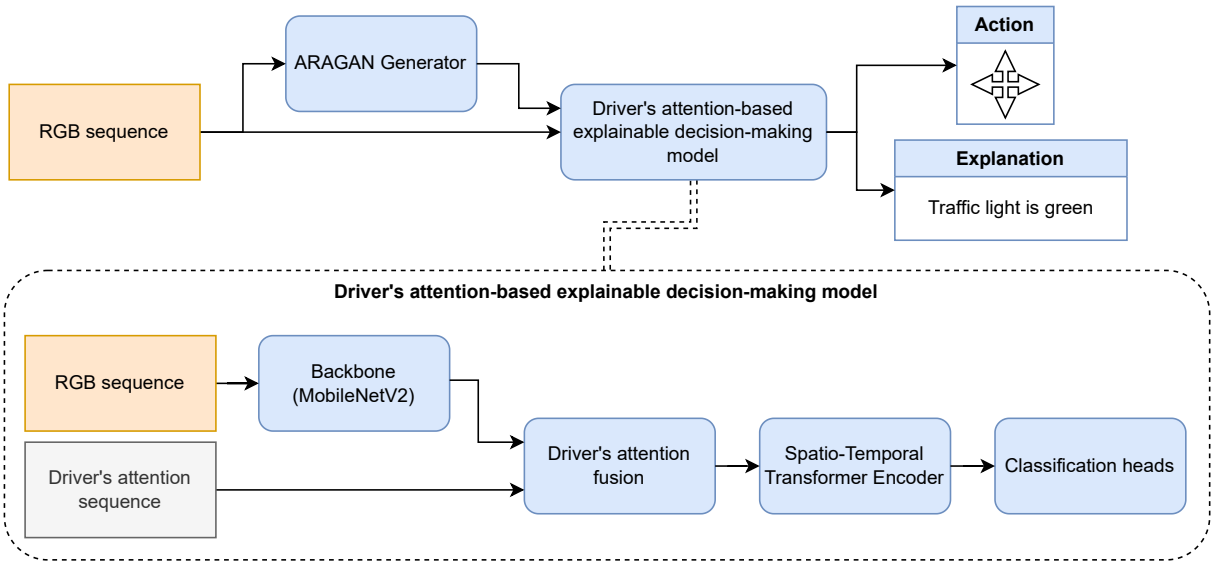


Figure 4.7: Driver attention-based explainable decision-making.

Figure 4.7 shows the architecture and modules of this proposal. These modules will be explained individually to understand the idea behind this proposal and the data flow in the model. In contrast to what is done in the literature [76], [170], the input model is a video sequence from the dataset instead of the last frame. It makes the model heavier, but with efficient modules, this proposal outperforms other architectures in terms of efficiency.

The model is composed of a Feature extractor or backbone; driver's attention fusion, which is based on the ARAGAN Generator; a Spatio-Temporal Transformer Encoder; and the classification heads.

4.5.1. Feature Extractor

As previously explained, one of the most used techniques in deep learning is the utilization of backbones formerly trained in other datasets to be used in other applications.

In this proposal, MobileNetV2 is employed [171], which is a mobile architecture that does not increase the computational cost of the final model. It is a commonly used computer vision backbone that is composed of convolutional layers and residual bottleneck layers. The version used has 2,261,632 parameters that are frozen because it has been trained in Imagenet, a large image dataset.

Using a backbone as an initial feature extractor makes it possible to work with this kind of dataset that has not got a great amount of data.

4.5.2. Driver's Attention Fusion based on ARAGAN

Besides the RGB branch, this proposal uses the driver's attention for an explainable decision-making model, which is the main idea of this chapter. For this purpose, the ARAGAN Generator, explained in previous chapters, is employed. It has been trained in the BDD-A dataset.

This work has not got pre-processing, and the feed-forward of the driver attention net is done simultaneously with the final proposal. After the feed-forward that is done in the forward pass of the complete proposal, there is no pre-processing in this proposal. The generated attention map is resized to the output of the feature extractor (7×7). After that, the sequence attention map is passed through convolutional layers, one for each clip image. The kernel size is (7×7) to have the complete resized attention map into account. The result of this layer is multiplied by the output of the backbone and concatenated to it. With this method, the driver's attention is used to understand the output characteristics of the backbone. Moreover, the complete information extracted by the backbone is passed to the following network steps. Finally, a ReLU activation function is employed.

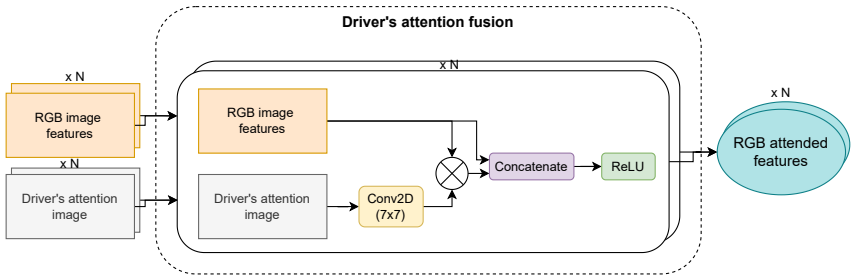


Figure 4.8: Driver's attention fusion based on ARAGAN.

Figure 4.8 shows the module parts. In the complete architecture, this module is repeated N times in parallel, one for each clip image, as this proposal works with video sequences. They do not share weights between different timestamps.

4.5.3. Spatio-Temporal Transformer Encoder

After the driver's attention module, the following section is in charge of the Spatio-Temporal Transformer Encoder. It understands the relationships between the features from the previous modules and filters them before passing them to the last network step.

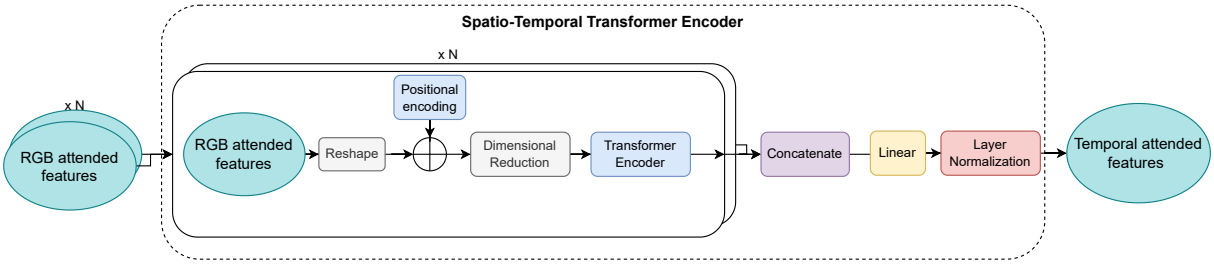


Figure 4.9: Spatio-Temporal Transformer Encoder.

Figure 4.9 shows the proposed module, which is composed of N modules that compute in parallel for each sequence frame. Each parallelised module has as input the concatenation of the multiplied attention features and the backbone features, which are flattened using a reshaping method. After that, positional encoding is employed to preserve the spatial information throughout the module, which follows Equation 4.9. The positional encoding is a vector that is added to the features to indicate its position in the sequence (feature map). The underlying idea is that it generates a unique coding for each location based on a combination of sine and cosine waves with different frequencies.

$$\vec{p_t} = f(t) := \begin{cases} PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \\ PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \end{cases} \tag{4.9}$$

Figure 4.10 shows a graphical example of positional encoding. The represented matrix has been built with the Equation 4.9, with $K = 49$ and $d = 100$. The values used in the proposed model are $K = 49$ and $d = 2560$.

After the positional encoding, an intermediate layer is necessary to reduce computational costs. For this reason, a dimensional reduction is performed. The Transformer encoder has a high memory consumption if the input tensor is too big. The reduction is carried out linearly, with a Linear layer of 32 units.

With the positional encoding and the dimensional reduction explained, a Transformer Encoder is applied, which encodes the Spatio-Temporal features of the sequence. It has been built with N Encoders for each frame, which are composed following the Figure 4.11. The encoder is composed of a stack of $M = 8$ identical layers.

The Spatio-Temporal Encoder is based on the original work [70]. It has a Layer Normalization, followed by Multi-Head Self-Attention, which has been previously explained, and a Dropout layer. This branch has a residual connection; a shortcut

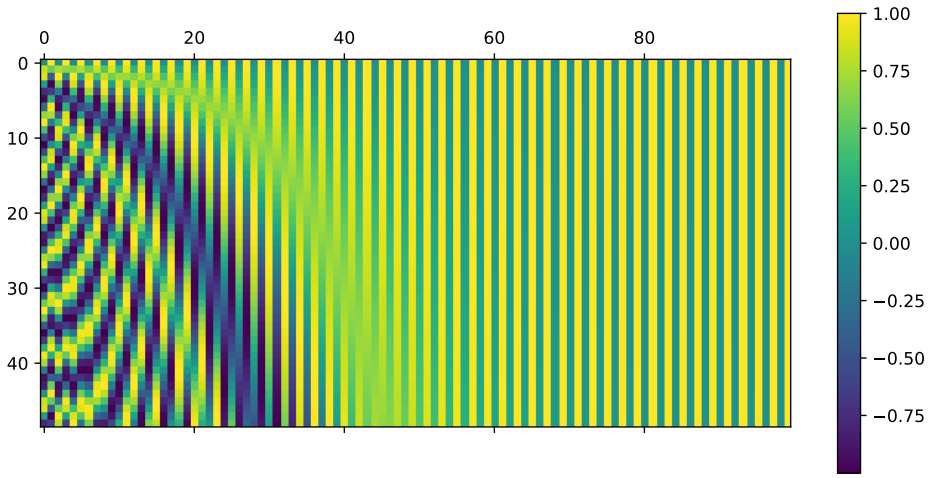


Figure 4.10: Example of positional coding for 49 positions and 100 channels. It is for display only because the presented model has 2560 channels.

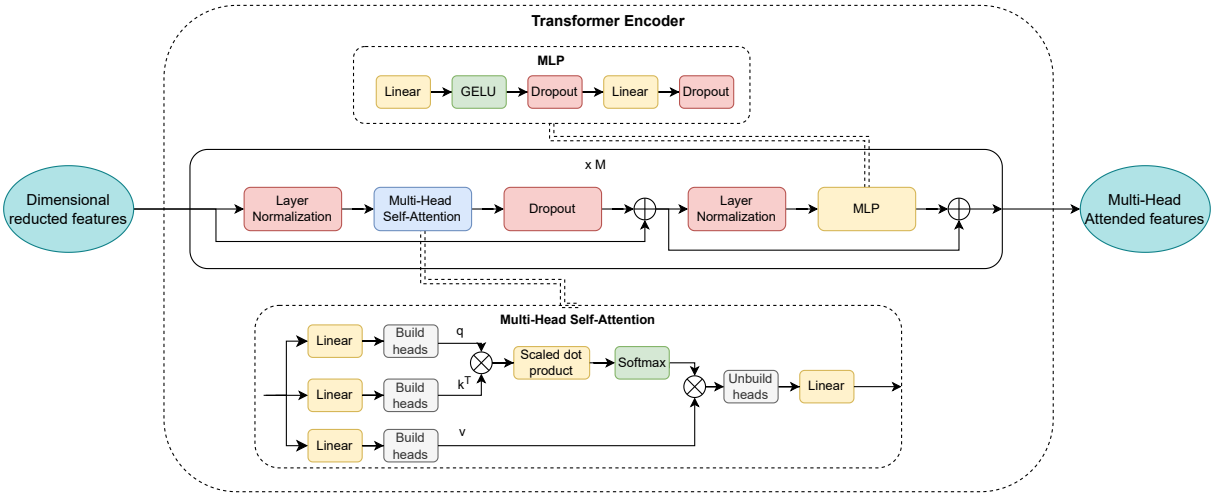


Figure 4.11: Transformer Encoder

connection that allows the gradient to flow directly from one layer to another, bypassing one or more intermediate layers. After that, another Layer Normalization is employed, followed by a Multi Layer Perceptron module (MLP) to end with another residual connection.

Figure 4.11 represents the Multi-Head Self-Attention employed and the architecture of this module. The theory inside the module has been explained earlier, and the only difference here is how the heads are parallelised. They are calculated in a single forward pass instead one by one. For this purpose, the outputs of the Linear layers are reshaped to have Multi-Head computing. In addition, Figure 4.11 shows the final MLP, which is composed of a Linear layer, a GELU activation function (Equation 4.10), a Dropout layer, another Linear layer, and a final Dropout.

$$GELU(x) = x \cdot \Phi(x) \approx 0.5 \cdot x \cdot (1 + tanh(\sqrt{2/\pi} \cdot (x + 0.044715x^3))) \tag{4.10}$$

After that, the features are concatenated and pass through a Linear layer and Layer Normalization to output the temporal attended features. The Spatio-Temporal Transformer encoder proposed to encode the spatio-temporal features is concluded. It allows parallel computing efficiently. Moreover, the features fed to this module are filtered by the driver's attention module. The image features are extracted using a pre-trained backbone, which allows using a low-dimensional encoder.

4.5.4. Classification Heads

The last modules are the classification heads composed of two branches, one for each output (Action, Explanation). They are represented in Figure 4.12, showing the module connections. The first branch is in charge of the Action output. It is a standard classification module built of a Linear layer, followed by Batch Normalization, a ReLU activation function, Dropout, another Linear layer and a sigmoid activation function to end the branch. The output is a 4-dimensional vector. The last activation function is sigmoid because the problem to solve is a multi-label classification problem, where all classes can be chosen or not.

The second branch, which predicts the Explanation layer, is quite different because the input of this branch is the latent vector that comes from the Spatio-Temporal Transformer Encoder and the action output. Multi-Head Cross-Attention is the first module that models the relationship between the action output and the latent vector from the previous module. The reason for this procedure is that the action conditions the explanation. As seen in Table 4.2, each action has its own explanations.

After the Multi-Head Cross-Attention there is a classification section that follows the same structure as the Action branch, but with the difference that the output is a 21-dimensional vector.

4.6. Experimental Setup

The following section explains the experimental setup of this proposal, with the different loss functions and metrics used to evaluate the proposal.

4.6.1. Loss Functions

A loss function in deep learning is a function that measures the difference or distance between the predicted output and the actual output. The objective of training a neural

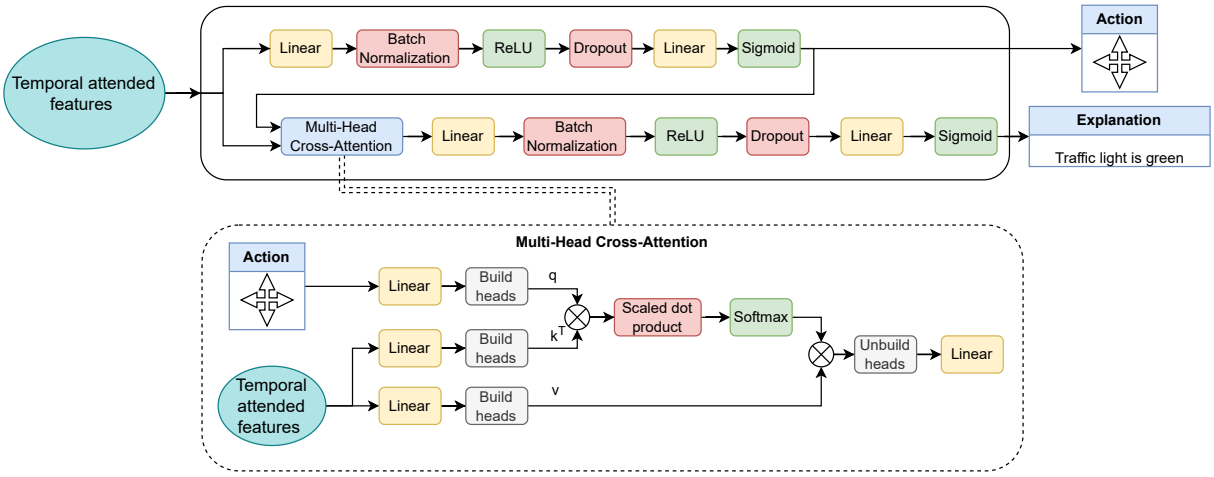


Figure 4.12: Classification heads.

network is to minimise the value of the loss function. Common examples of loss functions include mean squared error and cross-entropy. These loss functions are used to evaluate the model performance during training to optimise the model's parameters.

The task under study is a multi-label classification composed of two outputs. One for the action and another one for the explanation. In this work, the aim is to predict multiple labels or classes for each input sample. Some common loss functions for multi-label classification include:

- Binary Cross-Entropy (BCE) loss (Equation 4.11). It is the most commonly used loss function for multi-label classification. It is used for binary classification problems and can be extended to multi-label ones by independently computing the loss for each label.

$$\mathcal{L}_{BCE}(y, x) = \frac{1}{N}\sum_{i=0}^{N}[y \cdot log(x) + (1 - y) \cdot log(1 - x)] \tag{4.11}$$

- Sigmoid Cross-Entropy loss (Equation 4.12). It is similar to binary cross-entropy loss but uses the sigmoid activation function instead of the softmax activation function. It is used when the model needs to predict the probability of each label.

$$\mathcal{L}_{\sigma CE}(y, x) = \frac{1}{N}\sum_{i=0}^{N}[y \cdot \sigma(x) + (1 - y) \cdot log(1 - \sigma(x))] \tag{4.12}$$

- Categorical Cross-Entropy loss (Equation 4.13). It is used for multi-class classification problems and can be extended to multi-label classification by independently computing the loss for each label.

$$\mathcal{L}_{CCE}(y, x) = \frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(x_i) \tag{4.13}$$

- Hamming loss (Equation 4.14). It is a well-known loss function for multi-label classification. It is useful when the number of them is large, as it is more robust to class imbalance.

$$\mathcal{L}_H(y, x) = \frac{1}{N}\sum_{i=0}^{N} y_i \oplus x_i \text{ ; where } \oplus \text{ denotes exclusive-or} \tag{4.14}$$

- F1-score loss (Equation 4.15). It is used when the class imbalance is large. It optimises the F1-score of the model. The problem with the F1 score is that it is not differentiable, but it can be modified to make it differentiable. It can be computed as a continuous sum of likelihood values using probabilities without applying any threshold.

$$\mathcal{L}_{F1}(y, x) = \frac{1}{N}\sum_{i=0}^{N} \frac{2 \cdot x \cdot y}{2 \cdot x \cdot y + x \cdot (1-y) + (1-x) \cdot y} \tag{4.15}$$

Some architectures in the state-of-the-art [76], [170] used the BCE as their loss function. This work proposes to use the F1-score loss to find the optimal threshold that maximises this metric. Moreover, due to the unbalanced dataset, the loss is weighted for the action labels (Turn Left and Turn Right).

4.6.2. Metrics

Apart from the loss function that evaluates the error during the training, which will be backpropagated, another function is needed to calculate the error after the training. This function is called metric and allows developers to measure the performance of their models and compare them to other proposals. Metrics in deep learning are used to evaluate the performance of a trained model. They provide a way to quantitatively assess how well the model makes predictions on unseen data. In other words, metrics are utilised to measure the model's accuracy, robustness, and generalisation capabilities.

Many metrics can be used in deep learning, depending on the specific task and the type of data. The most commonly used metrics are accuracy, precision, recall, F1-score, area under the ROC curve (AUC), etc. Each metric has its own strengths and weaknesses, and choosing the appropriate metric for the specific task is significant.

In general, it is crucial to consider both the overall performance of the model as well as its performance on certain subsets of the data when evaluating the performance of a deep learning model.

- Accuracy (Equation 4.16). It is the most common metric and represents the proportion of correct predictions made by the model. It can be calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.16}$$

where, TP are the true positives; TN are the true negatives; FP are the false positives and FN are the false negatives.

- Precision (Equation 4.17). It measures the proportion of true positive predictions among all positive predictions made by the model. It can be calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{4.17}$$

- Recall (Equation 4.18). It measures the proportion of true positive predictions among all actual positive instances in the data. It can be calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{4.18}$$

- F1-score (Equation 4.19). It is the harmonic mean of precision and recall. It is commonly used to balance the trade-off between precision and recall. It can be calculated as:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.19}$$

Equation 4.19 shows how the F1-score is calculated. This proposal evaluates the models with two variations of this metric. Firstly the models are tested using Equation 4.20, which computes the F1-score for all the predictions. It averages the F1-score over all the forecastings.

$$F1_{all} = \frac{1}{|A|} \sum_{j=1}^{|A|} F1(\hat{A}_j, A_j) \tag{4.20}$$

The other variation computes the mean F1-score of each label (action or explanation). It is represented in Equation 4.21.

$$mF1 = \frac{1}{|A|} \sum_{j=1}^{|A|} \frac{1}{|C|} \sum_{i=1}^{|C|} F1(\hat{A}_j^i, A_j^i) \tag{4.21}$$

Finally, the mean of both $F1_{all}$ (action and explanation) is computed to show the best model in both metrics. This results follows Equation 4.22

$$\overline{F1_{all}} = \frac{F1_{all}^{decision} + F1_{all}^{explanation}}{2} \tag{4.22}$$

These metrics will be used to compare the proposed modules and the comparison against other proposals from the literature.

4.7. Experimental Results

In this section, the results obtained from different experiments are exposed. The evaluation is conducted in two ways. Firstly, the evaluation is done as an ablation study to determine the best option and the contribution of each module previously explained. After this, a comparison with other state-of-the-art proposals will be exposed. Finally, a discussion about the results will be presented.

4.7.1. Ablation Study

Various modules have been explained above. For this reason, an ablation study has been carried out to measure the contribution of each module. This study will pursue the following sections:

- The use of a sequence instead of the last frame.
- The contribution of adding driver attention to the pipeline.
- The advantage of using the Transformer Encoder to understand the Spatio-Temporal relationships.
- The use of Cross-Attention between the action head and latent vector to obtain the explanation.
- Loss function, BCE vs F1-score loss.

The ablation study is exposed in Table 4.5. It shows the contributions of each module to the final model, which will be compared with alternative solutions, and the number of parameters of the model under study (Par. (M)). The evaluation is expressed in the metrics explained above. The F1-score is calculated individually for each action, as well as the mean for all of them ($mF1$). After that, the F1-score for all the decisions is computed ($F1_{all}$). Finally, the mean F1-score ($mF1$) for each explanation and the F1-score ($F1_{all}$) for all of these predictions are calculated. The last exposed metric is the inference time of each proposal (Inf.).

This study has been carried out with the same experimental setup for every model. Moreover, the seed has been set to a value to avoid randomness and not reproducibility between experiments.

Every model has been trained with a batch size of 512, an Adam optimizer ($b_1 = 0.9$, $b_2 = 0.999$ and $\epsilon = 1e-7$) and the learning rate has been set to 0.001 with an Exponential Decay scheduler. The GPU used for training is an NVIDIA A100 with 80 GB of VRAM. Moreover, the inference time has been measured with an NVIDIA 2080 Ti with 11 GB of VRAM, which is more common to have by other researchers in order to compare with their proposals. The training has been configured with an early stopping of 3 epochs to

avoid overfitting. It will stop after 3 epochs without improving the metrics in the test set. Regarding the dataset, all the models have been trained with the training and validation set shuffled, and the exposed results correspond to the testing set. Denote that the dataset has not got the same amount of sequences as the authors claimed in the paper, where the dataset was presented [76]. The dataset is composed of 7,946 videos for training, 1,117 for validation and 2,236 for testing, instead of a training set of 16,082 images, a validation set of 2,270 and a test set of 4,572. The inference time results have been measured to the complete forward pass of the model, including the driver attention model. The batch size has been set to 1 to measure this parameter.

The strategy followed in this ablation study consists of using the modules explained above, understanding the contribution of each module to the final proposal and the benefits of using them together. The first model is composed of a feature extractor with classification heads (Id. 4.5.1). This configuration has been trained using only the last frame of the sequence instead of the video clip.

After that, the following approach adds an input sequence to the model (Id. 4.5.2). This configuration computes the temporal relationships along the series and feeds it to the final decision and explanation. The last ten frames of the sequence have been used but skipping pairs, which means they correspond to the two last seconds of the video clip. The problem with this approach is that the high quantity of parameters makes the model overfit in the early steps of the training without being able to generalize. That's why the results are worst, but this information should help the final task because it is crucial to understand the temporal relationships to predict the vehicle decision.

The next model adds the driver attention module based on ARAGAN (Id. 4.5.3). It contributes to the model with the comprehended visual scene. The driver's attention is employed, which is generated by ARAGAN, to achieve this visual understanding. This proposal again has a problem of over-adjustment due to the number of parameters to be trained.

The Spatio-Temporal Encoder module has been employed to overcome this situation (Id. 4.5.4). It achieves similar performance with 43 times less parameters. It contributes with a low-parameter model that understands all the relationships among the temporal and spatial information. Having a model with fewer parameters does not imply that it will infer faster because the architecture is the one that affects this the most. Moreover, it can be used in embedded computers such as Nvidia Jetson because it fits into these systems. However, the complexity of the architecture proposal makes it slower than the other techniques, and it will require some optimization methods that have not been addressed in this work.

Our final proposal adds the Multi-Head Cross-Attention to the explanation classification head in order to model the relationships between the action and the explanation (Id. 4.5.7). It is the final model of this work.

Moreover, evaluating the cross relationships between modules is important. For this reason, two more evaluations are added. The first one assesses the final model without the driver's attention module showing lower performance explaining the benefits of this module (Id. 4.5.5). And the second one shows the final model but was trained with the last frame of the sequence to prove the advantage of understanding the temporal information (Id. 4.5.6).

Finally, the loss function has been changed to the best model (Id. 4.5.7), which uses the F1-score loss, to show its contribution compared with using the BCE loss (Id. 4.5.8). The proposed loss outperforms the other one.

With that proposal, the ablation study is concluded, and the comparison with other state-of-the-art models can begin with the driver's attention-based explainable decision-making (Id. 4.5.7).

Table 4.5: Ablation study of modules carried out to build the driver's attention-based explainable decision-making. Every model has been trained in the same data, and the results exposed are for the testing set of BDD-OIA.

Id.	Modules						$\mathcal{L}$	Par. (M)	Action					Explanation		Inf. (ms)
	MN V2	Seq	Att	ST Enc	CH	CA			F	S	L	R	mF1	mF1	$F1_{all}$	
4.5.1	✓				✓		F1	18.5	0.61	0.77	0.46	0.48	0.58	0.27	0.48	**5.5**
4.5.2	✓	✓			✓		F1	165	0.56	0.79	0.37	0.40	0.54	0.28	0.53	34.7
4.5.3	✓	✓	✓		✓		F1	328	0.59	0.79	0.37	0.39	0.54	0.28	0.53	157
4.5.4	✓	✓	✓	✓	✓		F1	7.6	0.61	0.80	0.45	0.44	0.58	0.28	0.53	195
4.5.5	✓	✓		✓		✓	F1	3.8	0.60	0.79	0.43	0.43	0.56	0.25	0.52	163
4.5.6	✓		✓	✓		✓	F1	**3.0**	0.60	0.78	0.48	0.48	0.59	0.26	0.49	21
4.5.7	✓	✓	✓	✓		✓	F1	4.2	**0.63**	**0.80**	**0.48**	**0.46**	**0.59**	**0.69**	0.27	**0.54**	199
4.5.8	✓	✓	✓	✓		✓	BCE	4.2	0.52	0.78	0.34	0.29	0.49	0.63	0.14	0.38	199

Notes: Id. = identification; MNV2 = MobiliNet V2; Seq = Use of sequences; Att = Driver attention module; ST Enc = Spatio-Temporal Transformer Encoder; CH = Classification heads; CA = Multi-Head Cross-Attention explanation head; $\mathcal{L}$ = Loss

4.7.2. Comparison with State-of-the-art Proposals

This section compares the driver's attention-based explainable decision-making proposal explained above with the state-of-the-art models that have published their results. Denote that [76] provides the code and the weights to its model, so it has been tested in the same experimental setup as this proposal. However, [170] do not provide code, and the published results could not be verified.

As we depicts in Table 4.6, our proposal gives the best $F1_{all}$ for the explanation. Regarding the action $F1_{all}$, it is only ten tenths behind [76], [168], [169] and two tenths behind [170]. The $\overline{F1_{all}}$ has been added to measure the average score of both results (action and explanations). About this metric, our approach achieves the best outcome for the tested models, and it is one-tenth behind [170].

Finally, the inference time has been compared, and both tested models provide the same values. They work at 5 Hz. However, [76] processes only the last frame and our driver's attention-based explainable decision-making processes ten sequence frames. The other proposals have not been tested because the code is not available.

Table 4.6: Action and explanation prediction performance using the driver's attention-based explainable decision-making. It will be compared with other proposals in the literature. Results in blue colour represent the best of the evaluated models. **Bold** results represent the best over all the proposals.

Models	Action						Explanation			Inf. (ms)
	F	S	L	R	mF1	$F1_{all}$	mF1	$F1_{all}$	$\overline{F1_{all}}$	
ResNet-101 [168]	0.755	0.607	0.098	0.108	0.392	0.601	0.180	0.331	0.466	-
local selector [169]	0.810	0.762	0.600	0.624	0.699	0.711	0.196	0.406	0.578	-
† FRCNN + BDD-OIA [76]	**0.829**	0.781	**0.630**	**0.634**	0.718	**0.734**	0.208	0.422	0.57	199
GSA (resnet) [170]	-	-	-	-	**0.750**	0.729	**0.644**	0.525	**0.627**	-
GSA (mobilenet) [170]	-	-	-	-	0.746	0.718	0.642	0.531	0.624	-
† Driver's attention-based explainable decision-making (ours)	0.634	**0.802**	0.475	0.460	0.593	0.695	0.267	**0.538**	0.617	**199**

† indicates that the proposal has been tested on the same computer and the results have been verified.

4.7.3. Discussion

Our model outperforms the other proposals in explanations obtaining the best $F1_{all}$ for this metric. Moreover, it gets the best $\overline{F1_{all}}$ of all the tested models in this experiment and the second-best performance of all the models in the literature. The inference time comparison denotes that having more parameters does not imply higher inference time and that the model architecture is the most crucial parameter to evaluate this metric. Nevertheless, the proposed model achieves a reasonable frame rate to make a high level decision, understanding the previous frames to take the best decision and explaining it based on how drivers look at the scene.

4.7.4. Qualitative Results

In this section, some examples of the test dataset evaluated in a qualitative way to show the strengths of our proposal. Table 4.7 explains the qualitative results of Figure 4.13 to understand their predictions and the misunderstanding of the model.

4.8. Chapter summary

This chapter proposes a driver's attention-based explainable decision-making trained on BDD-OIA. It presents a novel architecture that understands the temporal data to make a driving decision and explain it. The architecture is based on driver attention obtained with the features provided by the ARAGAN Generator and a Spatio-Temporal Transformer Encoder. The explanation branch has Multi-Head Cross-Attention to understand the relationship between both results.

The architecture is evaluated with other state-of-the-art architectures obtaining comparative results. In addition, an ablation study was carried out to show the best approach using the proposed modules. It outperforms the tested architectures, showing the best explanations for all the options in the literature.

Table 4.7: Qualitative results explained for Figure 4.13. Correct predictions Incorrect predictions

Figure	Action		Explanation		Comments
	Pre	GT	Pre	GT	
4.13a	Forward Stop Right	Forward Right	Follow traffic Obstacle: car Obstacles on the left lane	Follow traffic Traffic light is green Obstacles on the left lane	The model does not detect the traffic light and predicts to stop due to an obstacle.
4.13b	Stop	Stop	Obstacle: car The traffic light Solid line on the left	Obstacle: car Solid lane on the left	The model mispredicts the traffic light due to night conditions.
4.13c	Stop	Stop	The traffic light No lane on the left Solid line on the left Obstacles on the right lane	The traffic light No lane on the left Solid line on the left Obstacles on the right lane	Perfect prediction.
4.13d	Forward	Forward	Follow traffic No lane on the left Obstacles on the right lane	Follow traffic No lane on the left Obstacles on the right lane	Perfect prediction.
4.13e	Stop	Stop	Obstacle: car	Obstacle: car Obstacle: person/pedestrian	The model understands that the pedestrian on the right is not an obstacle and mispredicts it.
4.13f	Stop	Stop	Obstacle: person/pedestrian The traffic light Obstacle on the left lane	Obstacle: person/pedestrian The traffic light	The model understands the situation but it also predicts the bicycle on the left is an obstacle.
4.13g	Stop	Stop	Obstacle: car Solid line on the left Obstacles on the right lane	Obstacle: car Solid line on the left Obstacles on the right lane	Perfect prediction.
4.13h	Stop	Stop	Obstacle: car Obstacles on the left lane Obstacles on the right lane	Obstacle: car Obstacles on the left lane Obstacles on the right lane	Perfect prediction.
4.13i	Stop	Stop	Obstacle: car Solid line on the left Obstacles on the right lane	Obstacle: car No lane on the left Solid line on the left Obstacles on the right lane	The model mispredicts the left lane explanation, but it understands the other explanations for the situation.
4.13j	Stop	Stop	The traffic light	The traffic light	Perfect prediction.
4.13k	Stop	Stop	The traffic light Obstacles on the left lane	The traffic light	The model predicts that there is an obstacle on the left lane, which is not labelled.

The driver attention explains the visual context of a driving decision and ends the exploration of driver attention mechanism achieving a complete vehicle-to-driver interaction able to predict a decision-making, which can assist a human driver or can help the learning of an autonomous vehicle.

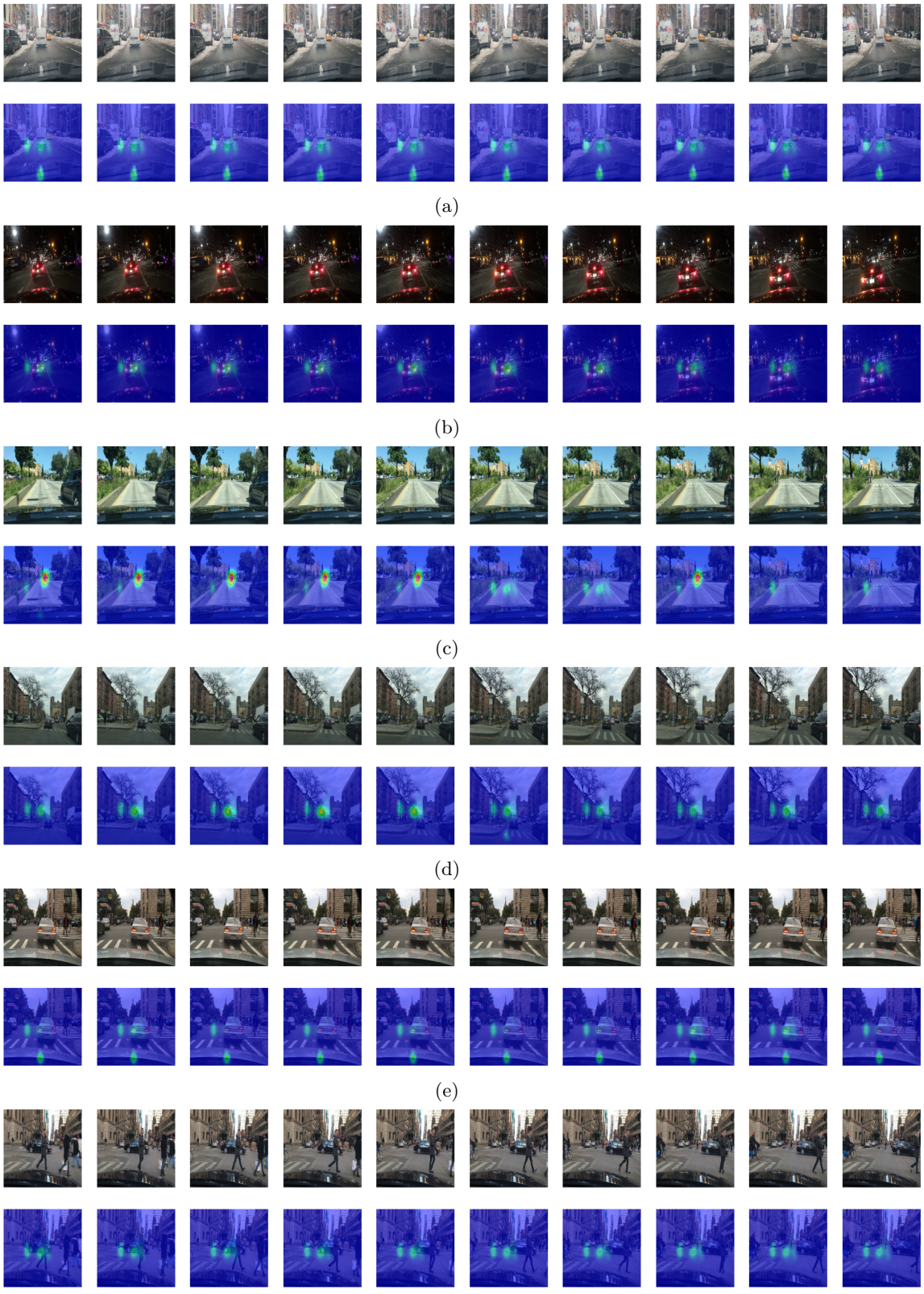
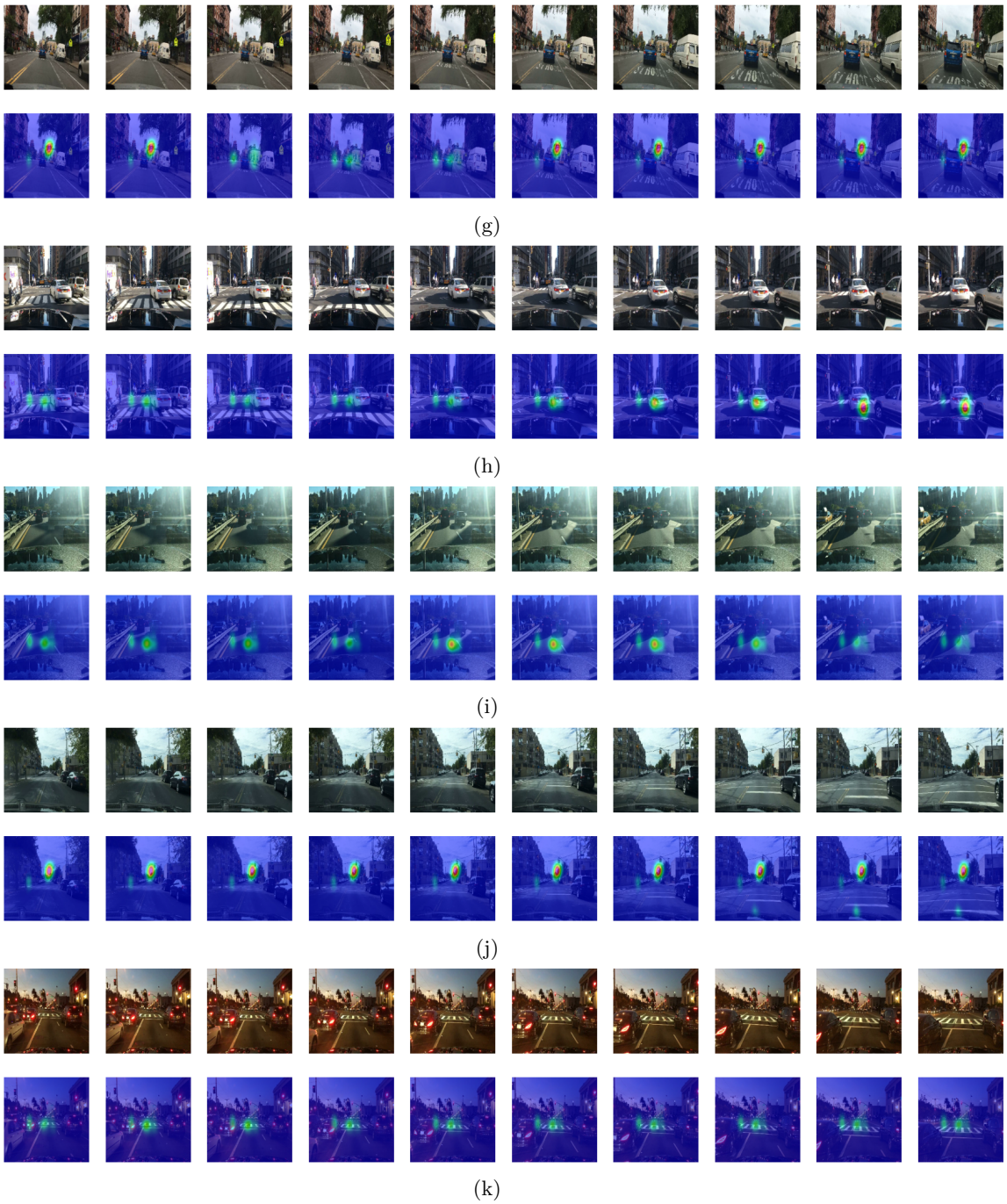


Figure 4.13: Qualitative results for the driver's attention explainable decision-making in BDD-OIA test set.



(g)

(h)

(i)

(j)

(k)

Figure 4.13: Qualitative results for the driver's attention explainable decision-making in BDD-OIA test set.

Chapter 5

V2D applications

¡Seguid en posición, hacedles frente! Hijos de Gondorn, y de Rohan, mis hermanos... Veo en vuestros ojos el mismo miedo que encogería mi propio corazón. Pudiera llegar el día en el que valor de los hombres decayera en que olvidáramos a nuestros compañeros y se rompieran los lazos de nuestra comunidad, pero hoy no es ese día. En que una hora de lobos y escudos rotos rubricaran la consumación de la edad de los hombres, pero hoy no es ese día. En este día lucharemos. Por todo aquello que vuestro corazón ama, de esta buena tierra, os llamo a luchar. ¡Hombres del Oeste!

Aragorn; El Señor de los Anillos: el retorno del Rey

This dissertation chapter presents two applications made in the CARLA simulator based in the models developed in the previous chapters. It consists of the following sections: a presentation of the simulator and its contributions, a driver take-over request application, which was partially published in the following journal Sensors paper [180]: "Driver take-over behaviour study based on gaze focalization and vehicle data in CARLA simulator", and an explainable vehicle's action prediction application.

5.1. Autonomous Driving Simulators

Testing autonomous vehicles on public roads is an expensive and time-consuming endeavour. In order to validate an Autonomous Driving System (ADS), it has to be tested in countless scenarios and environments, which would escalate the cost and development time exponentially in physical conditions. Considering this, using a photo-realistic simulation and an appropriate design of the evaluation metrics and driving scenarios are the two current keys to build safe and reliable ADS. Simulation as a form of accelerating the path to the market is not new to the AD industry. Tesla, Cruise, Waymo, Aurora, TuSimple and other companies have experimented the benefits

of using simulations made from real-world data to test their AV systems, particularly against made-up scenarios difficult to obtain in the real world.

Table 5.1 compares the main AD simulators, where some depicted features are: 2D/3D ground-truth, camera calibration, open-source, weather or traffic conditions.

For this work, some of the most crucial features are that the simulator is open-source and works with ROS, because all of the modules built in this dissertation use ROS, being the philosophy that has been conducted in the project.

With this in mind and the research group background, the CARLA simulator has been chosen, because it allows building scenarios specially designed to test the proposed applications using ROS compatibility.

Table 5.1: Autonomous driving simulators

Requirements	Gazebo	CarSim	CARLA	LGSVL
2D/3D GT	✗	✗	✓	✓
Camera calibrations	✗	✗	✓	✗
3D virtual environment	Indoor and outdoor	Outdoor-urban	Outdoor-urban	Outdoor-urban
Wrapper to other software	ROS	Matlab-Simulink	ROS, Autoware, Apollo	ROS, Autoware, Apollo
Open-source	✓	✗	✓	✓
Different weathers	✗	✗	✓	✓
Scalability via a server multi-client architecture	✗	✗	✓	✓
Traffic Scenario simulation: Support of different types of Dynamic objects	✗	✓	✓	✓

5.2. CARLA Simulator

CARLA simulator [181] is composed of an architecture based on server-client. The server computes the simulation: sensor rendering, physics computation, updating the world and actors, etc. The server must have GPU computing to achieve realistic results. On the other side is the client, which may be more than one in the same world. They control the actor's logic and the world's settings. It is done through the CARLA API (Python or C++), which mediates between the clients and the server. Figure 5.1 represents this communication between the world and the possible clients. CARLA has been built for flexibility and realism in rendering and physics simulation. It is implemented as an open-source layer over Unreal Engine 4 (UE4).

There are different modules that CARLA provides that are useful for autonomous driving development:

- **Traffic manager.** System in charge of the scene agents besides the ego-vehicle. It recreates urban driving behaviours to simulate real conditions.

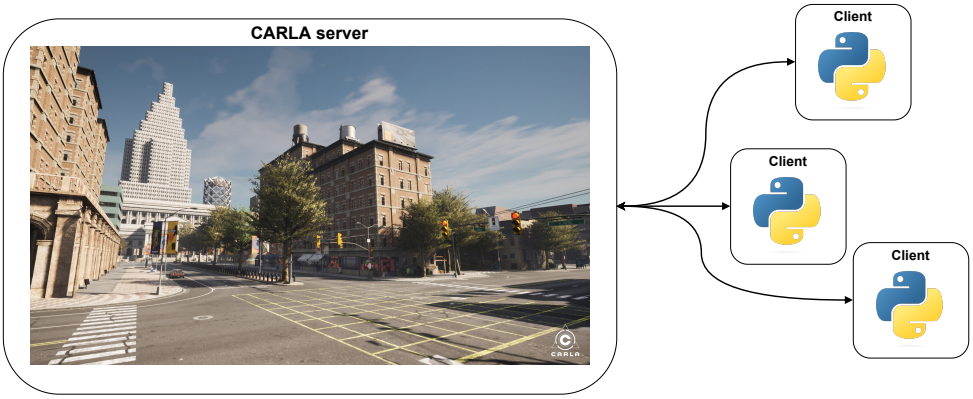


Figure 5.1: CARLA simulator communication between the world and clients.

- **Sensors.** They are used by vehicles to perceive their surroundings. Currently, there are mainly cameras, radars and LIDARs. They are in charge of the perception of the autonomous vehicle. In CARLA, there are several options available. About the camera sensor, there are different possibilities. The RGB camera provides an image of the surroundings being a sensor that can be easily added to a physical vehicle. The others sensor referred to the camera require processing to bring to a real car. These include: depth estimation, optical flow, semantic segmentation, instance segmentation and Dynamic Vision Sensor (DVS). Besides cameras, other sensors can be used in CARLA, such as GNSS, IMU, LIDAR, Radar, Responsibility Sensitive Safety (RSS) and Semantic LIDAR. As this work is focused on computer vision, we will not go further into these sensors. Moreover, there are other sensors that CARLA provides, which are valuable tools for development. Within these sensors, they are available: the collision sensor, which notifies if there is a collision between the ego-vehicle and other agents. The lane invasion sensor, which registers if the vehicle crosses a lane marking. And the obstacle sensor that registers an event every time the ego-vehicle has an obstacle ahead.

- **Recorder.** It allows recreating a simulation step by step to get statistical results. It is used for reproducibility purposes.

- **ROS bridge and Autoware implementation.** CARLA provides wrappers to ROS and Autoware to implement their architectures and communities.

- **Open assets.** CARLA provides different maps and vehicle blueprints for customization environments. The simulator is composed of nine cities that can be used freely for development and two more that are restricted to the CARLA Leaderboard[1], which is a competition to evaluate a self-driving proposal. Moreover, it allows customization and the creation of own maps.

[1] https://leaderboard.carla.org/

- **Scenario runner.** It allows the creation scenarios that reproduce use cases that a vehicle must face in simulation. The scenarios are defined using Python and the OpenSCENARIO standard.

5.2.1. Simulator Hardware

The simulator to test algorithms with users comprises three 24-inch screens (Full HD resolution 1920 x 1080p), a steering wheel and pedals (Logitech Driving Force GT) to control the vehicle and a ZED camera from Stereolabs for gaze tracking. The user under the test is placed in front of the simulator, as shown in Figure 5.2. It represents the driver's view during the experiment. The simulator set-up is similar to others used in the literature [182] and [183], which are composed of three screens, a steering wheel and pedals. The simulator is based on the CARLA simulator, which supports the development, training, and validation of autonomous driving systems in a hyper-realistic environment before transferring them to the real world. The simulator runs at 13 fps over a computer consisting of an Intel i7-9700k (overclocked to 4.9 GHz), NVIDIA 2080 Ti and 32 GB of RAM (3200 MHz).

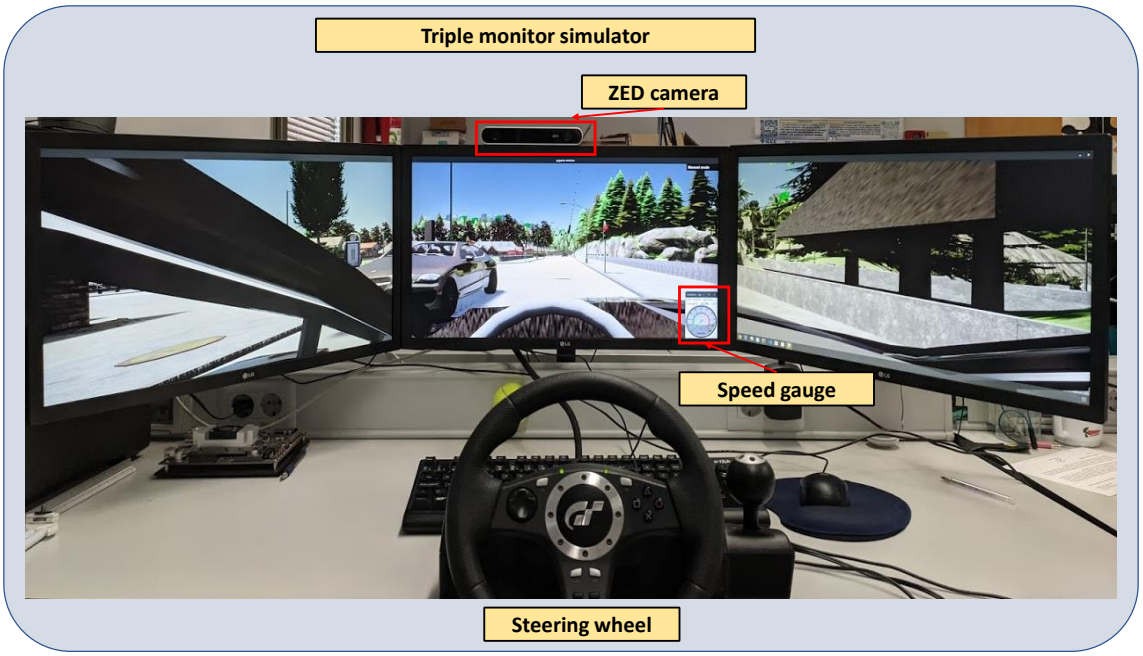


Figure 5.2: Point of view of the driver during the experiment.

5.3. Communication Interface through ROS

This section explains the communication between CARLA and the designed modules through ROS Noetic [105]

5.3.1. Driver Take-Over Request ROS Interface

The experiment to test the driver monitoring system to measure the DSA using the gaze focalization system designed uses ROS to communicate between modules. CARLA sensors publish data through the ROS bridge developed by CARLA creators. The sensors used in this experiment are: RGB camera and Semantic Segmentation, as well as some pseudo-sensors like Collision sensor and Lane Invasor sensor. Additionally, custom messages are utilised (e.g. vehicle transition status). Figure 5.3 shows the topic communication between modules.

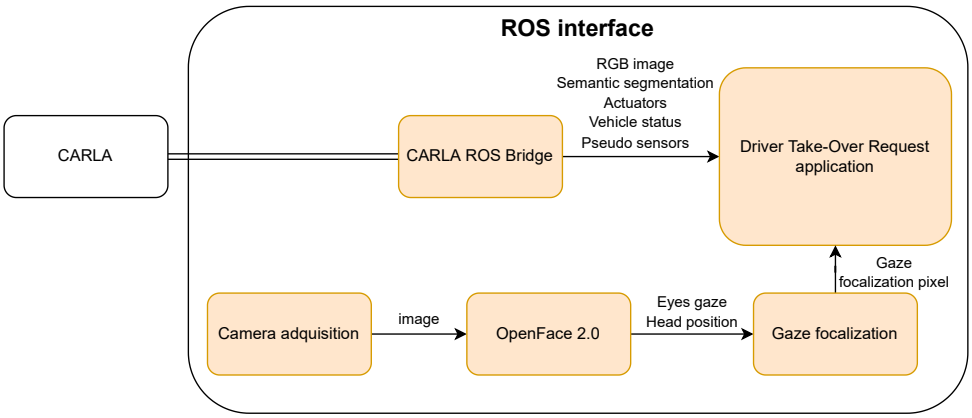


Figure 5.3: Driver Take-Over Request ROS interface.

Both image sensor resolutions (RGB and Semantic Segmentation) are set to 5760 x 1080p. This pixel density is the same as the simulator resolution, composed of three screens. These sensors publish images at this resolution to fit the simulator road scene watched from the driver position, allowing to match scene pixels with the driver's gaze vector. The vehicle camera's field of view is set to 100 degrees. This configuration provides the user with a view of the road in the simulator similar to that of a real vehicle. Figure 5.2 shows the drivers' view when they are sitting in the simulator. In that Figure, it can be observed that the camera is placed on top of the centre screen. This camera is in charge of the gaze focalization and will take a perfect view of the driver during the experiment.

5.3.2. Driver's Attention-based Explainable Decision-Making ROS Interface

As in the previous experiment, this one uses ROS to communicate between systems. CARLA generates the simulation environment and the scenario, and sends images through ROS to the system in charge of the decision-making. These images are reshaped to 256 x 256, for the driver attention estimation using ARAGAN Generator (Chapter 3), and to 224 x 224, for the explainable decision-making (Chapter 4). For this application, it is the only topic necessary, as this module is completely based on computer vision. Figure 5.4 shows the topic communication between modules.

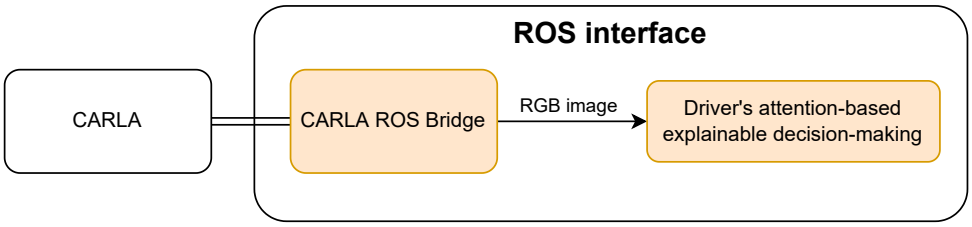


Figure 5.4: Driver's attention-based explainable decision-making ROS interface.

5.4. Scenarios

This section explains the different scenarios built using the simulator to evaluate the applications developed in this dissertation.

5.4.1. Driver Take-Over Request Scenario

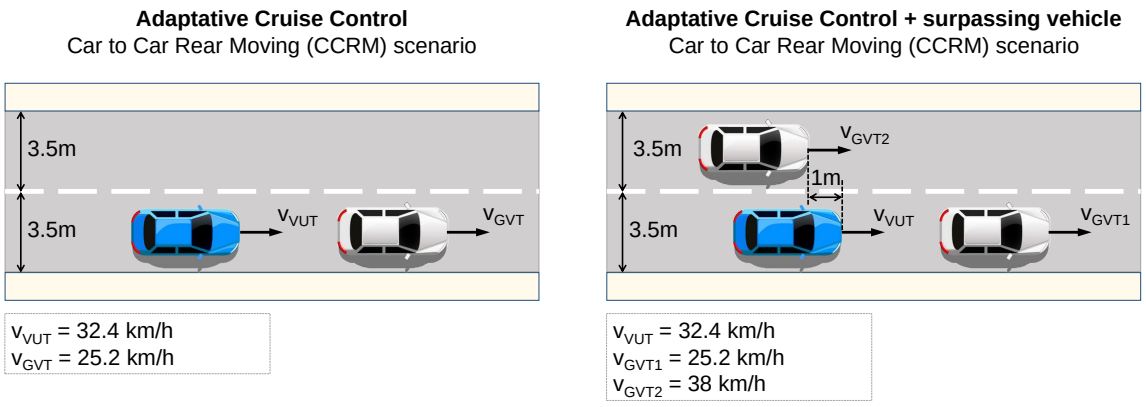
CARLA is mainly utilised for fully autonomous driving development. Consequently, the simulator has been adapted to recreate a real take-over experience, able to shift from autonomous to manual vehicle control. Two challenging scenarios were developed in which users must follow the same route to get statistical results.

Evaluation of Take-Over Request (TOR) experiments in challenging situations is scarce in the literature. This weakness is overcome designing an experiment in two scenarios on a two-lane highway where the ego-vehicle is driving in autonomous mode. To achieve that, the CARLA ScenarioRunner tool [2] has been employed, which allows the definition of scenarios in CARLA. The module works on Python and uses the OpenSCENARIO standard. This standard aims to describe complex manoeuvres in the simulator. They involve multiple entities, such as vehicles, pedestrians and other traffic participants. The description of a manoeuvre may be based on driver actions (e.g. performing a lane change) or on trajectories (e.g. derived from a recorded driving manoeuvre). The proposed scenario uses the second option, where the agents' initial and goal positions and velocities are defined. Additionally, a test bench has been created to test this application.

The first scenario is an ACC (Adaptive Cruise Control). The ego-vehicle is following an adversarial vehicle. The vehicle ahead is running at 25.2 km/h, and the ego-vehicle, which does not respect the safety distance, is running at 32.4 km/h without decreasing its velocity when it approximates the car ahead. Figure 5.5a shows the first scenario where the ego-vehicle (blue) and adversarial (white) are represented. Additionally, other agents in the scenario perform random trajectories following the traffic rules. The users should perform a braking or avoidance manoeuvre by lane change.

The second scenario follows Figure 5.5b, where besides the ACC, another adversarial is driving in the left lane, performing a passing manoeuvre. Vehicle speeds are 25.2 km/h for the one ahead and 38 km/h for the passing vehicle. The ego-vehicle is running at 32.4

[2] https://carla-scenariorunner.readthedocs.io/en/latest/



(a) Adaptive Cruise Control scenario. Ego-vehicle is represented in blue and the other agent in white.

(b) Adaptive Cruise Control while another vehicle overtakes the ego-vehicle in the side lane scenario. Ego-vehicle is the blue car, and the other agent is the white one.

Figure 5.5: Developed scenarios for the proposed experiment.

km/h. In this scenario, road users cannot make a lane change because they would collide with the vehicle is overtaking them.

5.4.2. Driver's Attention-based Explainable Decision-Making Scenario

The proposed scenario for this experiment is based on the CARLA Leaderboard[3]. The purpose of this Leaderboard is to evaluate holistically autonomous vehicle stacks. The chosen route takes place on Town01, provided by CARLA, where some events have been generated to force the module under test to make driving decisions as well as to give an explanation for each decision.

The route and the events can be seen in Figure 5.6. Event positions are marked with the symbols shown in the legend. The scenario is composed of the following use cases:

- Traffic light: the ego-vehicle approximates a traffic light, and the model has to handle the decision in that situation in order to respect the traffic rules.
- Unexpected child: in front of the ego-vehicle, a child pedestrian will cross the road, and the model has to detect that event to stop and not run over the child.
- Unexpected cyclist: a cyclist will cross the street in front of the ego-vehicle, and the model has to decide to stop the vehicle to avoid hitting the cyclist.
- Unexpected pedestrian: the same event as the child but with an adult pedestrian.
- Forward: the vehicle has to follow the path without deviations.
- Turns: the vehicle has to turn to change the trajectory in order to follow the path.

[3] https://leaderboard.carla.org/

The scenario has been designed by concatenating the above use cases into a complete path. The town additionally includes 100 non-playable characters (NPCs), which are autonomous agents (pedestrians, vehicles, etc.) that move according to the scenario.

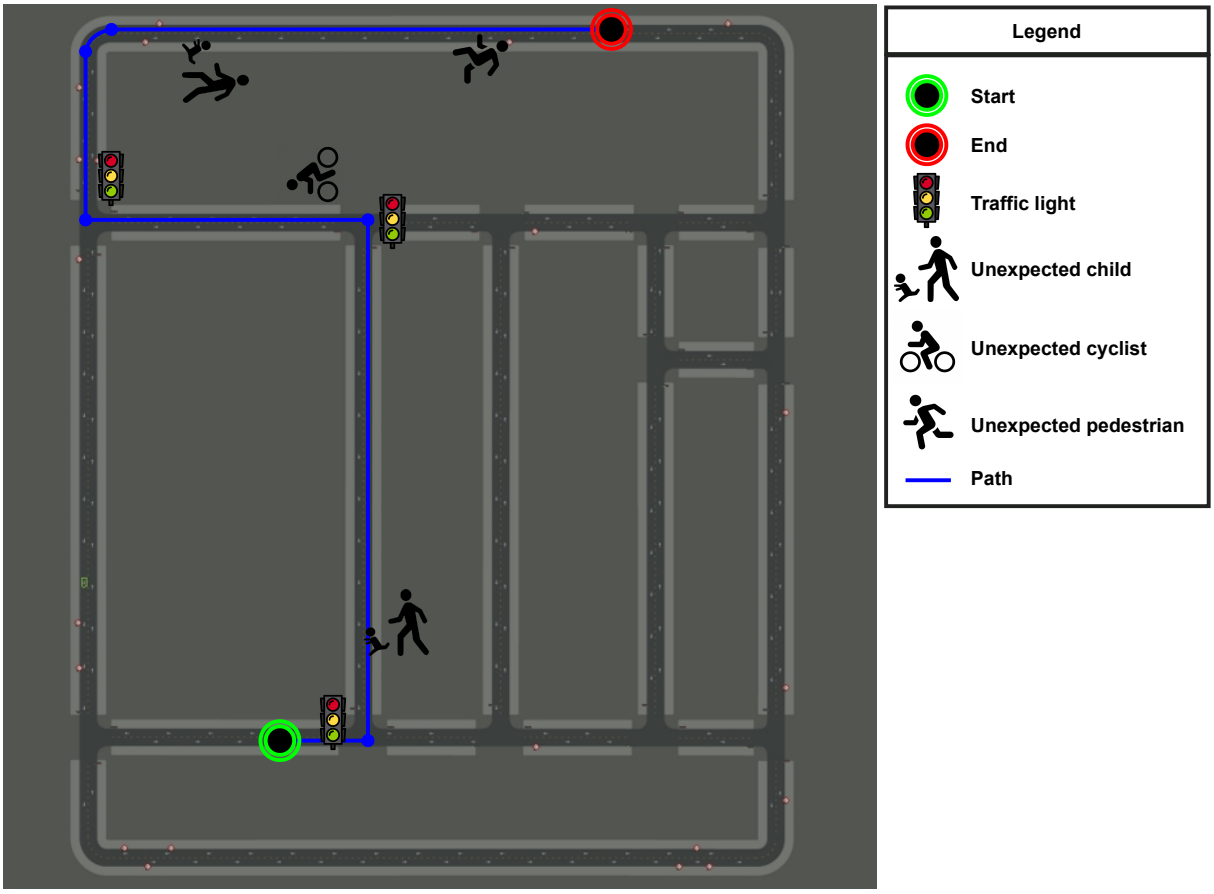


Figure 5.6: Town01 scenario for the driver's attention-based explainable decision-making

5.5. Driver Take-Over Request Application

This section studies driver attention during a take-over request manoeuvre. It was partially published in the following journal Sensors paper [180]: "Driver take-over behaviour study based on gaze focalization and vehicle data in CARLA simulator", and an explainable vehicle's action prediction application.

This application is placed in Levels 3 and 4 according to SAE (J3016), where the vehicle is partially autonomous and sometimes requires human interaction at specific events when the car cannot manage and needs the driver's intervention. Figure 5.7 shows the 5 Levels and the different driver engagement requirements to drive safely. It shows the autonomous Levels where this application takes place. In this study, the driver can engage in Non-Driving Related Tasks (NDRTs), requiring a lapse of time to do the take-over manoeuvre. This lapse begins with the Take-Over Request (TOR) and ends when the driver resumes manual control. The complexity of the driving task and the external conditions (complex

scenarios or bad weather conditions) mean that this period has to be long enough to complete the transition safely. To guarantee this requirement, it is mandatory to check whether the driver's state is suitable for driving. In [184], the authors provide a taxonomy for different forms of autonomous vehicle handover or take-over. The acknowledgement of the driver's awareness in the driving scenario (e.g. surrounding vehicles or pedestrians) is crucial just before the take-over. This evaluation is usually based on the driver's visual focalization, predicted by a gaze tracking estimation since the eye gaze is related to human activities [79].

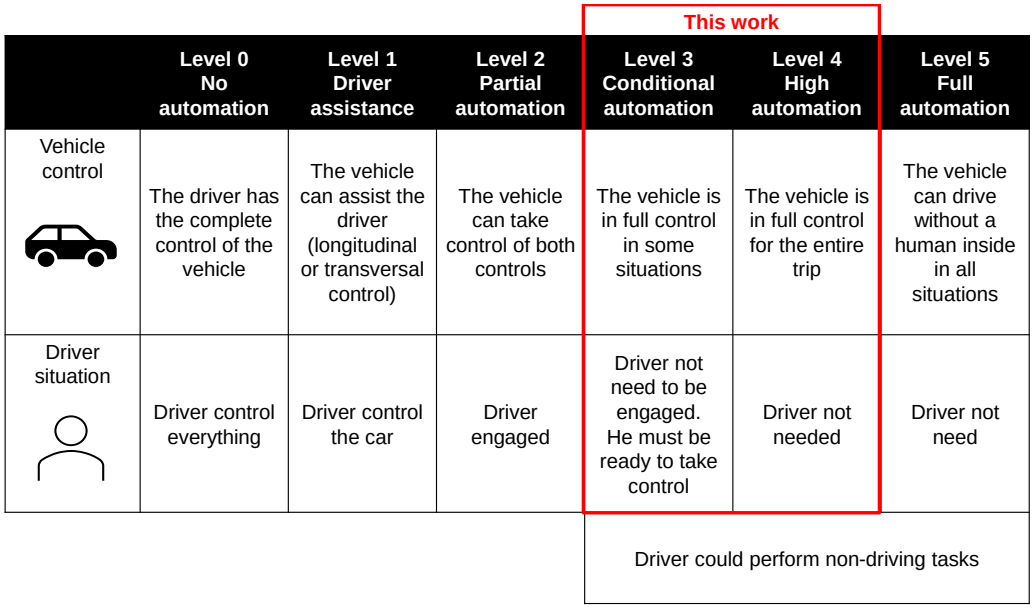


Figure 5.7: Automation Levels, with the level of attention required by the driver.

This application proposes a complete analysis of the driver's behaviour during the take-over manoeuvre to guarantee a safe transition. This study is complex due to the amount of data generated during the transition. Drivers perceive the environment by employing their gaze in a short period of time to make the best decision. For this reason, three parameters are analysed during the experiments:

1. Gaze focalization: the objects the drivers are focalizing their gaze at. This behaviour is studied during the transition and the first moments of manual driving after an emergency TOR.
2. Take-over time: the time it takes for the driver to resume manual driving after TOR.
3. Driver Situation Awareness (DSA): this novel metric assesses the awareness of the driver during the current traffic situation. DSA appends an additional step concerning the calculation of gaze focalization since it compares the focused objects with those that should have been focused according to a supervisor.

These parameters will be analysed in a simulation environment. Several studies have reproduced scenarios using driver simulation environments. They focused their studies

on answering how drivers manage take-over manoeuvres by forcing NDRTs just before the transition. Most of these studies comprising take-over time, quality and prediction performance are conducted in experimental scenarios in rural areas with low traffic density. However, driver's take-over performance still needs to be analysed in stressful traffic use cases in urban scenarios with prolonged NDRT periods engagement before TOR [185].

To meet the above requirements, the open-source CARLA simulator will be employed to devise the proposed experiments. It provides urban layouts, multiple vehicle models, pedestrians, traffic signs, etc., which are valuable tools to contrive and validate autonomous driving vehicles, but not for autonomous/manual transitions controlled from a V2D interface. CARLA provides facilities to build scenarios that can be reproduced to find statistical results. The proposed experiments are conducted in a stressful urban scenario in CARLA, which consisted of an autonomous vehicle approaching a vehicle ahead on a two-lane highway with some variations for each experiment. Inside this scenario, two main use cases are provided: with and without a vehicle passing through the left adjacent lane. The TOR is triggered as a function of the Time To Collision (TTC) parameter with the car ahead. Experiments in this scenario aim to study the drivers' ability to resume manual control while they are/are not engaged in NDRTs in the autonomous mode. A safety module ensures that the drivers are focused on the road during the transition time. If it does not happen, the vehicle will perform an emergency break instead of transferring the vehicle control to the user to avoid a collision with the car ahead.

The main contributions to the TOR study of this dissertation section are:

1. The gaze focalization method based on a camera explained in Chapter 2 will be fused with a semantic segmentation image. This fusion is used to detect the focalized objects.

2. It has been developed a challenging take-over urban scenario based on CARLA. It consists of transferring the control in an approaching manoeuvre with a vehicle ahead as a function of the TTC.

3. Driver's take-over performance has been evaluated in real-time. This application builds a unified framework based on gaze focalization to assess this manoeuvre.

4. A novel DSA metric has been implemented. It uses semantic segmentation and gaze tracking to assess the driver's awareness.

The concerning application has been widely explored in the literature. There are different studies regarding driver take-over behaviour mainly focused on predictive models using machine learning approaches and validated using take-over time and take-over quality performance parameters [186], [187], [188], [189], [190], [191]. Moreover, there are other works focused on the transition study without getting into the

modelling at a safety predictor model [192], [193], [194], [195]. All these works show accuracy metrics but do not consider all the variables that influence the driver's take-over behaviour [196]. Most of these studies are conducted in simulators in rural scenarios with low traffic density [185]. Existing approaches are focused on measuring the reaction time after the TOR signal and they validate their proposals in simulators with less sensory power than the one provided by CARLA or utilise real scenarios with very constrained use cases [197].

These limitations are overcome by a unified framework that analyses the drivers' road scene understanding through their gaze focalization fused with the powerful sensory system provided by CARLA. Moreover, a novel driver situation awareness metric is proposed next to different take-over times, as done in other works in the literature. Besides, two stressful urban scenarios have been developed based on CARLA. They consist of transferring the control of the vehicle while the ego-vehicle is approaching another one. These scenarios represent hazardous use cases with an imminent collision during a TOR. Performing these kind of experiments in physical environments would be very difficult due to the hazard of the situations we want to evaluate. Furthermore, the reproducibility of these experiments would be impossible due to the number of factors involved during the test.

The designed experiments follow the timeline shown in Figure 5.8. It begins with the driver engaged or not in a non-driving task during autonomous driving when an emergency occurs (TTC = 5 s). After that, it performs a take-over request giving the user a transition time (4.5 s). During the transition, the user will have time to gain awareness of the vehicles, pedestrians and other surrounding road elements to have a situation perception and get ready to use the actuators to avoid a hazard. During the experiment, the module analyses the driver behaviour in terms of driver perception using gaze focalization and actuators. Finally, at the end of the transition time, it gives the vehicle control to the user. To the best of our knowledge, this is the first time that the hyper-realistic CARLA simulator is employed in a take-over study that fuses the driver's gaze with the objects that the user under the test is looking at in real-time. With this fusion, the items focalized by the driver during the take-over manoeuvre after a dangerous situation that the autonomous vehicle cannot handle are studied. Besides, this study analyses take-over reaction times.

There are different approaches to perform a TOR. It can be done via auditory, visual or haptic modalities [198], [199], [200]. In these works, a combination of visual and auditory alerts is used. To request the driver to resume manual control both solutions are implemented because the driver will perform NDRTs during the experiment and needs both stimuli to get notified. Haptic modalities on the steering wheel can not fit these experiments because the driver is not in contact with the wheel and would not sense the

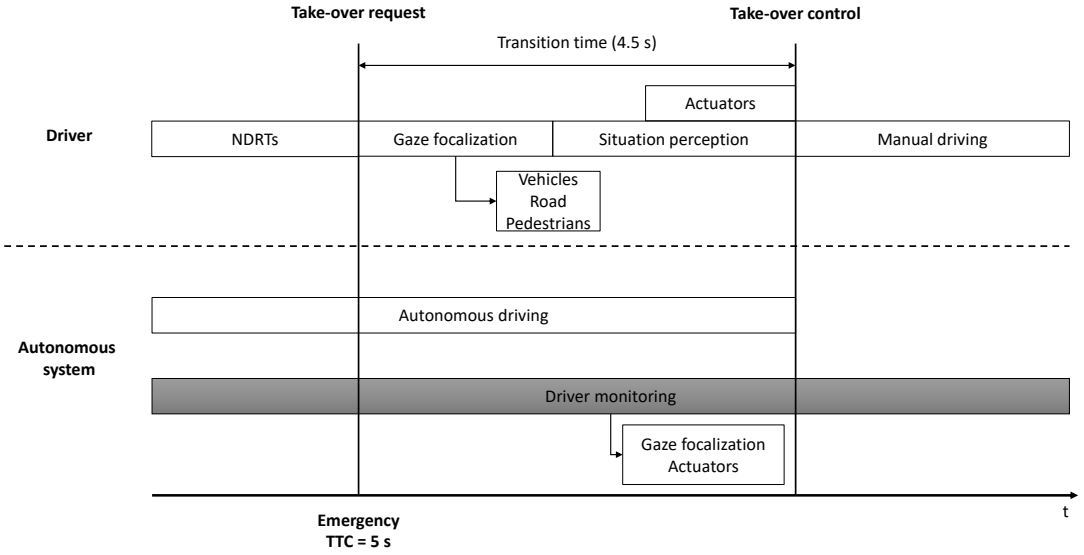


Figure 5.8: Experiment scenario take-over process timeline.

warning when involved in an NDRT. This last modality is extremely valuable in a TOR that occurs in Level 2, where the driver has to be in contact with the steering wheel.

There are several approaches to analysing eye gaze, which were outlined earlier in Chapter 2. These approaches capture the human attention but, from our knowledge, the only one that used gaze focalization in a take-over study is [191] using the Tobii Pro-Glasses 2, which is an expensive and intrusive eye-tracker. It provides accurate gaze information but uses an intrusive and costly sensor. That work used the drivers' gaze focalization to feed a Deep Neural Network to predict their take-over behaviour. But our proposed application goes further by presenting a fusion of gaze orientation with semantic information extracted from a camera in the vehicle, which is looking at the road. By merging the gaze vector and the semantic image, the object that the driver is looking at is obtained.

5.5.1. Experimental Framework

This proposal aims to study the driver's behaviour in take-over manoeuvres using a unified framework based on gaze focalization, scene understanding and vehicle data (steering wheel, throttle and brake). For this task, real-time communication is needed between two data sources (subsystems), the user and the simulator. Figure 5.9 shows the framework used in this work. The two subsystems communicate using Robot Operating System (ROS) [105] due to its power to distribute processes. Figure 5.9 shows the two subsystems with different colours. The first one (the user) is in charge of the OpenFace 2.0 toolkit and the user image monitoring using the ZED camera (green boxes). It transmits the information provided by OpenFace 2.0 toolkit using ROS to the second subsystem (simulator) to perform the calibration and run the simulator (blue boxes).

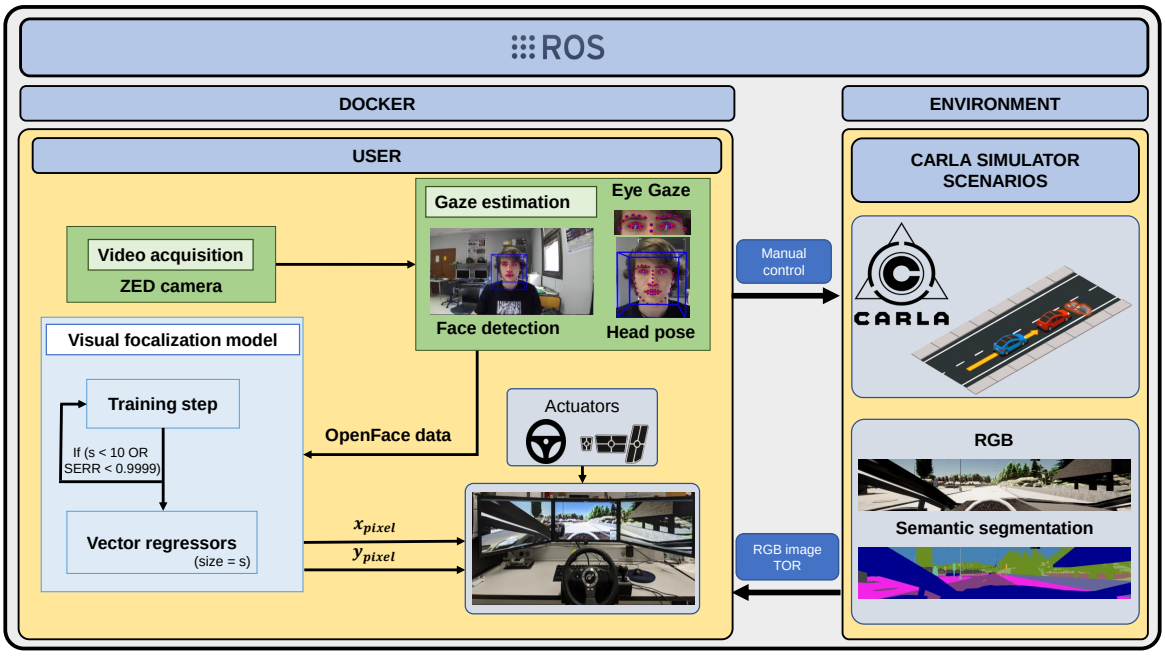


Figure 5.9: Proposed framework to analyse Take-Over Request (TOR). Green boxes determine the first subsystem/computer and blue boxes the second one.

Autonomous vehicles (Level 3 onwards) will allow drivers to engage in NDRTs. The assessment of the driver's visual attention and take-over readiness before the control transfer is crucial. As there is a proven high correlation between the driver's visual attention and gaze tracking, this parameter will be evaluated during driving. The proposed framework is in charge of the gaze focalization system of this application because the achieved results in other tasks make it suitable for it.

As stated before, a distributed system is implemented for performance reasons. Two computers are used due to the simulator requirements (CARLA), which almost needs all the computing power to run the world scenario and the driver view on three screens at a reasonable frame rate. The simulator replicates an authentic stressful scenario in a safe environment where accidents can occur with no personal or material losses.

5.5.1.1. Driver's Gaze Focalization (Subsystem 1)

The gaze focalization subsystem, presented in Chapter 2, is packed in a container using Docker to improve flexibility and compatibility. It runs on a computer (subsystem 1), which processes the camera acquisition and the OpenFace 2.0 processing. It sends the resulting data as a string by the ROS network, which is immensely lighter than an image.

To match the face data provided by OpenFace with the areas of the scene on which the drivers focus their gaze, each user has to perform a calibration procedure at the beginning of each experiment.

5.5.1.2. Simulator (Subsystem 2)

The data sent by the subsystem 1 is unique for each user because the model fits the user under test from scratch. The driver must perform a calibration process previously to the experiments, achieving a model that estimates gaze focalization at 30 fps.

The NARMAX model utilises this data for training. It predicts gaze focalization over the simulator screens using the Forward Orthogonal Least Squares (FOLS) algorithm. Once the model is obtained and loaded, the user must do a quick validation test to verify that the model is working well. The user has to look at the same points as during the training procedure over the three screens. If the gaze projected with the model resembles the test point path, the calibration is over. Otherwise, the user will repeat the calibration process until getting a correct model, following the instructions provided by the supervisor.

The application objective is to study where drivers focalize their attention during the take-over manoeuvre and their actions during this time. For this reason, the autopilot provided by CARLA is employed. It offers autonomous driving facilities for the ego-vehicle. It provides the position of all the objects in the simulated world without compromising the system's performance to pursue safe driving. By default, the vehicle drives at 70% of the allowed speed in the lane and maintains a safe distance from the vehicle ahead. The traffic manager is used to perform the stressful take-over scenario, an API that controls the dynamic of all the objects in the CARLA world. In our case, has been changed to allow collisions of the ego-vehicle with other agents. In addition, the speed limit in the lane has been removed.

The semantic segmentation sensor is used to match the gaze focus to the road scene. It gives a pixel-wise-labelled image with the corresponding class. This technique has been implemented in the autonomous driving paradigm with different neural networks such as ERFNet [201] or DeepLabV3 [202]. Due to the fact that this is not in the scope of this work, we have used the one provided by CARLA, which generates the semantic image synthetically from the RGB image. As the sensor has the same resolution as the image displayed on the screens, the obtained driver's gaze vector using the NARMAX calibration model is projected over the semantic segmentation image giving the focalized class. Bearing in mind the error of the employed focalization system, a circular uncertainty gaze area with a radius of 30 pixels is projected on the image following a conical projection (Figure 5.10). Thus, the segmented object will be the object with the most pixels within this area. This approach is similar to the one carried out in [38] and [39]. They proposed to model the gaze as a cone from the eyes to the windshield.

5.5.1.3. Take-Over Request Trigger

The experiment requests the take-over under the same conditions in both scenarios. The trigger is a function of the Time To Collision (TTC) with the vehicle ahead. The

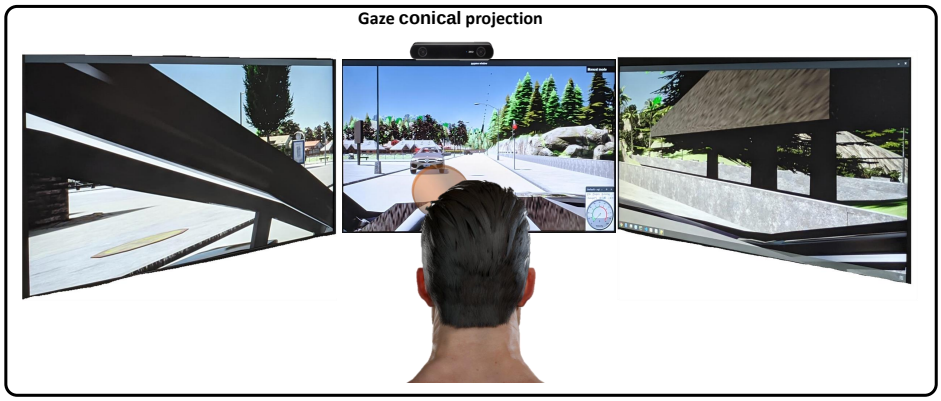


Figure 5.10: Circular uncertainty gaze area with a radius of 30 pixels on the image following a conical projection represented with an orange cone.

Equation 5.1 calculates it, where t is the transition time. This trigger has been chosen to generate a stressful situation. It has been set t to 4.5 seconds because [192] claimed that 4.54 seconds is the average time needed by a driver to take control from autonomous to manual at a simulator.

$$TTC > t \tag{5.1}$$

The Constant Turn Rate and Velocity model (CTRV) [203] obtains the ego-vehicle TTC with the adversarial vehicle. The following positions of each of them are calculated. CTRV follows the state vector shown in Equation 5.2, whose next state is calculated according to the non-linear matrix shown in Equation 5.3. This model predicts the new vehicle position, assuming that the velocity v and the yaw rate ω do not correlate.

$$\overrightarrow{x}(t) = \begin{pmatrix} x & y & \theta & v & \omega \end{pmatrix}^T \tag{5.2}$$

$$\overrightarrow{x}(t+T) = \begin{pmatrix} \frac{v}{\omega} sin(\omega T + \theta) - \frac{v}{\omega} sin(\theta) + x(t) \\ -\frac{v}{\omega} cos(\omega T + \theta) + \frac{v}{\omega} sin(\theta) + y(t) \\ \omega T + \theta \\ v \\ \omega \end{pmatrix} \tag{5.3}$$

The incoming positions for each vehicle are calculated for the next 5 seconds. Kinematic models without additional rules or semantic information are limited in terms of short-term prediction, commonly in a range of up to 3 s. This restriction is not fulfilled because vehicles do not perform any velocity change during autonomous mode.

All the vehicle's positions in the scene are situated as a bounding box (bbox), including the ego-vehicle. In addition, a safety distance of 20 centimetres around each car has been assumed. The following 5 seconds of each vehicle have been discretized in

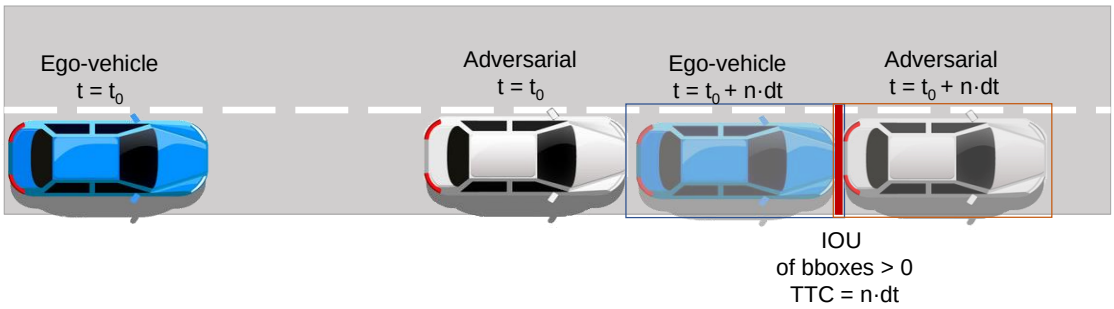


Figure 5.11: CTRV model prediction and IOU between both vehicles to determine TTC.

periods of 0.1 seconds (dt = 0.1). With these timesteps, the positions of each of them have been predicted using the CTRV model as bbox. With these bounding boxes (bboxes), the Intersection Over Union (IOU) is calculated between the ego-vehicle and the surrounding vehicles for each time step. The first time step, when the IOU is more than 0, corresponds with the TTC with the ahead vehicle. Figure 5.11 shows how this approximation works, with two vehicles having an imminent collision due to the difference between both velocities.

5.5.2. Experiment Setup

For this experiment, 20 drivers (15 men and 5 women) were requested to perform the test. They must have been their driving licence and not have ever seen the scenario before. Before the experiment, all the users were informed about the pursued study. Their mean age was 33.5, with a standard deviation of 11.9 years. They had a mean driving licence of 15.3 years with a standard deviation of 12.1 years. The experiment was conducted in Spain, where the minimum age to get a driving licence is 18 years old.

When a dangerous situation (TTC = 5 seconds) takes place, the vehicle will warn the users and give them a transition time of 4.5 seconds to be prepared to take the car control. A driver's behaviour study in this interval is crucial to ensure a safety control transition. This warning (TOR) is received by the user in two different ways, as a sound and warning message on the screen, which indicate that in the next 4.5 seconds the car will change its navigation mode.

During the transition time, if the users do not focalize on the scene because they are engaged in an NDRT, the car will make an emergency break and give them control when the user's gaze is correct. All the data is recorded in a text file for further analysis and comparison with other users.

Before the experiment begins, the user has to perform some previous steps. Firstly, the user must calibrate the gaze focalization system following the procedure exposed before. With the calibrated system, the user will perform some routes using the simulator to get used to it. Because it is the first time they are in the simulator, they need to feel self-

confident about using it to have the best possible results for their behaviour. Finally, with the tool calibrated and the user feeling confident within the simulator, the experiment can begin with the two scenarios presented earlier.

5.5.3. Experiment Results

The presented results correspond to a study with 20 users in a TOR experiment conducted in CARLA. The fusion between gaze focalization and semantic segmentation is studied after a TOR. Moreover, the driver's reaction times are studied. Finally, to understand the situation of the driver and their ability to resume driving, Driver Situation Awareness (DSA) is introduced in this work. It is a novel metric presented on [48] that has never been used in this kind of experiment. It will be used to measure driver awareness after the TOR. The aviation field uses Situation Awareness (SA) [114] to measure the pilots' attention level. Table 5.2 introduces experiments carried out corresponding to each studied behaviour, next to the section localization and Figures/Tables associated to each one.

Table 5.2: Results summary agenda

Behaviour studied	Section	Figure/Table
Gaze focalization object analysis	Section 5.5.3.1.1	Fig. 5.12 and Tab. 5.3
Gaze focalization screen analysis	Section 5.5.3.1.2	Tab. 5.4
Reaction times behaviour analysis	Section 5.5.3.2	Tab. 5.5
Driver Situation Awareness (DSA) analysis	Section 5.5.3.3	Tab. 5.6

5.5.3.1. Gaze focalization analysis

The principal analysis of this work is the study of gaze focalization during the transition in some users. For this purpose, the results are exposed in two ways: focalization over the objects of the scene and over the different screens of the simulator.

5.5.3.1.1. Gaze Focalization over the Objects of the Scene

The fusion between the drivers' gaze focalization and the semantic segmentation during the transition gives information about their behaviour. To summarise the results, the experiment is split into three sections (Manual, Autonomous and Transition), which are the different driving modes the vehicle will find during the route. Matching the gaze and the semantic information will help to understand the driver's awareness in this study.

The drivers focus their gaze on different objects according to the task they are involved in before the TOR. This analysis can be seen in Figure 5.12. It shows the percentage of focalized objects for the three different driving modes. Figure 5.12 is split into two columns, one for each scenario, and four rows, one for each driving behaviour before the

TOR, including some NDRTs cases. The left column shows the ACC use case and the right column the ACC with a passing vehicle use case. Regarding the rows, the first row shows the case in which the users are focused on the autonomous driving. The second row shows the experiment in which the users use their mobile phone during autonomous driving. The third row shows the experiment with the users reading before the TOR. And finally, the fourth row shows the experiment with the users conversing with the co-pilot.

In these figures, it can be appreciated the object where the drivers are focusing their gaze at thanks to the fusion with semantic segmentation. According to Figure 5.12, with the vehicle in autonomous mode and the user involved in monitoring the driving, they focalize their gaze on cars, roads and buildings. It means that they are aware of the driving situation. However, when they are involved in a NDRTs, their gaze is not focused on any object of the image but in "Out of the screen" item.

After the autonomous driving mode, the transition is triggered. According to the result obtained, when the users monitor the driving, they keep focalizing the same objects as in the previous section (autonomous driving mode). That means that the drivers are involved in the driving task even if they are not driving, and they will be able to resume control by being aware of the situation.

If the drivers are involved in NDRTs, they need more time to be aware of the situation. For this reason, the most watched category during the transition is "Out of the screen". What this means is that they do not focus on the simulator and are, therefore, out of the loop. The reason for this result is that the drivers are not aware of the driving, needing time to look at the simulation scene after the request to resume control. After focalizing the gaze on the simulator, the most watched category is "Vehicles". The drivers feel that the cars are the most relevant objects to be aware of to avoid the collision.

Finally, after the transition, the vehicle will change to manual mode if it detects that the drivers have focalized their gaze on the road. Therefore, the ego-vehicle will be in manual mode, and the driver will have to drive. During these experiments, if the vehicle is in manual mode, the most watched category is "Vehicles". Most of the users under the test focalized their gaze on the vehicle ahead to keep performing an ACC.

Besides Figure 5.12, in which the different gaze focalizations during all the modes are shown, Table 5.3 shows the same results but expressed quantitatively. It shows the three most watched objects during the modes (Manual, Autonomous and Transition),S split into focus and NDRTs experiments to compare both results.

When the user is monitoring the driving (focus experiment), the most watched class is "Vehicles" for the three modes (33.07, 38.40 and 46.26 %). This result is because the driver is aware of the vehicles and is the class with most fixations. The second and third most watched classes during this experiment are "Roads", "Buildings" and "Vegetation" because the scenario is composed of these elements that attract the driver's glance.

Besides the focus experiments, the NDRTs experiments are grouped to compare them with the focus experiment. Manual driving has similar class percentages to the focus experiment because the drivers were monitoring the driving (Vehicles (34.64%), Roads (21.96%), and Buildings (13.77%)). Moreover, autonomous and transition modes show that the most watched class is "Out of the screen", which is understandable since the driver was not watching the scene. The percentage of these classes shows that during the transition, users spent less time not looking at the simulator than during autonomous mode. The second most watched class in both modes is "Vehicles" because they focus their attention on them to be aware of their movements. The third class is composed of "Vegetation" and "Buildings" due to the scenario.

Table 5.3: Gaze focalization analysis with semantic segmentation.

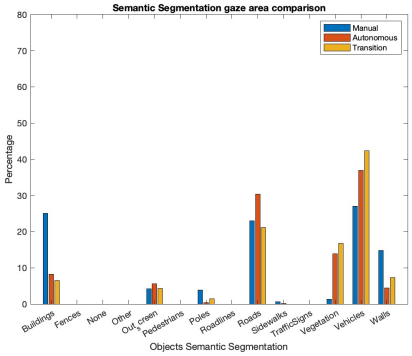
		Gaze focalization classification		
Vehicle state	**NDRTs**	**First class**	**Second class**	**Third class**
Manual	✗	Vehicles (33.07%)	Roads (20.16%)	Buildings (18.64%)
Autonomous	✗	Vehicles (38.60%)	Roads (21.99%)	Vegetation (15.37%)
Transition	✗	Vehicles (46.26%)	Vegetation (17.58%)	Roads (12.66%)
Manual	✓	Vehicles (34.64%)	Roads (21.96%)	Buildings (13.77%)
Autonomous	✓	Out of the screen (66.16%)	Vehicles (11.09%)	Vegetation (6.89%)
Transition	✓	Out of the screen (40.91%)	Vehicles (25.80%)	Buildings (10.81%)

5.5.3.1.2. Gaze Focalization over the Different Screens

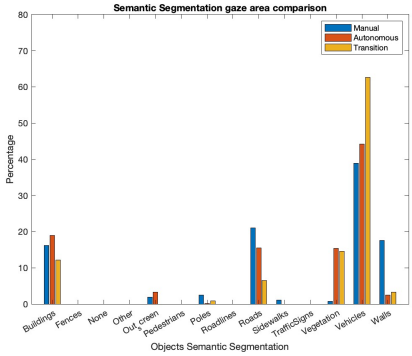
Gaze semantic information about the most watched object by the users during each section of the experiment is the principal evaluation for the TOR study regarding the gaze. Nevertheless, gaze focalization over the different screens of the simulator is an interesting analysis. Table 5.4 shows the screen percentage focalized during the transition after the TOR. Results show that if the drivers are not performing any non-driving task, they focalize the centre screen because it displays the principal object that should be focused on (the vehicle ahead). Otherwise, the gaze is Out of the screen.

Table 5.4: Gaze focalization percentage over the three screens during the transition after the Take-Over Request.

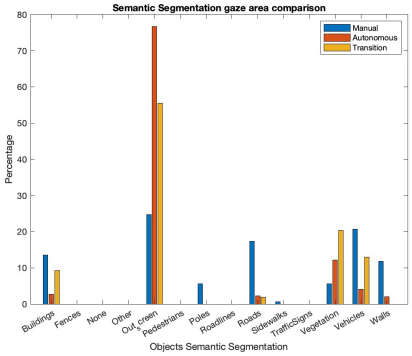
Use case	**Distraction**	**Screen analysis**			
		Left	**Center**	**Right**	**Out of Screen**
ACC	Focus	0.00%	100.00%	0.00%	0.00%
ACC	Mobile	3.59%	72.66%	5.75%	17.98%
ACC	Reading	1.43%	90.64%	3.59%	4.31%
ACC	Talking	6.47%	78.41%	10.07%	5.03%
ACC + Passing vehicle	Focus	0.00%	100.00%	0.00%	0.00%
ACC + Passing vehicle	Mobile	1.44%	74.63%	4.34%	19.56%
ACC + Passing vehicle	Reading	11.51%	76.25%	7.19%	5.03%
ACC + Passing vehicle	Talking	0.00%	84.89%	7.91%	7.19%



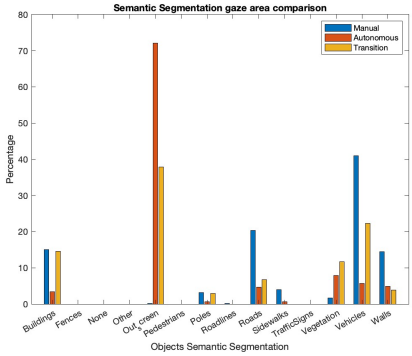
(a) Experiment: ACC. Driver attention: focus



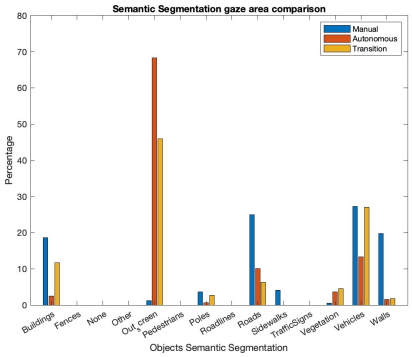
(b) Experiment: ACC + Passing. Driver attention: focus



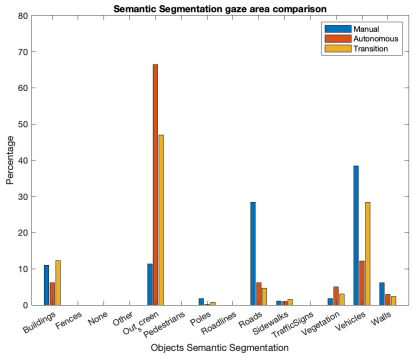
(c) Experiment: ACC. Driver attention: using the mobile phone



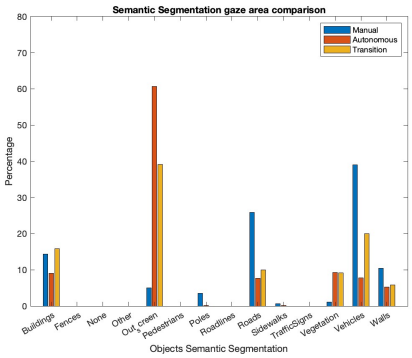
(d) Experiment: ACC + Passing. Driver attention: using the mobile phone



(e) Experiment: ACC. Driver attention: reading



(f) Experiment: ACC + Passing. Driver attention: reading



(g) Experiment: ACC. Driver attention: talking to passenger

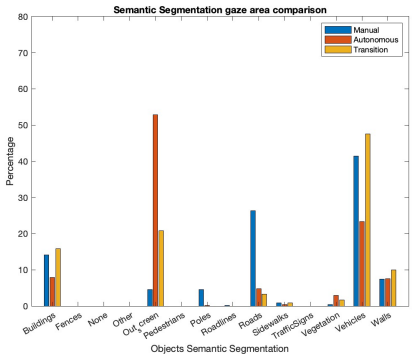


Figure 5.12: Gaze focalization analysis with semantic segmentation.

5.5.3.2. Reaction Times Analysis

This section shows the relationship between the driver and the time. The gaze focus and actuator reaction time will be measured in order to provide this metric evaluated in the literature.

Table 5.5 presents the mean results for all the users. The first three columns are the reaction times in terms of gaze focalization. It is the average time it takes the users to focus on the three most representative objects. These object classes are "Roads", "Vehicles", and "Screen". The last three columns show the actuators' reaction times (throttle, brake and steering wheel). The results are split into use cases and distractions, showing the average values for the NDRTs experiments.

The focus experiment in reaction times shows slight differences for both experiments. It shows that the users first pay attention to the vehicle (0.13 and 0.86 s), and the first used actuator is the brake (2.01 and 1.66 s). Regarding the NDRTs experiments, Gaze focalization reaction times show that the users first pay attention to the screen (1.23 and 1.07 s) because they need to be aware of the scene after performing an NDRT. After that, they focalize the vehicle (3.07 and 1.83 s), with more than a second difference between the ACC and the ACC + Passing vehicle experiment. As for the actuators, the first to be used is, once again, the brake to avoid a collision (1.96 and 2.06 s).

The difference in reaction times between focusing experiments and NDRT is high in terms of the gaze because changing user attention from the NDRT to the driving after the TOR takes more time than when the user is focused on driving, as is reflected in Table 5.5. Moreover, the brake-use times are quite similar between both experiment classifications.

On the other hand, in the most stressful experiment, the reaction times are shorter, except for the throttle, because it is used once the driver has controlled the vehicle. Getting this control takes longer when the stress level is higher.

Table 5.5: Reaction time for the different experiments performed.

		Reaction Time (s)					
Use case	**Distraction**	**Roads**	**Vehicle**	**Screen**	**Throttle**	**Brake**	**Steer**
ACC	Focus	3.08	0.13	0.00	3.22	2.01	3.21
ACC	Mobile	7.07	2.93	1.34	1.88	1.69	2.29
ACC	Reading	2.11	3.56	1.23	1.87	1.87	2.95
ACC	Talking	3.13	2.73	1.13	4.44	2.33	4.37
ACC	NDRTs Average	4.10	3.07	1.23	2.73	1.96	3.20
ACC + Passing vehicle	Focus	4.69	0.86	0.00	5.41	1.66	2.66
ACC + Passing vehicle	Mobile	4.36	1.97	1.08	4.70	2.08	2.25
ACC + Passing vehicle	Reading	3.78	2.22	1.47	5.24	2.24	2.24
ACC + Passing vehicle	Talking	3.64	1.31	0.65	4.29	1.85	2.31
ACC + Passing vehicle	NDRTs Average	3.93	1.83	1.07	4.74	2.06	2.27

5.5.3.3. Situation Awareness

Finally, a metric for measuring the drivers' Situation Awareness (SA) to determine their take-over ability as claimed in [48] is added. It follows the theory explained in Chapter 2.

Table 5.6 shows the Situation Awareness metric results. There is a clear difference between the experiments with the user focused on driving and the experiments where the user performs a NDRTs for the two scenarios. The values are higher in the most stressful scenario (ACC + passing vehicle) because it requires a higher level of awareness. The most distracting task is the mobile in both cases, with reading and talking at an intermediate level of distraction. These conclusions are in line with what was expected and therefore validate the use of this metric in the context of the driver take-over behaviour.

Table 5.6: Situation awareness ratio for the different experiments performed.

Use case	Distraction	Awareness ratio transition
ACC	Focus	0.6131
ACC	Mobile	0.287
ACC	Reading	0.4144
ACC	Talking	0.3583
ACC + Passing vehicle	Focus	0.7602
ACC + Passing vehicle	Mobile	0.4272
ACC + Passing vehicle	Reading	0.4308
ACC + Passing vehicle	Talking	0.6958

5.6. Driver's Attention-Based Explainable Decision-Making Application

This section presents an application conducted in CARLA to evaluate the model proposed in Chapter 4. The experiment assesses the proposal performance in a driving scenario consisting of multiple use cases when the model is assisting a driver. To the best of our knowledge, this is the first time a BDD-OIA trained model has been evaluated in a simulation environment. The driver's attention-based explainable decision-making application shows a Vehicle-to-Driver (V2D) interaction, where the user receives advices from the model on how the vehicle should be driven. The user is required to follow a pre-determined route, which is overlaid in the simulator. During the driving, the model advises the users on possible decisions to be taken in their manoeuvres and provides explanations about these decisions.

To quantify the proposal's performance, we have designed three approaches. Firstly, the driver is queried after finishing the experiment about the decision and explanations taken to quantitative compare with the taken by the model. Secondly, the model's predictions are quantitative compared with all the possibilities given for each designed

use case. Finally, some qualitative examples are presented to showcase the model's performance in some use cases.

In summary, this section presents an experiment to evaluate a BDD-OIA trained model's performance in a simulated environment, where the model advises the driver. The evaluations aim to provide insights into the model's performance by comparing its predictions with the user's decisions and explanations.

5.6.1. Experiment Framework

This experiment aims to assess the driver's attention-based explainable decision-making in a simulation environment. For this purpose, the simulator CARLA is employed. Figure 5.13 shows the framework of this experiment, where the communication has been done using ROS, as stated before. Figure 5.13 shows the three modules that interact with each other to perform the application. These modules are the simulator, the driver and the V2D application.

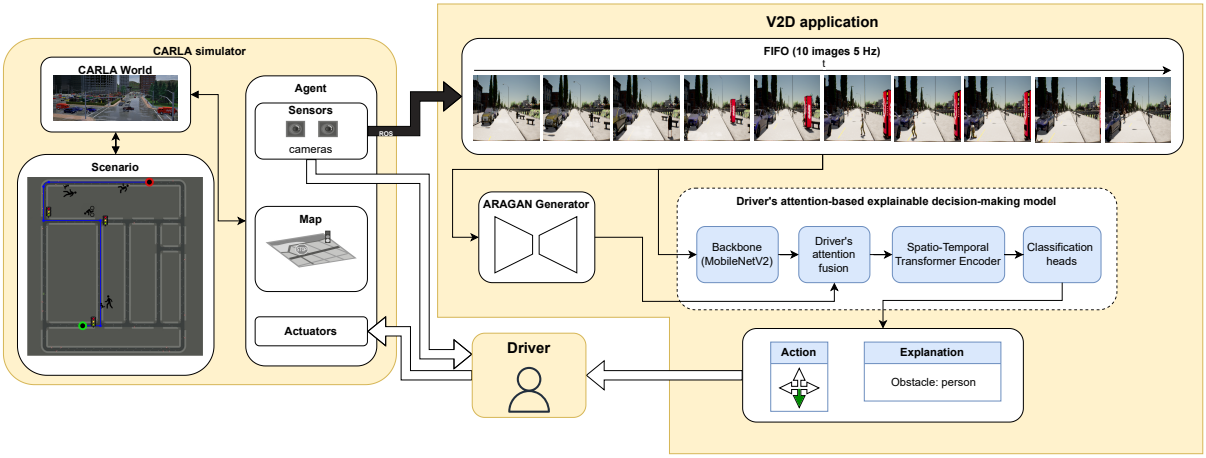


Figure 5.13: Proposed framework to analyse the driver's attention-based explainable decision-making in CARLA.

For this experiment, the CARLA simulator (first module of the framework) has been configured to execute the scenario explained above. It has been done using the scenario manager provided by CARLA.

This scenario sends the information to the CARLA world, which interacts with the agent. The world includes all the NPC agents that the scenario requires to perform all the use cases. The vehicle (agent) has two cameras: one for the user view, which is configured as the TOR experiment to have an interior view of the vehicle, and an external camera that sends the information to the model. Regarding the actuators, the simulator has a steering wheel and pedals (throttle and brake).

The next module are the drivers, who naturalistically drive along the route. They receive the model's output and will be in charge of the posterior analysis to have

quantitative results of the application. They have a view of the vehicle interior similar to a real vehicle using our three-screen simulator.

The last module is in charge of the V2D application. It builds a bridge between the vehicle and the driver to achieve the interaction between both systems. It is composed of three parts: the first collects data from the simulator. The simulation runs at 15 Hz, but as explained in Chapter 4, the driver's attention-based explainable decision-making uses frames at 5 Hz. The model's input is a ten images sequence corresponding to two seconds. To transform from 15 Hz to 5 Hz, one in three frames is fed to a FIFO pipeline, where the sequence is built. The images are resized from the CARLA resolution to the models' input resolution; 256 x 256 for the ARAGAN Generator and 224 x 224 for the explainable decision-making model.

This sequence is passed in batches to the second part of the module, the ARAGAN Generator, which obtains their attention maps in one forward pass. After that, the RGB sequence and their attention maps are fed to the last part, the driver's attention-based explainable decision-making, to predict the action and the explanation of the sequence. This decision and explanation are sent to the driver to close the loop between the driver and the vehicle, performing a V2D application.

5.6.2. Experiment Results

This section explains the result of the developed driver's attention-based explainable decision-making application in the scenario designed in CARLA. For this experiment 5 divers were requested to perform the test in the same conditions that the explained for the take-over application. It exposes the results in three ways: firstly, the predicted actions and explanations have been compared with the decision and the reason made by the user for each designed use case. After that, the predictions have been compared with all the possible actions and explanations associated to each use case, not only the ones made by the driver, as the model has not got route information and cannot predict the decision according to that. For this purpose, the supervisor labels the possible action and explanation for the sequence and compares it with the model's output. Finally, some qualitative examples show some use cases pursued during the experiment to comment on the obtained results.

Table 5.7 shows the first two quantitative comparisons. It has been done in some use cases distributed along the route and for the 5 users of the experiment. The first row shows the comparison against the drivers' decisions and explanations. The results show that the Forward and Stop predictions are perfect while the Left and Right decisions are not fulfilled, with a $F1 = 0.67$. This is because of the unbalanced dataset, which has more sequences of the first two classes. Regarding the global decision results, $mF1$ and $F1_{all}$ are higher than the ones achieved in the test datasets, because the use cases variability for this simulated environment is lower. Nevertheless, the evaluation could be

extended to more use cases to prove the generalization capability of the proposal. Table 5.7 second row also shows the comparison with all the possible actions and explanations associated to each use case. Decision results show better performance than in the dataset, achieving $F1_{all} = 0.81$, but explanation performance decreased from the obtained during the previous test, achieving $F1_{all} = 0.46$. The cause may be the explanations are more difficult to predict, but the scenario actions are easier in a simulated town with less complexity than the real world.

Finally, Figure 5.14 shows some qualitative results for the principal use cases described above. Moreover, Table 5.8 adds the expected decision and explanations labelled by a supervisor to measure the proposal's performance. It shows the actual predictions and the expected ones. After that, the same metric as in previous chapter that computes both $F1_{score}$ ($\overline{F1_{all}}$) is employed.

These results show the domain adaptation of the model trained in a real dataset in a simulation environment. Fine-tuning in some recorded sequences in the simulator could improve the results. Moreover, the exposed results should be taken in caution, because the users' sample is low, and they are subjective, because the supervisors' bias could influence them. However, the decisions made in almost all use cases are the expected ones.

Table 5.7: Comparison between the predicted decisions and explanations in some labelled sequences.

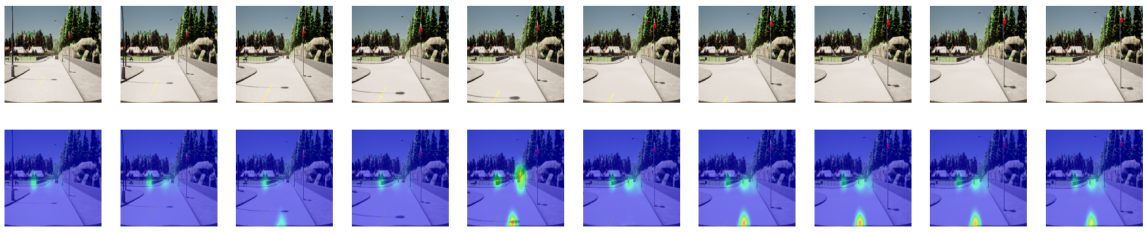
Comparison	Action				Explanation				
	F	S	L	R	mF1	$F1_{all}$	mF1	$F1_{all}$	$\overline{F1_{all}}$
Against the drivers' questionnaire	1	1	0.67	0.67	0.83	0.86	0.16	0.43	0.64
Against all possibilities	0.67	0.80	0.50	0.81	0.69	0.81	0.18	0.46	0.64

5.7. Chapter Summary

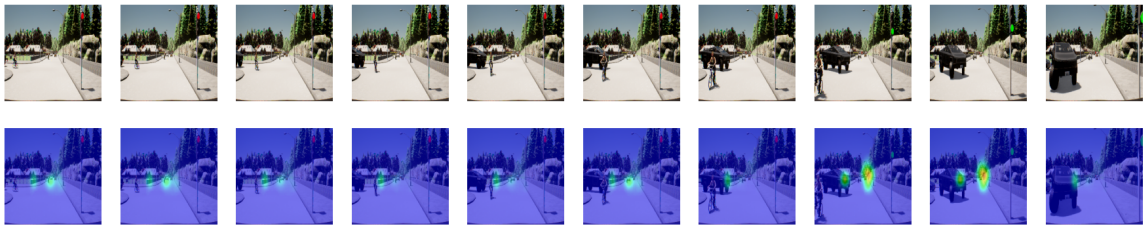
This chapter shows two applications conducted in CARLA to evaluate the models proposed in this thesis. After the validation of our gaze focalization system over a challenging dataset in Chapter 2, the model has been assessed in a Take-Over Request application with novel contributions to the literature. This is a clear V2D application, goal of this thesis. It evaluates the driver's behaviour during the transition in challenging use cases while the drivers were performing NDRTs, applying state-of-the-art metrics (reaction times) and a novel DSA metric for 20 users. Experiments conclude that after the transition, users focalize their gazes on vehicles and perform a sequence according to the hazard situation proposed in the experiments. They aim to avoid a collision with the car ahead. Slight differences depending on the NDRTs are shown on gaze focalization, showing that during the transition, the drivers were more focused on the vehicles and the road than the environment, which is understandable for safe driving. Moreover, the Situation Awareness metric evaluated on these scenarios (DSA) shows that users were aware of the situation most of the time during the transition. The DSA analysis shows that when the drivers were using their

Table 5.8: Qualitative and quantitative results explained for Figure 5.14. Green options denote correct predictions and red options denote incorrect predictions.

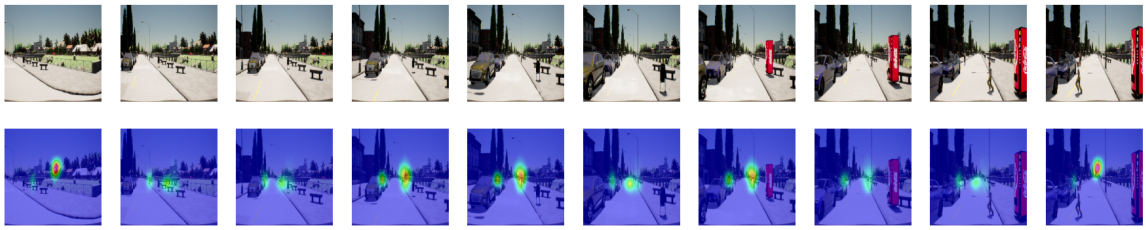
Use case	Fig.	Action		Explanation		$\overline{\text{F1}_{\text{all}}}$	Comments
		Pre	Sup.	Pre	Sup.		
Traffic light	5.14a	Stop	Stop	The traffic light	The traffic light	1	Prediction correct.
Traffic light green	5.14b	Forward Stop Left	Forward Left	Follow traffic The road is clear Obstacles on the left lane	The traffic light is green	0.4	The model mispredicts the explanation but understands the decision.
Unexpected child	5.14c	Forward Stop	Stop	Obstacle: car Obstacle on the left lane No lane on the right	Obstacle: person/pedestrian Obstacle on the left lane No lane on the right	0.67	The model stops due to vehicle detection instead of the pedestrian.
Feed forward	5.14d	Forward	Forward	Follow traffic The road is clear Solid line on the left No lane on the right	The road is clear Solid line on the left No lane on the right	0.75	The model predicts the correct behaviour except for the explanation ("Follow traffic") because there was no traffic in that situation.
Traffic jam	5.14e	Stop	Stop	Obstacle: car No lane on the right	Obstacle: car No lane on the right	1	Perfect prediction.
Traffic light	5.14f	Stop	Stop	The traffic light Obstacles on the right lane	The traffic light	0.83	The model understands the situation and predicts the correct behaviour, but the detected obstacles do not affect it.
Traffic light green	5.14g	Stop Right	Left Right	Obstacle: car Obstacles on the left lane Obstacles on the right lane	Traffic light allows	0.25	The model predicts one of the possible actions but detects obstacles instead of the traffic light.
Unexpected cyclist	5.14h	Stop	Stop	Obstacle: person/pedestrian The traffic light No lane on the right	Obstacle: rider	0.5	The model predicts the emergency but confuses the cyclist with a pedestrian.
Traffic light	5.14i	Stop	Stop	The traffic light Solid line on the left	The traffic light	0.83	The model detects the traffic light and responds correctly to it.
Traffic light green	5.14j	Forward Stop Right	Left Right	Follow traffic Obstacle:car	The traffic light is green	0.2	The model mispredicts the explanations, and it is confused in the decision.
Turning right	5.14k	Forward Stop	Right	The road is clear Obstacle:car Obstacle on the left lane No lane on the right	On the right-turn lane Obstacle on the left lane	0.33	The model does not understand the turning situation. It could be because it is not an intersection, and the model sees a road that continues with a curve, which explains the results.
Unexpected child	5.14l	Stop	Stop	The road is clear No lane on the right	Obstacle: person/pedestrian No lane on the right	0.75	The model fails the most crucial explanations (pedestrian) but correctly predicts the action.
Follow path	5.14m	Forward	Forward	Follow traffic The road is clear Obstacles on the left lane No lane on the right	The road is clear No lane on the right	0.83	The model understands the action, but it gives more than the correct explanations.
Unexpected pedestrian	5.14n	Stop	Stop	Obstacle: car	Obstacle: person/pedestrian	0.5	The model confuses the obstacle, but the decision was correct.



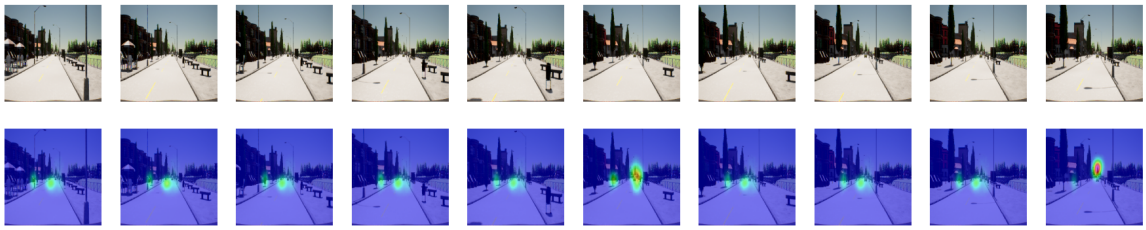
(a) Use case: traffic light is red.



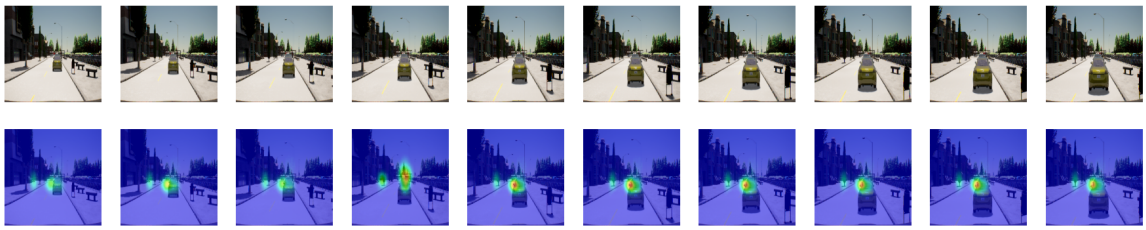
(b) Use case: traffic light is green.



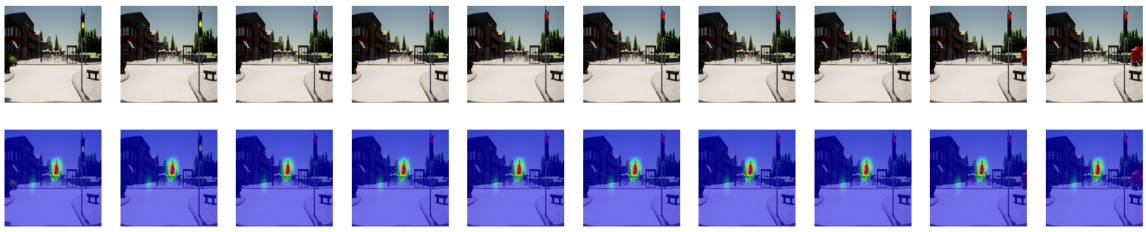
(c) Use case: Unexpected child cross the street.



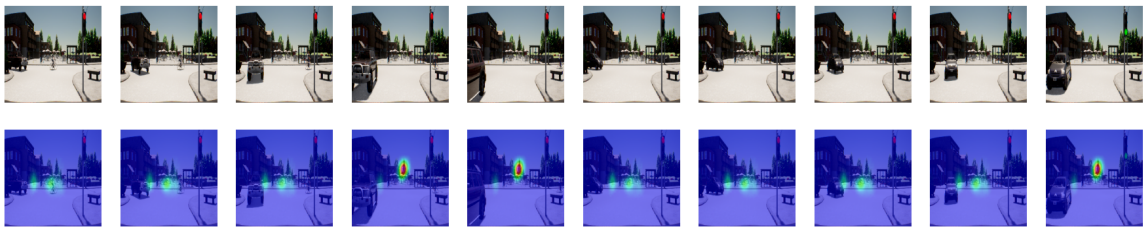
(d) Use case: follow path.



(e) Use case: follow traffic.

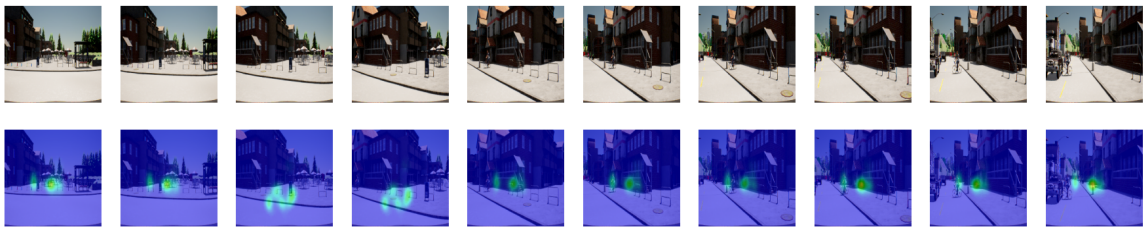


(f) Use case: traffic light is red.

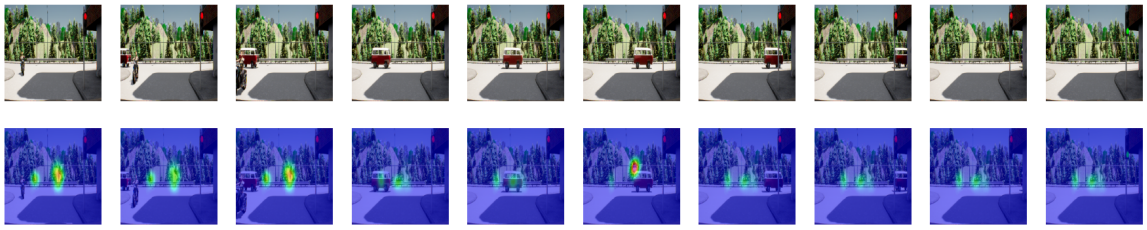


(g) Use case: traffic light is green.

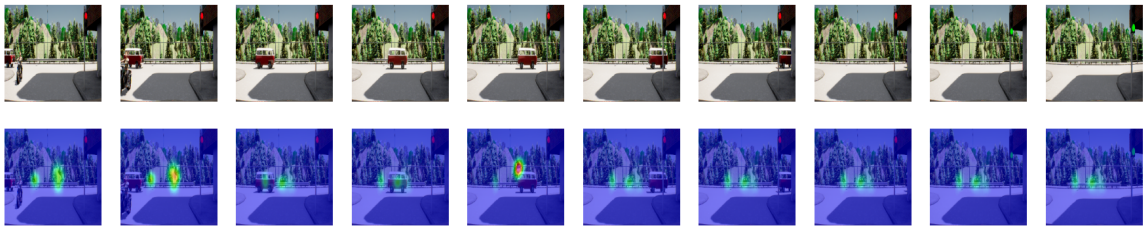
Figure 5.14: Experiment conducted in CARLA to evaluate the driver's attention-based explainable decision-making.



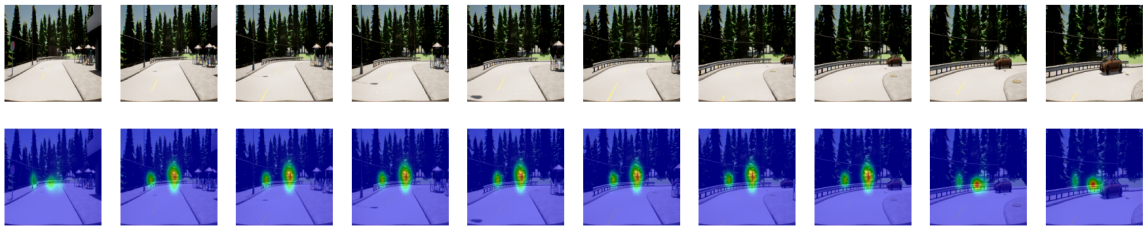
(h) Use case: unexpected cyclist cross the street.



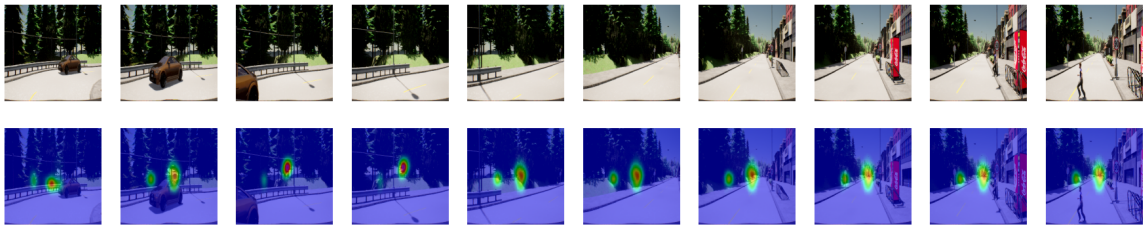
(i) Use case: traffic light is red.



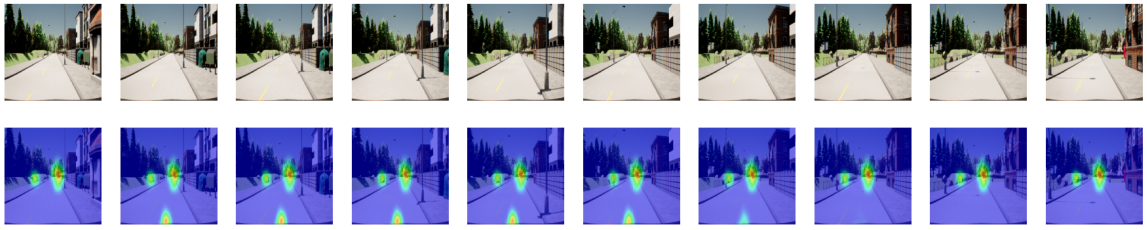
(j) Use case: traffic light is green.



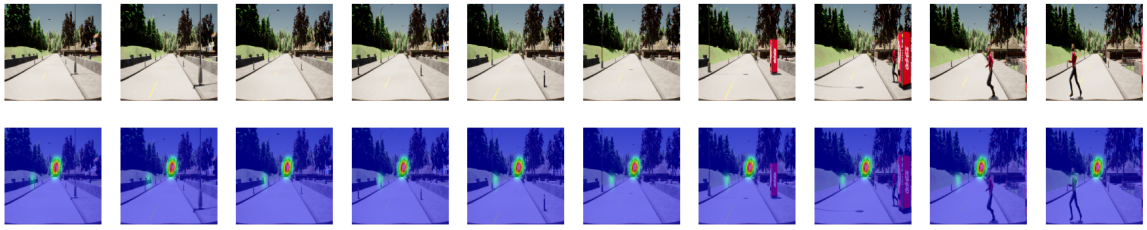
(k) Use case: turning left.



(l) Use case: unexpected child.



(m) Use case: follow path.



(n) Use case: unexpected pedestrian.

Figure 5.14: Experiment conducted in CARLA to evaluate the driver's attention-based explainable decision-making.

phones, they achieved the lowest DSA, inducing that this task is the most distracting one among the proposed here. Our study shows that DSA correlates with user actions.

About this study, it has to be underlined that the sample used in this work is limited, and conclusions obtained may not be generalized. This study is a proof of concept of an intensive study that should follow the evaluations presented here. However, results show similar trends to other state-of-the-art studies, which validate them.

Regarding the V2D application conducted to test the driver's attention-based explainable decision-making model, a scenario with concatenated use cases has been used to evaluate its performance in a simulation environment.

The proposed scenario includes challenging use cases, such as unexpected pedestrians, traffic lights, etc. The model performs correctly, as shown in the result section, with some sequences of the predefined use cases. The results have been commented on qualitatively and quantitatively to show its potential in another domain than the one in which it was trained. Despite using a limited scenario with few users, we believe the model could be generalized for more use cases and more users, being the first to consider a study like this in CARLA.

Chapter 6

Conclusions and future works

"Yo sólo puedo mostrarte la puerta, tú eres quien la tiene que atravesar"

Morfeo; Matrix

6.1. Conclusions

This thesis proposes the study of driver attention through deep learning for a V2D interaction. It is split into three principal models to explore driver attention: monitoring, estimation and assisting in self-driving applications. After that, we build two applications: the first assesses the driver awareness situation after a TOR, and the second assists the driver in challenging use cases.

We present a new low-cost and non-intrusive method to obtain visual focalization maps using the NARMAX method. It works with a camera that can operate in a vehicle. We have validated it in a public dataset (DADA2000), achieving on-par results with an expensive, intrusive and difficult to install in vehicle eye-tracker.

After that, the thesis goes one step further to enter into the deep learning paradigm to develop a driver attention estimator. A novel architecture based on cGANs and Multi-Head Self-Attention is exposed to obtain attention maps in two datasets (BDD-A and DADA2000). It outperforms some representative state-of-the-art proposals, and its inference time allows us to use it in driving applications.

We leverage that model to build another one based on the benefits of our driver attention and a Transformer Encoder to understand relationships among sequential events to obtain an explainable decision-making. The model has been trained and tested in BDD-OIA, outperforming the main state-of-the-art techniques in the test set. We obtain a better scene understanding thanks to the sequential and driver information.

Finally, two V2D applications are exposed to show the proposed models' performance in a simulation environment. Firstly, a take-over request study in a challenging situation

is evaluated with some users to test our driver focalization system. Moreover, it asses the gaze during the transition in different ways to end with a novel DSA evaluation, which fuses the glance and the semantic segmentation to obtain the driver situation awareness. Secondly, we develop a scenario composed of different use cases to evaluate the performance of the driver's attention-based explainable decision-making assisting a driver. We compare the results with some drivers' behaviours and all the possible decisions and explanations generated for multiple use cases. Moreover, we show qualitative results to explain each use case.

6.2. Future Works

Regarding future works to develop after this thesis, we propose several lines of work:

- The driver monitoring system allows for being used in a vehicle for dataset recording or driver attention experiments in real conditions. A recorded dataset would be the first driver attention dataset collected by a camera instead of expensive eye trackers existing in the literature.
- The driver attention estimation model could work in other image-to-image tasks, such as semantic segmentation or depth estimation. In addition, the architecture could be optimised for embedded systems. In order to carry out this task, TensorRT could be employed, which optimises the inference for this kind of system.
- At the beginning of this thesis, generation models were mainly constructed with GANs, because they were the most trending deep learning techniques. However, currently Diffusion models [204] are becoming more popular in data generation. An architecture based on diffusion models for driver attention estimation would be interesting to explore.
- The driver's attention-based explainable decision-making model could advise with textual messages. We have not explored Natural Language Processing (NLP) in this thesis. Some datasets are composed of driver's behaviour explanations as text, such as [74]. Changing its architecture and training in these datasets could be an interesting research topic to analyse, which could generate a text to advise the driver from the information taken from images.
- Regarding the V2D applications in simulation, two improvements can be explored in the future. In the TOR application, the number of users should be extended to different age groups and to the number of scenarios in CARLA. Regarding the driver's Attention-Based Explainable Decision-Making application, the number of scenarios should be extended to other towns with more complexity. Moreover, the V2D interaction should be done with a chatbot to advise the driver.

- The proposed models have been explored in simulation. However, our autonomous vehicle platform [205] should be employed to test these models in a real environment, which is possible thanks to the embedded systems and the optimised architectures used in its implementations.

List of Publications

Journal Papers

- **Araluce, Javier**, L. M. Bergasa, and M. Ocaña, "Driver's attention-based explainable decision-making for V2D applications", Manuscript submitted for IEEE Transactions on Intelligent Transportation Systems, 2023
- **Araluce, Javier**, L. M. Bergasa, M. Ocaña, E. López-Guillén, R. Gutiérrez-Moreno, and J. F. Arango, "Driver take-over behaviour study based on gaze focalization and vehicle data in CARLA simulator", *Sensors*, vol. 22, no. 24, p. 9993, 2022
- **Araluce, Javie**r, L. M. Bergasa, M. Ocaña, E. López-Guillén, P. A. Revenga, J. F. Arango, and O. Pérez, "Gaze focalization system for driving applications using openface 2.0 toolkit with NARMAX algorithm in accidental scenarios", *Sensors*, vol. 21, no. 18, p. 6262, 2021

Conference Contributions

- **Araluce, Javier**, L. M. Bergasa, M. Ocaña, R. Barea, E. López-Guillén, and P. Revenga, "ARAGAN: A dRiver Attention estimation model based on conditional Generative Adversarial Network", in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 1066–1072
- **Araluce, Javier**, L. M. Bergasa, C. Gómez-Huélamo, R. Barea, E. López-Guillén, F. Arango, and Ó. Pérez-Gil, "Integrating OpenFace 2.0 Toolkit for Driver Attention Estimation in Challenging Accidental Scenarios", in *Advances in Physical Agents II: Proceedings of the 21st International Workshop of Physical Agents (WAF 2020), November 19-20, 2020, Alcalá de Henares, Madrid, Spain*, Springer, 2021, pp. 274–288

Bibliography

[1] I. Kotseruba and J. K. Tsotsos, "Attention for vision-based assistive and automated driving: A review of algorithms and datasets", *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[2] M. Sivak, "The information that drivers use: Is it indeed 90% visual?", *Perception*, vol. 25, no. 9, pp. 1081–1089, 1996.

[3] S. G. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, D. J. Ramsey, *et al.*, "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data", United States. Department of Transportation. National Highway Traffic Safety, Tech. Rep., 2006.

[4] C. Robbins and P. Chapman, "How does drivers' visual search change as a function of experience? a systematic review and meta-analysis", *Accident Analysis & Prevention*, vol. 132, p. 105 266, 2019.

[5] *How prevalent are advanced driver assistance systems (adas)?*, 20 2017.

[6] *Consumer reports. guide to cars with advanced safety systems*, 20 2021.

[7] A. El Khatib, C. Ou, and F. Karray, "Driver inattention detection in the context of next-generation autonomous vehicles design: A survey", *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4483–4496, 2019.

[8] S. International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles", *SAE international*, vol. 4970, no. 724, pp. 1–5, 2018.

[9] J. Gaspar and C. Carney, "The effect of partial automation on driver attention: A naturalistic driving study", *Human factors*, vol. 61, no. 8, pp. 1261–1276, 2019.

[10] J. C. De Winter, R. Happee, M. H. Martens, and N. A. Stanton, "Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence", *Transportation research part F: traffic psychology and behaviour*, vol. 27, pp. 196–217, 2014.

[11] A. Feldhütter, C. Gold, S. Schneider, and K. Bengler, "How the duration of automated driving influences take-over performance and gaze behavior", in *Advances in ergonomic design of systems, products and processes: Proceedings of the annual meeting of GfA 2016*, Springer, 2017, pp. 309–318.

[12] T. Vogelpohl, M. Kühn, T. Hummel, and M. Vollrath, "Asleep at the automated wheel-sleepiness and fatigue during highly automated driving", *Accident Analysis & Prevention*, vol. 126, pp. 70–84, 2019.

[13] INPOFV ZERO, "European New Car Assessment Program (Euro NCAP) 2025 roadmap", 2017.

[14] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 5, pp. 1–32, 2021.

[15] G. Osterberg, "Topography of the layer of rods and cones in the human retina", *Acta ophthalmologica*, 1935.

[16] C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson, "Human photoreceptor topography", *Journal of comparative neurology*, vol. 292, no. 4, pp. 497–523, 1990.

[17] M. I. Posner, "Orienting of attention", *Quarterly journal of experimental psychology*, vol. 32, no. 1, pp. 3–25, 1980.

[18] S. Yantis, "Control of visual attention", *attention*, vol. 1, no. 1, pp. 223–256, 1998.

[19] A. M. Treisman and G. Gelade, "A feature-integration theory of attention", *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[20] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[21] A. L. Yarbus and A. L. Yarbus, "Eye movements during perception of complex objects", *Eye movements and vision*, pp. 171–211, 1967.

[22] *Eyelink 1000 plus technical specifications*, 2020.

[23] D. Stavrinos, P. R. Mosley, S. M. Wittig, *et al.*, "Visual behavior differences in drivers across the lifespan: A digital billboard simulator study", *Transportation research part F: traffic psychology and behaviour*, vol. 41, pp. 19–28, 2016.

[24] C. J. Robbins and P. Chapman, "Drivers' visual search behavior toward vulnerable road users at junctions as a function of cycling experience", *Human factors*, vol. 60, no. 7, pp. 889–901, 2018.

[25] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system", *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, 2015.

[26] B. Vasli, S. Martin, and M. M. Trivedi, "On driver gaze estimation: Explorations and fusion of geometric and data driven approaches", in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, pp. 655–660.

[27] J. Huang, Y. Long, and X. Zhao, "Driver glance behavior modeling based on semi-supervised clustering and piecewise aggregate representation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8396–8411, 2021.

[28] M. Lundgren, L. Hammarstrand, and T. McKelvey, "Driver-gaze zone estimation using bayesian filtering and gaussian processes", *IEEE transactions on intelligent transportation systems*, vol. 17, no. 10, pp. 2739–2750, 2016.

[29] A. Palazzi, D. Abati, F. Solera, R. Cucchiara, *et al.*, "Predicting the driver's focus of attention: The dr (eye) ve project", *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1720–1733, 2018.

[30] J. K. Caird and W. J. Horrey, "Twelve practical and useful questions about driving simulation", *Handbook of driving simulation for engineering, medicine, and psychology*, vol. 2, 2011.

[31] Y. Xia, D. Zhang, J. Kim, K. Nakayama, K. Zipser, and D. Whitney, "Predicting driver attention in critical situations", in *Asian conference on computer vision*, Springer, 2018, pp. 658–674.

[32] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li, "DADA-2000: Can driving accident be predicted by driver attention analyzed by a benchmark", in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 4303–4309.

[33] *Road safety. distraction*, 2020.

[34] I. Honda Research Institute USA, *Driver situational awareness.*

[35] Y. Ma, J. Wu, and C. Long, "Gazefcw: Filter collision warning triggers by detecting driver's gaze area", in *Proceedings of the 1st Workshop on Machine Learning on Edge in Sensor Systems*, 2019, pp. 13–18.

[36] A. Tawari, A. Møgelmose, S. Martin, T. B. Moeslund, and M. M. Trivedi, "Attention estimation by simultaneous analysis of viewer and view", in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2014, pp. 1381–1387.

[37] T. Bär, D. Linke, D. Nienhüser, and J. M. Zöllner, "Seen and missed traffic objects: A traffic object-specific awareness estimation", in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2013, pp. 31–36.

[38] S. Zabihi, S. Zabihi, S. S. Beauchemin, and M. A. Bauer, "Detection and recognition of traffic signs inside the attentional visual field of drivers", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 583–588.

[39] T. Langner, D. Seifert, B. Fischer, D. Goehring, T. Ganjineh, and R. Rojas, "Traffic awareness driver assistance based on stereovision, eye-tracking, and head-up display", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 3167–3173.

[40] J. Schwehr, M. Knaust, and V. Willert, "How to evaluate object-of-fixation detection", in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 570–577.

[41] M. Roth, F. Flohr, and D. M. Gavrila, "Driver and pedestrian awareness-based collision risk analysis", in *2016 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2016, pp. 454–459.

[42] M. Rezaei and R. Klette, "Look at the driver, look at the road: No distraction! no accident!", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 129–136.

[43] J. Schwehr and V. Willert, "Multi-hypothesis multi-model driver's gaze target tracking", in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 1427–1434.

[44] T. Kowsari, S. S. Beauchemin, M. A. Bauer, D. Laurendeau, and N. Teasdale, "Multi-depth cross-calibration of remote eye gaze trackers and stereoscopic scene systems", in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, 2014, pp. 1245–1250.

[45] K. Gillmeier, F. Diederichs, and D. Spath, "Prediction of ego vehicle trajectories based on driver intention and environmental context", in *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 963–968.

[46] C. Ahlström, G. Georgoulas, and K. Kircher, "Towards a context-dependent multi-buffer driver distraction detection algorithm", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4778–4790, 2021.

[47] D. Tran, J. Du, W. Sheng, D. Osipychev, Y. Sun, and H. Bai, "A human-vehicle collaborative driving framework for driver assistance", *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3470–3485, 2018.

[48] M. Hofbauer, C. B. Kuhn, L. Püttner, G. Petrovic, and E. Steinbach, "Measuring driver situation awareness using region-of-interest prediction and eye tracking", in *2020 IEEE International Symposium on Multimedia (ISM)*, IEEE, 2020, pp. 91–95.

[49] T. Deng, K. Yang, Y. Li, and H. Yan, "Where does the driver look? top-down-based saliency detection in a traffic driving environment", *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2051–2062, 2016.

[50] F. Katsuki and C. Constantinidis, "Bottom-up and top-down attention: Different processes and overlapping neural systems", *The Neuroscientist*, vol. 20, no. 5, pp. 509–521, 2014.

[51] A. J.-W. Chen, M. Britton, G. R. Turner, J. Vytlacil, T. W. Thompson, and M. D'Esposito, "Goal-directed attention alters the tuning of object-based representations in extrastriate cortex", *Frontiers in human neuroscience*, vol. 6, p. 187, 2012.

[52] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2722–2730.

[53] J. S. Perry and W. S. Geisler, "Gaze-contingent real-time simulation of arbitrary visual fields", in *Human vision and electronic imaging VII*, International Society for Optics and Photonics, vol. 4662, 2002, pp. 57–69.

[54] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving", *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3135–3151, 2019.

[55] Y. Xia, D. Zhang, J. Kim, K. Nakayama, K. Zipser, and D. Whitney, "Predicting driver attention in critical situations", in *Asian conference on computer vision*, Springer, Cham, 2019, pp. 658–674.

[56] H. R. Tavakoli, E. Rahtu, J. Kannala, and A. Borji, "Digging deeper into egocentric gaze prediction", in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 273–282.

[57] M. Ning, C. Lu, and J. Gong, "An efficient model for driving focus of attention prediction using deep learning", in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 1192–1197.

[58] D. Gopinath, G. Rosman, S. Stent, *et al.*, "Maad: A model and dataset for" attended awareness" in driving", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3426–3436.

[59] P. V. Amadori, T. Fischer, and Y. Demiris, "Hammerdrive: A task-aware driving visual attention model", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5573–5585, 2021.

[60] L. Palmer, A. Bialkowski, G. J. Brostow, J. Ambeck-Madsen, and N. Lavie, "Predicting the perceptual demands of urban driving with video regression", in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, pp. 409–417.

[61] Z. Zhang, A. Tawari, S. Martin, and D. Crandall, "Interaction graphs for object importance estimation in on-road driving videos", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 8920–8927.

[62] A. Palazzi, F. Solera, S. Calderara, S. Alletto, and R. Cucchiara, "Learning where to attend like a human driver", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 920–925.

[63] A. Tawari, P. Mallela, and S. Martin, "Learning to attend to salient targets in driving videos using fully convolutional rnn", in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 3225–3232.

[64] J. Fang, D. Yan, J. Qiao, J. Xue, and H. Yu, "DADA: Driver attention prediction in driving accident scenarios", *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[65] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?", *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 740–757, 2018.

[66] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit, "Saliency and human fixations: State-of-the-art and study of comparison metrics", in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1153–1160.

[67] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautamaki, "Multi-task learning with attention for end-to-end autonomous driving", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2902–2911.

[68] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2942–2950.

[69] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, "Explaining autonomous driving by learning end-to-end visual attention", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 340–341.

[70] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[71] J. Kim, S. Moon, A. Rohrbach, T. Darrell, and J. Canny, "Advisable learning for self-driving vehicles by internalizing observation-to-action rules", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9661–9670.

[72] Y. Xia, J. Kim, J. Canny, K. Zipser, T. Canas-Bajo, and D. Whitney, "Periphery-fovea multi-resolution driving model guided by human attention", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1767–1775.

[73] Y. Chen, C. Liu, L. Tai, M. Liu, and B. E. Shi, "Gaze training by modulated dropout improves imitation learning", in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 7756–7761.

[74] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual explanations for self-driving vehicles", in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 563–578.

[75] J. Kim, T. Misu, Y.-T. Chen, A. Tawari, and J. Canny, "Grounding human-to-vehicle advice for self-driving vehicles", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 591–10 599.

[76] Y. Xu, X. Yang, L. Gong, *et al.*, "Explainable object-induced action decision for autonomous vehicles", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9523–9532.

[77] J. Araluce, L. M. Bergasa, M. Ocaña, *et al.*, "Gaze focalization system for driving applications using openface 2.0 toolkit with narmax algorithm in accidental scenarios", *Sensors*, vol. 21, no. 18, p. 6262, 2021.

[78] The European Commission, *New rules to improve road safety and enable fully driverless vehicles in the EU, A subtitle (optional)*. The European Commission, 2022. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/IP_22_4312.

[79] L. Yang, K. Dong, A. J. Dmitruk, J. Brighton, and Y. Zhao, "A dual-cameras-based driver gaze mapping system with an application on non-driving activities monitoring", *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[80] E. Dalmaijer, S. Mathôt, and S. Stigchel, "Pygaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments", *Behavior Research Methods*, vol. 46, Nov. 2013. DOI: 10.3758/s13428-013-0422-2.

[81] M. Cognolato, M. Atzori, and H. Müller, "Head-mounted eye gaze tracking devices: An overview of modern devices and recent advances", *Journal of rehabilitation and assistive technologies engineering*, vol. 5, p. 2 055 668 318 773 991, 2018.

[82] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results", in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec. 2015, pp. 1003–1011. DOI: 10.1109/ICCVW.2015.132.

[83] N. Mizuno, A. Yoshizawa, A. Hayashi, and T. Ishikawa, "Detecting driver's visual attention area by using vehicle-mounted device", in *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, IEEE, 2017, pp. 346–352.

[84] R. A. Naqvi, M. Arsalan, G. Batchuluun, H. S. Yoon, and K. R. Park, "Deep learning-based gaze detection system for automobile drivers using a nir camera sensor", *Sensors*, vol. 18, no. 2, p. 456, 2018.

[85] P. Jiménez, L. M. Bergasa, J. Nuevo, N. Hernández, and I. G. Daza, "Gaze fixation system for the evaluation of driver distractions induced by ivis", *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1167–1178, 2012.

[86] M. Q. Khan and S. Lee, "Gaze and eye tracking: Techniques and applications in adas", *Sensors*, vol. 19, no. 24, 2019, ISSN: 1424-8220. DOI: 10.3390/s19245540. [Online]. Available: https://www.mdpi.com/1424-8220/19/24/5540.

[87] J.-G. Wang, E. Sung, and R. Venkateswarlu, "Estimating the eye gaze from one eye", *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 83–103, 2005.

[88] A. Villanueva, R. Cabeza, and S. Porta, "Eye tracking: Pupil orientation geometrical modeling", *Image and Vision Computing*, vol. 24, no. 7, pp. 663–679, 2006.

[89] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head", in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, IEEE, vol. 2, 2003, pp. II–451.

[90] T. Ohno and N. Mukawa, "A free-head, simple calibration, gaze tracking system that enables gaze-based interaction", in *Proceedings of the 2004 symposium on Eye tracking research & applications*, 2004, pp. 115–122.

[91] A. Meyer, M. Böhme, T. Martinetz, and E. Barth, "A single-camera remote eye tracker", in *International Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems*, Springer, 2006, pp. 208–211.

[92] D. W. Hansen and A. E. Pece, "Eye tracking in the wild", *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 155–181, 2005.

[93] D. W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen, and M. B. Stegmann, "Eye typing using markov and active appearance models", in *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, IEEE, 2002, pp. 132–136.

[94] X. L. Brolly and J. B. Mulligan, "Implicit calibration of a remote gaze tracker", in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, IEEE, 2004, pp. 134–134.

[95] Y. Ebisawa and S.-I. Satoh, "Effectiveness of pupil area detection technique using two light sources and image difference method", in *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Societ*, IEEE, 1993, pp. 1268–1269.

[96] M. S. A. bin Suhaimi, K. Matsushita, M. Sasaki, and W. Njeri, "24-gaze-point calibration method for improving the precision of ac-eog gaze estimation", *Sensors*, vol. 19, no. 17, p. 3650, 2019.

[97] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance", *Real-time imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[98] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications", *Computer vision and image understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[99] H. Singh and J. Singh, "Human eye tracking and related issues: A review", *International Journal of Scientific and Research Publications*, vol. 2, no. 9, pp. 1–9, 2012.

[100] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "OpenFace 2.0: Facial behavior analysis toolkit", in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, IEEE, 2018, pp. 59–66.

[101] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, "Webgazer: Scalable webcam eye tracking using user interactions", in *IJCAI*, 2016.

[102] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers", in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2014, pp. 207–210.

[103] Z. Ye, Y. Li, A. Fathi, *et al.*, "Detecting eye contact using wearable eye-tracking glasses", in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 699–704.

[104] R. Bartolozzi and F. Frendo, "Evaluation of drivers' attention level by a driving simulator", in *Engineering Systems Design and Analysis*, American Society of Mechanical Engineers, vol. 44847, 2012, pp. 155–165.

[105] M. Quigley, K. Conley, B. Gerkey, *et al.*, "Ros: An open-source robot operating system", in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009, p. 5.

[106] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-world dataset and deep appearance-based gaze estimation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 162–175, 2017.

[107] J. Araluce, L. M. Bergasa, C. Gómez-Huélamo, *et al.*, "Integrating OpenFace 2.0 toolkit for driver attention estimation in challenging accidental scenarios", in *Workshop of Physical Agents*, Springer, 2020, pp. 274–288.

[108] S. Chen and S. A. Billings, "Representations of non-linear systems: The narmax model", *International journal of control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[109] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[110] N. Wiener, N. Wiener, C. Mathematician, N. Wiener, N. Wiener, and C. Mathématicien, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, MA, 1949, vol. 113.

[111] S. Billings, M. Korenberg, and S. Chen, "Identification of non-linear output-affine systems using an orthogonal least-squares algorithm", *International Journal of Systems Science*, vol. 19, no. 8, pp. 1559–1568, 1988.

[112] S. Alletto, A. Palazzi, F. Solera, S. Calderara, and R. Cucchiara, "DR(eye)VE: A dataset for attention-based tasks with applications to autonomous and assisted driving", in *Proceedings of the ieee conference on computer vision and pattern recognition workshops*, 2016, pp. 54–60.

[113] M. Mori, C. Miyajima, P. Angkititrakul, *et al.*, "Measuring driver awareness based on correlation between gaze behavior and risks of surrounding vehicles", in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, IEEE, 2012, pp. 644–647.

[114] B. L. Hooey, B. F. Gore, C. D. Wickens, *et al.*, "Modeling pilot situation awareness", in *Human modelling in assisted transportation*, Springer, 2011, pp. 207–213.

[115] J. Araluce, L. M. Bergasa, M. Ocaña, R. Barea, E. López-Guillén, and P. Revenga, "Aragan: A driver attention estimation model based on conditional generative adversarial network", in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 1066–1072.

[116] S. Grubmüller, J. Plihal, and P. Nedoma, "Automated driving from the view of technical standards", in *Automated driving*, Springer, 2017, pp. 29–40.

[117] F. Lateef, M. Kas, and Y. Ruichek, "Saliency heat-map as visual attention for autonomous driving using generative adversarial network (gan)", *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[118] S. Baee, E. Pakdamanian, I. Kim, L. Feng, V. Ordonez, and L. Barnes, "Medirl: Predicting the visual attention of drivers via maximum entropy deep inverse reinforcement learning", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 178–13 188.

[119] Z. Liu, H. Hu, Y. Lin, *et al.*, "Swin transformer v2: Scaling up capacity and resolution", *arXiv preprint arXiv:2111.09883*, 2021.

[120] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet large scale visual recognition challenge", *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[121] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context", in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[122] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20k dataset", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[123] M. Mirza and S. Osindero, "Conditional generative adversarial nets", *arXiv preprint arXiv:1411.1784*, 2014.

[124] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks", *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[125] A. Linardos, M. Kümmerer, O. Press, and M. Bethge, "Deepgaze iie: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 919–12 928.

[126] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling", in *European Conference on Computer Vision*, Springer, 2020, pp. 419–435.

[127] G. Ding, N. Imamouglu, A. Caglayan, M. Murakawa, and R. Nakamura, "SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks", *arXiv preprint arXiv:2112.03731*, 2021.

[128] Z. Bylinskii, T. Judd, A. Borji, *et al.*, "Mit saliency benchmark", 2015.

[129] M. Kummerer, T. S. Wallis, L. A. Gatys, and M. Bethge, "Understanding low-and high-level contributions to fixation prediction", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4789–4798.

[130] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2798–2805.

[131] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research", *CVPR 2015 workshop on "Future of Datasets"*, 2015, arXiv preprint arXiv:1505.03581.

[132] A. Tawari and B. Kang, "A computational framework for driver's visual attention using a fully convolutional architecture", in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 887–894.

[133] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[134] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[135] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", *arXiv preprint arXiv:1511.06434*, 2015.

[136] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.

[137] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures", *arXiv preprint arXiv:1703.03906*, 2017.

[138] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module", in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[139] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[140] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks", in *International conference on machine learning*, PMLR, 2019, pp. 7354–7363.

[141] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[142] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?", in *International Conference on Machine Learning*, PMLR, 2019, pp. 5389–5400.

[143] B. Zhou, H. Zhao, X. Puig, *et al.*, "Semantic understanding of scenes through the ADE20k dataset", *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.

[144] W. Kay, J. Carreira, K. Simonyan, *et al.*, "The kinetics human action video dataset", *arXiv preprint arXiv:1705.06950*, 2017.

[145] C. Gou, Y. Zhou, and D. Li, "Driver attention prediction based on convolution and transformers", *The Journal of Supercomputing*, pp. 1–17, 2022.

[146] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error", *arXiv preprint arXiv:1511.05440*, 2015.

[147] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[148] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention", in *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.

[149] V. Mnih, N. Heess, A. Graves, *et al.*, "Recurrent models of visual attention", *Advances in neural information processing systems*, vol. 27, 2014.

[150] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words", in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/680390c55bbd9ce416d1d69a9ab4760d-Paper.pdf.

[151] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint arXiv:2010.11929*, 2020.

[152] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization", *arXiv preprint arXiv:1607.06450*, 2016.

[153] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks", in *European conference on computer vision*, Springer, 2016, pp. 702–716.

[154] C. Gómez-Huelamo, L. M. Bergasa, R. Barea, E. López-Guillén, F. Arango, and P. Sánchez, "Simulating use cases for the uah autonomous electric car", in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 2305–2311.

[155] Q. Liu, X. Li, S. Yuan, and Z. Li, "Decision-making technology for autonomous vehicles: Learning-based methods, applications and future outlook", in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 30–37.

[156] C. Gómez-Huélamo, A. Diaz-Diaz, J. Araluce, *et al.*, "How to build and validate a safe and reliable autonomous driving stack? a ros based software modular architecture baseline", in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 1282–1289.

[157] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8806–8813.

[158] A. Bewley, J. Rigley, Y. Liu, *et al.*, "Learning to drive from simulation without real world labels", in *2019 International conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 4818–4824.

[159] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[160] O. Makansi, J. Von Kügelgen, F. Locatello, *et al.*, "You mostly walk alone: Analyzing feature attribution in trajectory prediction", *arXiv preprint arXiv:2110.05304*, 2021.

[161] Y. Yao, X. Wang, M. Xu, Z. Pu, E. Atkins, and D. Crandall, "When, where, and what? A new dataset for anomaly detection in driving videos", *arXiv preprint arXiv:2004.03044*, 2020.

[162] T. You and B. Han, "Traffic accident benchmark for causality recognition", in *European Conference on Computer Vision*, Springer, 2020, pp. 540–556.

[163] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7699–7707.

[164] Y. Shen, S. Jiang, Y. Chen, *et al.*, "To explain or not to explain: A study on the necessity of explanations for autonomous vehicles", *arXiv preprint arXiv:2006.11684*, 2020.

[165] F. Yu, H. Chen, X. Wang, *et al.*, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[166] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2174–2182.

[167] R. Girshick, "Fast r-cnn", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[168] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[169] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving", in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8853–8859.

[170] J. Dong, S. Chen, S. Zong, T. Chen, and S. Labi, "Image transformer for explainable autonomous driving system", in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 2732–2737.

[171] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[172] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[173] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[174] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[175] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[176] A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *arXiv preprint arXiv:1704.04861*, 2017.

[177] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

[178] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks", *arXiv preprint arXiv:1905.11946*, 2019.

[179] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset", *International Journal of Robotics Research (IJRR)*, 2013.

[180] J. Araluce, L. M. Bergasa, M. Ocaña, E. López-Guillén, R. Gutiérrez-Moreno, and J. F. Arango, "Driver take-over behaviour study based on gaze focalization and vehicle data in carla simulator", *Sensors*, vol. 22, no. 24, p. 9993, 2022.

[181] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator", *arXiv preprint arXiv:1711.03938*, 2017.

[182] D. Nagaraju, A. Ansah, N. A. N. Ch, *et al.*, "How will drivers take back control in automated vehicles? a driving simulator test of an interleaving framework", in *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2021, pp. 20–27.

[183] J. Wan and C. Wu, "The effects of lead time of take-over request and nondriving tasks on taking-over control of automated vehicles", *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 6, pp. 582–591, 2018. DOI: 10.1109/THMS.2018.2844251.

[184] R. McCall, F. McGee, A. Mirnig, *et al.*, "A taxonomy of autonomous vehicle handover situations", *Transportation research part A: policy and practice*, vol. 124, pp. 507–522, 2019.

[185] G. H. G. Santos and A. P. C. Larocca, "Drivers take-over performance from partial automation to manual driving",

[186] F. L. Berghöfer, C. Purucker, F. Naujoks, K. Wiedemann, and C. Marberger, "Prediction of take-over time demand in conditionally automated driving-results of a real world driving study", *Proceedings of the human factors and ergonomics society Europe*, pp. 69–81, 2018.

[187] A. Lotz and S. Weissenberger, "Predicting take-over times of truck drivers in conditional autonomous driving", in *International Conference on Applied Human Factors and Ergonomics*, Springer, 2018, pp. 329–338.

[188] N. Deo and M. M. Trivedi, "Looking at the driver/rider in autonomous vehicles to predict take-over readiness", *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 41–52, 2019.

[189] N. Du, F. Zhou, E. Pulver, *et al.*, "Predicting takeover performance in conditionally automated driving", in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–8.

[190] N. Du, F. Zhou, E. M. Pulver, *et al.*, "Predicting driver takeover performance in conditionally automated driving", *Accident Analysis & Prevention*, vol. 148, p. 105 748, 2020.

[191] E. Pakdamanian, S. Sheng, S. Baee, S. Heo, S. Kraus, and L. Feng, "Deeptake: Prediction of driver takeover behavior using multimodal data", in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–14.

[192] A. Eriksson, V. Banks, and N. Stanton, "Transition to manual: Comparing simulator with on-road control transitions", *Accident Analysis & Prevention*, vol. 102, pp. 227–234, 2017.

[193] N. Merat, A. H. Jamson, F. C. Lai, M. Daly, and O. M. Carsten, "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle", *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 27, pp. 274–282, 2014, Vehicle Automation and Driver Behaviour, ISSN: 1369-8478. DOI: https://doi.org/10.1016/j.trf.2014.09.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369847814001284.

[194] N. Du, F. Zhou, E. M. Pulver, *et al.*, "Examining the effects of emotional valence and arousal on takeover performance in conditionally automated driving", *Transportation research part C: emerging technologies*, vol. 112, pp. 78–87, 2020.

[195] M. Ebnali, K. Hulme, A. Ebnali-Heidari, and A. Mazloumi, "How does training effect users' attitudes and skills needed for highly automated driving?", *Transportation research part F: traffic psychology and behaviour*, vol. 66, pp. 184–195, 2019.

[196] K. Zeeb, A. Buchner, and M. Schrauf, "What determines the take-over time? an integrated model approach of driver take-over after automated driving", *Accident analysis & prevention*, vol. 78, pp. 212–221, 2015.

[197] A. Rangesh, N. Deo, R. Greer, P. Gunaratne, and M. M. Trivedi, "Autonomous vehicles that alert humans to take-over controls: Modeling with real-world data", *arXiv preprint arXiv:2104.11489*, 2021.

[198] F. Naujoks, C. Mai, and A. Neukum, "The effect of urgency of take-over requests during highly automated driving under distraction conditions", *Advances in human aspects of transportation*, vol. 7, no. Part I, p. 431, 2014.

[199] E. Pakdamanian, L. Feng, and I. Kim, "The effect of whole-body haptic feedback on driver's perception in negotiating a curve", in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications Sage CA: Los Angeles, CA, vol. 62, 2018, pp. 19–23.

[200] J. Wan and C. Wu, "The effects of vibration patterns of take-over request and non-driving tasks on taking-over control of automated vehicles", *International Journal of Human–Computer Interaction*, vol. 34, no. 11, pp. 987–998, 2018.

[201] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[202] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation", *CoRR*, vol. abs/1706.05587, 2017. arXiv: 1706.05587. [Online]. Available: http://arxiv.org/abs/1706.05587.

[203] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems(book)", *Norwood, MA: Artech House, 1999.*, 1999.

[204] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models", *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[205] J. F. Arango, L. M. Bergasa, P. A. Revenga, *et al.*, "Drive-by-wire development process based on ros for an autonomous electric vehicle", *Sensors*, vol. 20, no. 21, p. 6121, 2020.