POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Degree Thesis

Enhancing Automotive Safety through Deep Learning: Development of a Real-Time Gaze Tracking Software for Advanced Driver Assistance Systems (ADAS)

Supervisors

Prof. Massimo SALVI

Dott. Luca BUSSI

Candidate

Giovanni BUONFRATE

March 2024

Freedom is the freedom to say that two plus two make four.
If that is granted, all else follows.
George Orwell, 1984

Summary

Driver distraction during driving remains one of the leading causes of road accidents, with a devastating impact on road safety. Despite significant progress made in recent years in the automotive safety sector, the number of casualties remains unacceptably high, underscoring the urgent need to adopt additional measures to improve road safety.

In this context, the use of Advanced Driver Assistance Systems (ADAS) emerges as a fundamental resource to counteract driver distraction and enhance road safety. The European Directive 2019/2144 represents a significant step forward, mandating the equipping of new vehicles with specific ADAS systems, including the Driver Monitoring System (DMS), designed to monitor driver distraction and prevent potential accidents.

The main objective of the research is to develop advanced software for gaze tracking within the car cabin, aiming to enhance driver distraction detection systems. The proposed methodology consists of several phases, including a detailed analysis of available gaze tracking technologies, selection of the most suitable solution, and development of a sophisticated algorithm for real-time tracking.

To tackle this challenge, two distinct pipelines have been developed. The first adopts a classical one-shot classification approach, widely documented in the literature. However, to enhance performance, a second pipeline based on cascaded networks has been created. Additionally, an ablation study involving various networks and image pre-processing techniques has been conducted to identify the most effective combination for gaze tracking.

For the implementation of this project, it was necessary to create a dedicated gaze tracking dataset in collaboration with Brain Technologies Srl. Recordings were performed under static conditions inside stationary vehicles, involving a group of 20 individuals. A single camera positioned above the speedometer was used, with a resolution of 1080x720 pixels and a frame rate of 30 frames per second (fps).

The research results indicate that a cascade model-based approach for gaze classification, combined with image pre-processing through cropping the upper half of the face, offers superior performance compared to other alternative methodologies. To improve the system and detect not only visual distraction but also other types

of distraction, future steps include the integration of data from additional sensors. Furthermore, it will be crucial to test and adapt the software in a variety of real driving conditions to ensure its effectiveness in different situations. These additional developments will contribute to making the gaze monitoring system more comprehensive and suitable for ensuring greater road safety.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
Artificial Intelligence

PoR
Point of Regard

CNN
Convolutional Neural Network

WHO
World Health Organization

ADAS
Advanced Driver-Assistance Systems

DMS
Driver Monitoring Systems

AoI
Area of Interest

VR
Virtual Reality

PCCR
Pupil Center Corneal Reflection

RF
Random Forest

k-NN

k-Nearest Neighbors

GPR

Gaussian Process Regression

SVR

Support Vector Regression

RCNN

Recurrent Convolutional Neural Networks

mHoG

Histogram of Oriented Gradients

RVR

Relevance Vector Regression

ILSVRC

ImageNet Large Scale Visual Recognition Challenge

HOG

Histogram of Oriented Gradients

SSD

Single Shot Detector

Chapter 1

Introduction

1.1 Background

Road accidents are a significant global cause of concern, with driver distraction playing a crucial role, affecting the frequency and severity of such incidents. According to statistics from the World Health Organization (WHO) [1], despite a 5% decrease in road traffic deaths compared to 2010, approximately 1.19 million people lost their lives in 2021. While this reduction indicates progress, it falls short of the United Nations Decade of Action for Road Safety 2021-2030 target, which aims to halve deaths by 2030.

Driver distraction, as outlined in sources [2], can stem from various factors both inside and outside the vehicle, including technology use unrelated to traffic, and can be spontaneous or contextually imposed. Distractions are typically categorized as visual, auditory, biomechanical, or cognitive, with visual distraction—such as momentarily diverting gaze from the road—being particularly critical and correlated with accidents.

In this regard, the implementation of Advanced Driver Assistance Systems (ADAS) is crucial, representing a significant stride towards safer and more intelligent driving [3]. ADAS systems are designed to monitor vehicles and provide drivers with assistance, offering insights into the current road conditions [4]. These systems encompass technologies like advanced speed control, road sign recognition, and environmental monitoring [4].

Directive 2019/2144, passed by the European Parliament and Council on November 27, 2019, amends Directive 2007/46/EC concerning the type-approval of motor vehicles and trailers [5]. This directive mandates the standard inclusion of certain ADAS systems in all newly approved vehicles, effective from July 6, 2022 [5].

The systems mandated by Regulation (EU) 2019/2144 of the European Union include:

- Lane Departure Warning System (LDWS): This system identifies horizontal road markings and notifies the driver if they unintentionally veer out of their lane.
- Emergency Lane Keeping Assist (ELKS): ELKS automatically guides the vehicle back to the center of the lane if the driver drifts out unintentionally.
- Autonomous Emergency Braking (AEB): AEB senses obstacles on the road and initiates the vehicle's brakes automatically to prevent or mitigate the severity of a collision.
- Driver Drowsiness and Attention Warning Systems (DDAW): DDAW identifies signs of inattention or drowsiness, such as frequent blinking or drooping eyelids, and alerts the driver.
- Advanced Driver Distraction Warning Systems (ADDW): ADDW detects signs of distraction like cellphone usage or distraction from other passengers, providing warnings to the driver.

In addition to the aforementioned systems, there is the Driver Monitoring System (DMS), which is crucial in the context of Advanced Driver Distraction Warning Systems (ADDW). DMS employs sensors and algorithms to analyze the driver's eye and facial movements, providing a comprehensive assessment of their level of concentration [6]. When the software detects prolonged driver distraction, it triggers a warning intended to quickly redirect the driver's focus back to the road ahead [6].

1.2 Driver Monitoring Systems

According to [7], driver monitoring systems are primarily categorized into two types: passive DMS and active DMS.

Passive DMS assesses the driver's state by analyzing steering wheel rotation and driving behavior. In contrast, active DMS, often integrated with cameras and infrared proximity technology, monitors the driver's state by detecting blink frequency, eyelid closures, gaze direction, signs of fatigue, and head movements.

Active DMS employ specific methodologies to achieve the goal of closely monitoring the driver during driving. "**Eye-tracking**" focuses on precisely recording the exact point where the eye is directed, often referred to as the *Point of Regard* (PoR). This method aims to monitor the exact location where the individual is looking inside the vehicle cabin, providing detailed information about eye movement.

Conversely, "**gaze classification**" concentrates on identifying specific *Areas of Interest* (AoI) toward which the driver directs their gaze inside the cabin. This

methodology targets particular areas or regions within the cabin that the driver pays attention to, without necessarily identifying the exact point where the eye is positioned.

Both methodologies, eye-tracking and gaze classification, utilize visual information from the face and eyes to analyze the driver's visual behavior inside the vehicle. However, they differ in the precision of monitoring the exact point of interest or in identifying broader areas of interest. Accurately identifying where the driver is focusing attention inside the vehicle plays a crucial role in preventing distractions and increasing awareness, thus ensuring a safer driving experience.

Several companies (refer to Table 1.1) are dedicated to gaze tracking, extending their operations beyond the automotive sector to various industries such as gaming, medicine, scientific research, marketing, transportation, and more.

Company Name	**Year of Establishment**	**Description**
EyeGaze	1986	The company was founded with the goal of creating a discreet human-computer interface. Their eye-tracking technology has been implemented in research, virtual reality, hospitals, gaming, and other industries.
SR Research	1991	The company produces and distributes high-speed video-based eye-tracking equipment. They have in-house expertise in all aspects of eye-tracking technology, hardware development, production, and software development.
EyeTech	1996	The company primarily focuses on AI applications for health such as patient screening, medical management, progress monitoring, predisposition testing, and more. They offer eye-tracking tools for research and as assistive technology.
Smart Eye	1999	They offer both eye-tracking and head-tracking devices. Their offerings range from small mobile solutions to large-scale eye-tracking systems that can be customized to each client's specifications. Smart Eye primarily focuses on automotive solutions and research tools.
Eye Square	1999	It is one of the leading global providers of implicit and contextual market research technologies. The company pioneered the use of eye-tracking for user and market research. Their technologies measure experience, consumer behavior, advertising impact, neuromarketing, and other market research topics.

Tobii Pro	2001	It is the Tobii Group division providing a range of eye-tracking-based research solutions. They produce screen-based eye trackers, eye-tracking units with glasses, and VR headsets with integrated eye tracking. The company primarily focuses on three specific areas: scientific research, marketing and user research, and industry and human performance.
Visage Technologies	2002	The company is a reliable provider of facial tracking, analysis, and recognition technology with nearly two decades of experience in the field. In addition to eye and gaze tracking, the company also offers head tracking, emotion recognition, age estimation, face verification, and more. Their technology has been used for various applications, from automotive to research, from marketing to entertainment.
Pupil Labs	2014	The company was founded around an open-source eye-tracking platform used by a global community of researchers. The company places a strong emphasis on design and aesthetics, evident in their latest product, the Pupil Invisible, which represents the inaugural eye-tracking device designed to resemble and provide the comfort of regular glasses.
Eyeware	2016	It is a Swiss computer vision company developing 3D eye-tracking software. The company's software development kit can be licensed to be integrated with any device equipped with a 3D camera.

Table 1.1: Table of companies specializing in eye-tracking technology: a list of companies, their respective years of establishment, and a brief description of their activities. The information includes the company name, year of establishment, and an overview of their specializations, such as specific applications, research areas, or types of technologies developed in the field of eye-tracking.

1.3 Thesis Objectives

This thesis aims to develop sophisticated software for real-time gaze tracking within the car cabin, with a primary focus on enhancing driver distraction detection systems in accordance with automotive safety regulations.

The research and development process is divided into three key phases:

1. **Evaluation of Gaze Tracking Technologies**: This phase involves a comprehensive analysis of various available gaze tracking technologies. It will examine their features, performance, strengths, and limitations to determine the most suitable solution for monitoring inside the automotive cabin.

2. **Selection of Optimal Technology**: Building upon the detailed analysis, the optimal eye-tracking technology that best meets the specific requirements of the automotive environment will be chosen. Factors such as accuracy, response times, and practical implementation will be taken into account.

3. **Algorithm Development**: Once the most suitable technology is identified, the focus will shift to designing and implementing a robust algorithm for real-time gaze tracking. This algorithm will be tailored to classify points of interest within the car cabin.

The ultimate goal of this research is to make a significant contribution to the advancement of Advanced Driver Assistance Systems (ADAS). The primary aim is to enhance the ability to detect and prevent driver distraction situations, thereby promoting safer driving practices.

Chapter 2

State of the Art

2.1 Tools for Eye Tracking/Gaze Classification

The evolution of gaze tracking techniques represents a field constantly in flux, adapting to the diverse needs of various applications. In the past, investigating cognitive behavior involved using sensors directly applied to the skin of the face, such as pairs of electrodes, to detect subtle changes in potential generated by eye movements [8]. However, despite the precision of this methodology, it often proved uncomfortable for users [9]. The increasing miniaturization of modern eye trackers, now as compact as a pencil case and increasingly flexible compared to bulky previous versions, is driving their widespread adoption across various research domains [10]. These devices allow for the acquisition of quantitative data without causing discomfort to participants, offering timely results and meaningful visual representations. Eye tracking, by measuring visual attention and interest, has emerged as a key tool for diverse disciplines such as cognitive linguistics, psychology, medicine, marketing, engineering, education, and other fields [11].

Modern eye tracking technologies mainly consist of two elements: a light source, typically infrared, directed towards the eye, and a camera that monitors the pupil and ocular characteristics by reflecting the emitted light [11]. These systems measure the position, movement, and dimensions of the pupil to identify user areas of interest at any given time, thus extending their use in various research domains. There are several categories of eye tracking devices, mainly divisible into two macro-categories: **remote tracking** and **intrusive tracking** [12].

Remote tracking includes screen-based and/or webcam-based devices, allowing monitoring without the need for additional elements on the face. In contrast, intrusive tracking requires the use of wearable elements, such as glasses or other physical devices, to record and monitor eye movement [13]. The choice of the most suitable device depends on the nature of the research, with screen-based

trackers ideal for display interactions, wearables useful for studies in real-world environments, and webcam trackers suitable for large-scale research, albeit with lower precision compared to other types.

2.1.1 Screen-Based Eye Trackers

Screen-based eye trackers are widely used tools in scientific research, especially when the participant interacts with visual stimuli on a display (Figure 2.1) [14]. These devices are distinguished by their high sampling frequency, allowing the acquisition of a large volume of detailed data regarding eye movements [11]. Their utility emerges in analyzing the behavior of individuals with specific medical conditions and in cases where infants or subjects unable to control movements are studied [14, 15]. Data collection and analysis take place in a controlled laboratory environment, enabling researchers to delve into studies on visual behavior, from gaze fixation to microsaccade analysis [16].

2.1.2 Wearable Eye Trackers

Wearable eye trackers are devices similar to common eyeglasses, equipped with cameras positioned near the eyes and infrared LEDs illuminating the eye area [17]. These cameras closely record eye movements, while a front-facing camera captures the driver's view (Figure 2.2), allowing correlation of gaze data with the surrounding environment [18]. The technology for gaze detection through wearable devices is based on tracking ocular features such as the pupil, iris contours, and reflections produced by the infrared LEDs [19]. Typically, calibration of the tools is necessary before using the system for driver gaze detection, to adapt them to each individual's anatomical specifics [18]. However, there are cutting-edge devices that can operate without requiring such calibration [20].

This type of system allows the driver to move their head freely without compromising the accuracy of gaze detection. Numerous studies have been conducted to analyze the visual behavior of drivers in both indoor simulation environments and real driving conditions [18].

These devices emerge as significant tools in the study of human behaviors, enabling other devices such as EEG or other biometrics to gain detailed insights into individual visual interactions [18]. Virtual Reality (VR) headsets with integrated eye-tracking systems allow the analysis of contextual interactions in virtual environments, reducing the need for physical presence. This innovative technology proves extremely beneficial in many fields such as healthcare training. However, its use requires safety precautions and advanced skills for proper configuration [11].

2.1.3 Webcam Eye Trackers

In addition to wearable eye-tracking devices, webcam-based approaches are considered relevant for conducting large-scale research, particularly suited for quantitative analyses [11]. By utilizing webcams, eye tracking leverages integrated or external cameras connected to laptops or monitors (Figure 2.3) to capture data regarding the user's field of view, highlighting the direction of their gaze [21]. However, it is worth noting that the depth and accuracy of information collected through this methodology may be less reliable compared to more advanced eye-tracking methodologies [22].

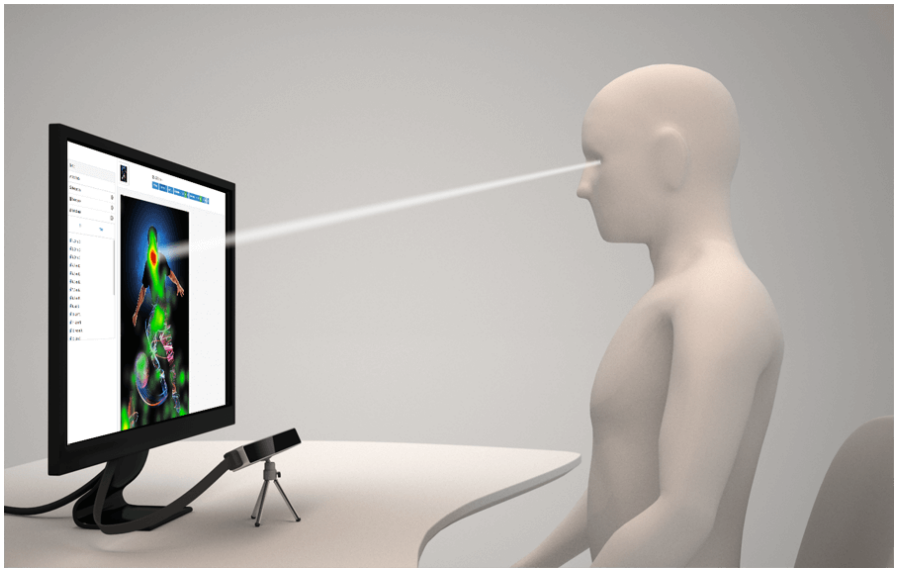


Figure 2.1: The figure illustrates the functioning and setup of screen-based eye trackers. Source [23].

Within the scope of the ongoing study, invasive eye trackers were excluded. Among non-invasive options, a webcam-based solution for eye tracking within the automotive environment was chosen.

Wearable eye trackers, such as those integrated into glasses, offer greater freedom of movement for the driver and the ability to correlate gaze data with the external environment. However, the accuracy of such devices can be influenced by various variables, such as vehicle vibrations, irregular glasses placements, and challenges related to calibration in a dynamic environment, such as during driving.

It should also be noted that they require specific calibration for each individual driver, adding complexity and requiring more time during setup. These wearable eye trackers still represent a significant investment, which could limit accessibility. Besides financial costs, the requirement to wear such devices while driving might

Figure 2.2: The figure illustrates the Tobii Pro Glasses 3 wearable device, highlighting four eye-tracking cameras (indicated by red circles), sixteen illuminators (shown with orange circles), and the scene camera. Source [24].

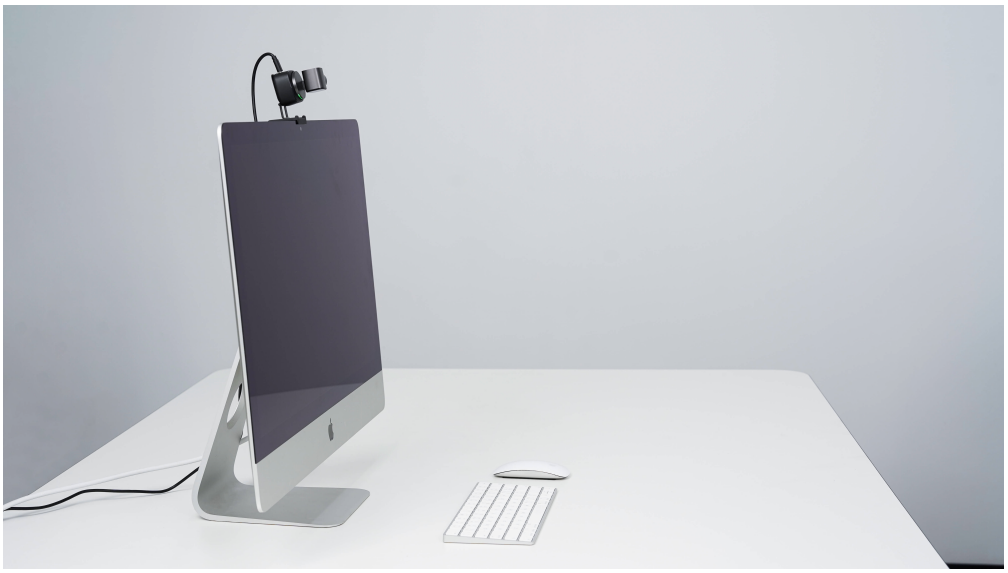


Figure 2.3: The figure illustrates an example configuration of an eye-tracking system using a webcam, with a monitor located below it and an external webcam positioned above the monitor. Source [25].

compromise comfort and naturalness of gesture, introducing potential interferences in the overall driving experience.

Although screen-based eye trackers provide detailed data in controlled environments, it is important to emphasize that they present significant challenges in adapting to real-world conditions of the automotive environment. Differences in the environment, characterized by variables such as vibrations, lighting changes, and irregular movements, could compromise the effectiveness of such devices during use in a moving vehicle. These environmental differences could negatively impact the accuracy and reliability in data collection, as these tools are primarily designed for display environments, such as computer monitors, and may not be optimal for the wide context and unique challenges of the driving experience.

As a result, the webcam-based approach has emerged as the most adaptable and scalable choice for the automotive environment. Despite limitations regarding the accuracy and depth of collected data, the webcam has proven to be more versatile in data acquisition during driving, without restricting the driver's freedom of movement and ensuring more efficient data collection in variable and dynamic conditions.

2.2 History of Eye-Tracking

The history of eye tracking can be traced back to ancient times, where early observations were conducted without the aid of technological instruments. One of the earliest recognized works on eye tracking dates back to 1879 and is attributed to Louise Émile Javal. Javal analyzed people's movements during reading, identifying rapid movements known as saccades, alternating with periods of fixation. However, signs of rapid eye movements had already been detected by Wells in 1792, using a residual image technique, and subsequently illustrated by Crum Brown in 1878 [26].

The significant breakthrough in understanding eye movements occurred in 1908 when Edmund Huey [27] constructed the first device designated as an eye tracker. This pioneering instrument consisted of an aluminum pointer connected to a contact lens, allowing for the tracking of eye movements during reading and significantly contributing to the analysis of this process.

However, the early approaches to monitor eye fixations were invasive, involving direct mechanical contacts with the cornea [28]. For instance, electrooculography relied on the application of electrodes around the eye to measure the electrical potential differences generated by eye movements. Other methods required the use of contact lenses with embedded metal coils at the edges to measure fluctuations in an electromagnetic field caused by movements [29].

As early as 1901, some non-invasive studies were proposed: Dodge and Cline

invented a device that used a camera to measure the speed of eye movements. Although it could detect only horizontal speeds, this instrument represented a first step towards the "remote detection" of the eye and gaze [30]. In 1905, Judd, McAllister, and Steel recorded the temporal aspects of eye movements in two dimensions through a Kinetoscope that projected photos on film, thus advancing in the non-invasive recording of eye movements [31]. Further progress in simultaneous bidimensional recording of eye movements occurred in the early twentieth century, leading to the creation of the first bidimensional records [32, 33].

The first modern head-mounted eye tracker was developed in 1948 [34]. It featured a mouthpiece plate fitting to the subject's teeth, enabling the recording of eye movements even in the presence of head movements.

A pivotal moment in the study of eye movements came in 1967 with the work of Yarbus [35], who aimed to understand how human eyes explore complex objects. The theory of "scanpaths," proposed a few years later by Noton and Stark [36], further expanded the understanding of scene exploration.

From the 1970s, eye movement analysis predominantly involved the use of cameras to detect and locate specific features, facilitating the understanding of eye movements. These approaches focused on detecting variations in luminous contrast, with particular attention to the limbus, the transition zone between the sclera and the iris. Artificial lights and lenses were often utilized to enhance detection. This methodology introduced a new approach to examining gaze, based on the analysis of eye movements through the vector reflexes between the corneal surface (the outer part of the eye) and the pupil. This technique is known as the Purkinje Corneal Reflection Center technique (PCCR).

During this period, two significant contributions emerged in the study of eye movements: the "bright-pupil" technique [37] and the Purkinje image [38]. The "bright-pupil" technique involved using a light source placed near the optical axis of the imaging device to illuminate the pupil. Purkinje images, on the other hand, are at least four images that form on the eye due to reflection with a light source; a more detailed explanation is provided in [39].

Technology based on the analysis of eye reflexes gradually gained ground, and in the 1980s, it was enhanced and standardized with the use of active light sources, often in the infrared spectrum, to improve eye detection and tracking [40]. These reflexes were tracked by infrared cameras, leading to the rise of numerous companies providing PCCR-based eye trackers, employed in both commercial and scientific applications, such as those developed by Tobii, EyeSee, SR Research EyeLink, and Smart Eye.

In the 2000s, RGB-D cameras simplified eye tracker configuration, reducing functional constraints but at the expense of accuracy [41]. For example, Microsoft Kinect's RGB-D camera was widely used in various fields, including eye tracking, due to its lower cost, smaller size, and ability to perceive depth in 3D scenes [42].

Other depth sensors, such as ASUS Xtion Pro Live and Intel RealSense, have been successfully employed for gaze tracking.

Furthermore, glasses have been developed that utilize active infrared light sources to estimate eye directions, also based on corneal reflexes [43].

It's important to note that while these technologies have contributed to the development of gaze tracking, they often involve complex hardware configurations, high costs, and patent protections.

In the last two decades, advances in computer vision and machine learning, combined with increasingly sophisticated and accessible optical sensors, have promoted the development of new methods and systems for automatic extraction of information from eye images [44]. These advanced approaches have allowed gaze tracking to become a key element in various applications, thanks to the ability to extract significant information from raw visual data through advanced machine learning strategies [45, 46, 47, 48, 49].

In conclusion, excluding invasive techniques, the selection of the most suitable technology for investigating gaze tracking within the car could lean towards two main non-invasive approaches. On one hand, the use of RGB-D cameras, such as Microsoft Kinect's camera or other sensors like ASUS Xtion Pro Live and Intel RealSense, provides detailed depth information, simplifying device configuration.

On the other hand, hardware simplification through the use of RGB cameras could be a viable alternative. By leveraging artificial intelligence systems, such as convolutional neural networks (CNN) [50], for facial image processing, accurate results are achieved without the need for complex hardware configurations [51].

2.3 Non-invasive Eye-Tracking Methodologies

In the realm of non-invasive remote eye tracking, researchers typically adopt two main approaches. Some studies delve into modeling the eye itself, emphasizing its geometric aspects. Conversely, other approaches focus on the overall appearance of the face and/or eyes. Appearance-based methods can be further classified into two categories: those that rely on specific facial or eye features for analysis, and those that take an end-to-end approach, aiming to comprehend the overall appearance without detailed distinction of individual features [39].

2.3.1 Geometry-Based Gaze Estimation Methods

In the realm of geometry-based gaze estimation, a variety of approaches are employed. Some methods utilize three-dimensional models of the head and/or eye, grounded in geometric principles. These approaches identify key eye features, such as the center of the pupil, iris margins, and arrangement of facial features, to

ascertain the head and eye positions, as well as their spatial orientation. This gathered information, along with available or estimated three-dimensional knowledge, facilitates estimating the Point of Regard (PoR) by intersecting a three-dimensional straight line with the reconstructed or calibrated scene [39].

Exploring the breadth of existing studies, a crucial factor lies in the presence or absence of a calibration phase. Some approaches depend on person-specific parameter estimation, necessitating initial personal calibration to obtain all parameters [52]. Conversely, in other scenarios, anatomical data [53], mediated [54], or person-independent constraints [55] are employed.

To derive gaze through a geometric approach, two primary challenges must be addressed: creating a three-dimensional model of the eye and integrating it with the head. In its simplest form, the eyeball can be modeled as a sphere [56, 57]. However, more advanced models eschew a single sphere, opting for a combination of two parts [52], which aligns more closely with the medical anatomy of the eye [58]. Generally, works presenting these sophisticated eye models require precise parameter identification in images, often at the expense of operating only in ideal conditions and/or using manually labeled data during the experimental phase. The emergence of low-cost depth sensors (RGB-D) has led to more flexible automatic solutions. However, the precision in terms of depth information and RGB resolution is frequently inadequate for estimating parameters derived from complex anatomical models. This advancement has resulted in less precise but real-time estimates that do not necessitate calibration and are person-independent [39].

Once the parameters of the 3D eye model are estimated, integrating this information with the head pose becomes crucial for tracking a three-dimensional straight line. Various solutions exist for this integration. Some methods estimate the angle between the optical axis and the three-dimensional position of the pupil, effectively tracing a straight line from the corneal center to the pupil position [52, 59]. Alternatively, in certain scenarios, this model is simplified by estimating the center of the eyeball via the direction of the head position vector [57].

In the context of this research, geometry-based methods have been deliberately excluded for several reasons. Firstly, such approaches necessitate personal data of the driver, acquired through either a calibration phase or estimated from a sample population's data. Parameters like interpupillary distance, eyelid fissure width, and distance between outer eye contours vary among individuals, even within the same gender. Errors in estimating such parameters or inaccuracies during calibration can compromise tracking accuracy and functionality, particularly in the dynamic environment of a moving vehicle.

Moreover, the complexity associated with creating three-dimensional eye models presents practical challenges. Models that are overly sophisticated may prove challenging to implement in real-world situations, while overly simplistic models may oversimplify the system, rendering it unsuitable.

Lastly, integrating eye information with head position introduces additional complexities, particularly in uncontrolled environments. These factors justify the preference for alternative approaches, such as appearance-based methods.

2.3.2 Appearance-Based Gaze Estimation Methods

Appearance-based models extract features from regions of the face or eyes, which are subsequently utilized to train a model for predicting the Point of Regard (PoR) or Area of Interest (AoI). The creation of such mappings involves utilizing cropped images of the eyes of one or more subjects directing their gaze towards known positions [57]. In the dedicated literature, various approaches have emerged regarding the utilization of one or both eyes, as well as considering freedom of head movement. In the initial phase, some studies assumed a fixed head position or allowed only slight movements [60, 61, 62]. Subsequently, some investigations integrated head pose information, allowing the user to move their head freely within the scene [63, 64].

Features-Based Methods

In addition to linear regression and its variants [60], various machine learning techniques employing variable feature extractors have been explored in the literature to estimate gaze. One such example involves the utilization of a multi-level Histogram of Oriented Gradients (MHOG) to train Support Vector Regression (SVR) and Relevance Vector Regression (RVR) models within a head-mounted camera context [65]. This demonstrates how low-level gradient features can effectively capture variations in eye appearance.

The challenge of estimating users' gaze while using a tablet is tackled in [66]. Initially, eye detection is performed using a cascade detector, followed by the isolation of identified eye regions. MHOG features are then extracted, and subsequently, linear discriminant analysis is applied to reduce the dimensionality of the feature vector. The output of this process feeds into a Random Forest (RF) regressor. The introduction of the TabletGaze dataset for performance evaluation highlights the innovative aspect of this research. Comparing different combinations of feature extractors and four regressors—k-Nearest Neighbors (KNN), RF, Gaussian Process Regression (GPR), and SVR—constitutes a distinctive element of the proposed approach.

Furthermore, some works have employed depth sensors such as Microsoft Kinect [67]. The central approach involves a 3D rectification of the eye images, creating a standardized view of the head and its scale, irrespective of the actual head position. The use of RGB-D data is crucial for monitoring head position and distorting the eye texture based on the estimated head position. This process aims to achieve

invariance with respect to head pose but requires the adoption of a specific 3D mesh model for each person, learned during the procedure.

End-to-End Systems

The recent widespread adoption of deep learning [68] represents a revolutionary breakthrough in the field of artificial intelligence, extending to gaze analysis as well. The ability to employ end-to-end approaches is a distinctive feature, although alternatives exist without the use of deep neural networks.

An example is found in the work of Noris et al. [69], where Support Vector Regression (SVR) is directly applied to stacked eye image pixels. This methodology aims to minimize intrusiveness by utilizing a motorized mirror to infer gaze, thereby eliminating the need for active illumination. To ensure system effectiveness even with non-cooperative users, online calibration is implemented. Additionally, the system incorporates an eyelid blink detection module to mitigate potential tracking errors.

In the study conducted by Sugano et al. [70], a Random Forest (RF)-based approach is employed to estimate gaze direction from specific normalized eye regions. These regions are determined based on the three-dimensional head position, calculated using six facial landmarks. An innovative aspect is the introduction of synthetic data during training to enhance model performance.

The research presented by Zhang et al. [71] proposes an end-to-end system based on deep learning, where the eye patch is normalized following a methodology similar to that mentioned earlier by Sugano et al. [70]. A distinctive aspect is the integration of the eye patch with head pose angles, configuring a complete input for a LeNet neural network.

Within this study, the authors introduce the MPIIGaze dataset, specifically created to learn the mapping of gaze coordinates. During comparative analysis, the proposed method is juxtaposed with classical machine learning approaches. The results, both in the evaluation across different datasets and within the same dataset, underscore the remarkable performance of Convolutional Neural Networks (CNN), prompting initial reflection on the unique learning capabilities of CNNs in gaze estimation.

In the work by Krafka et al. [72], iTracker is unveiled, a CNN engineered to provide real-time gaze estimation on devices such as smartphones and tablets. This network takes input in the form of a patch containing both eyes, the face image, and a binary mask used to locate the head position in the entire image. The network outputs the distance from the camera, expressed in centimeters.

Another study by Zhu et al. [73] employs two distinct CNNs to model head pose and eyelid movements, respectively. Gaze prediction is subsequently derived through synergistic aggregation of information from both sources, introducing an innovative

layer known as the "gaze transformation layer." This layer is specifically designed to encode the transformation between gaze vectors in head and camera coordinate systems, without the need for introducing additional learnable parameters. Such a decomposition approach is conceived to prevent overfitting related to head position and gaze, mitigating the risk of data recycling intended for different contexts.

Additionally, the use of Recurrent Convolutional Neural Networks (RCNN) has been adopted for gaze estimation, as illustrated in the work by Palmero et al. [74]. This approach integrates face, eye regions, and facial landmarks as individual streams, synergistically combining appearance, shape, and temporal information of image sequences. For each frame, static streams are fused within a multi-stream CNN. The resulting feature vectors are subsequently fed into a many-to-one recurrent module, aimed at predicting the gaze vector for the last frame of the sequence. Incorporating geometric features into appearance-based methods has been shown to exert a regularization effect, thereby contributing to the overall increase in system accuracy.

Often, uncalibrated systems exhibit precision limits and show significant subject-dependent variance and bias. Recently, as discussed in Liu et al. [75], an attempt has been made to improve the performance of end-to-end systems by introducing subject-specific calibration images. This innovative approach directly predicts differences in gaze within the same subject using a differential CNN. Based on a set of these subjective differences, the system can estimate the gaze of a new ocular sample. This methodology proves particularly advantageous in providing more robust estimates in the presence of eyelid closures or lighting perturbations. In experimental contexts, it has been noted that even a single calibration sample can outperform more advanced uncalibrated solutions in performance.

Chapter 3

Eye Tracking

In Section 1.2, the distinction between eye tracking and gaze classification was outlined. This initial chapter aims to explore eye tracking through a detailed examination of a study titled "Monocular Free-head 3D Gaze Tracking with Deep Learning and Geometry Constraints" [73]. This document serves as a recent example that proves useful for illustrating the approach adopted by researchers in the context of gaze tracking via gaze vector regression.

Through this thorough analysis, the necessary clarity will be attained to carefully assess whether to pursue the approach outlined in this study for the specific gaze tracking project within a vehicle or to consider more suitable alternatives.

3.1 Introduction

In the realm of research, the authors delve into the challenge associated with estimating head position via facial landmarks. They acknowledge that such an approach is susceptible to influence from facial expressions and various face configurations, leading to ambiguity in determining head pose both within an individual subject and across different subjects. In response to this ambiguity, the authors opt to separate head tilt modeling from eye movement. Subsequently, they integrate these two components into a gaze vector through geometric transformations, a process termed the "gaze transformation layer." To alleviate the ambiguity introduced by landmarks, they also propose directly estimating head tilt from the face using a Convolutional Neural Network (CNN). Finally, they present a two-phase training strategy aimed at enhancing network learning, as depicted in Figure 3.1.

The advantages of their approach include:

- **Reduced risk of overfitting**: The model mitigates the risk of overfitting by employing a deterministic geometric relationship, making it less susceptible to bias in the head-gaze distribution in the training set.

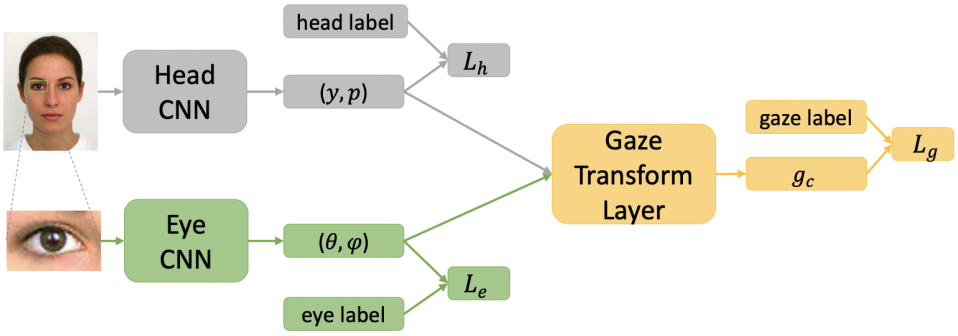


Figure 3.1: The graphical representation of the network structure, as presented in the work of Zhu et al. [73], provides a clear overview of the system. Utilizing two CNNs, it specifically details the head position and eye movements. A key element of this configuration is the introduction of a gaze transformation layer, which effectively integrates this information to accurately formulate the prediction of the focal point. It is noteworthy that this gaze transformation layer is designed to be fully trained, without involving learnable parameters. Source [73]

- **Increased amount of training data**: Existing datasets on head pose can be leveraged for pre-training head pose, and images of eyes from head-worn devices can be used for pre-training on eye movements.
- **Enhanced interpretability and flexibility**: Separate prediction of head pose and eye movements simplifies diagnosis and facilitates updates to the gaze tracking system. Additionally, this approach enables handling scenarios where only head pose information or only eye movements are required, such as when using head-worn devices.

3.2 Gaze Vector

The gaze vector, which is the physical vector to be estimated at the end of the model training, is composed of both eye movement and head pose. The authors use g to denote the physical gaze vector, which can be represented respectively as g_h and g_c in head and camera coordinate systems, as illustrated in Figure 3.2.

Head Coordinate System

The researchers introduce an approximate head coordinate system based on facial landmarks, as described in [70, 71]. This initial system is approximate to handle uncertainties related to head pose. During the second training step, their gaze

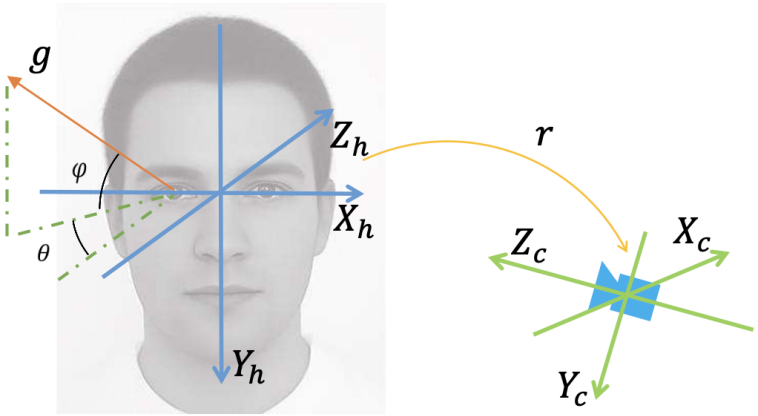


Figure 3.2: Illustration of the head coordinate system (in blue) and the camera coordinate system (in green). The connection between g_h and g_c is determined by the head position R, as indicated in Equation (3.2). Source [73].

transformation layer automatically learns a more accurate head coordinate system, stable within subjects and consistent across subjects.

Eye Movement

As shown in Figure 3.2, the authors define eye movement as the angles between the gaze vector g and the head coordinate axes, denoted as (θ, ϕ), where θ and ϕ represent horizontal and vertical rotation angles respectively. In reality, eye movement represents a gaze vector in the head coordinate system as:

$$\mathbf{g}_h = \left[-\cos(\phi)\sin(\theta), \sin(\phi), -\cos(\phi)\cos(\theta)\right]^T \tag{3.1}$$

Head Pose

Head pose reflects the rotation matrix R between the head and camera coordinate systems. Following data normalization, the head pose, originally with 3 degrees of freedom, is simplified to 2 degrees of freedom as (y, p), where y is the yaw angle and p is the pitch angle in the normalized spherical coordinate system. The relationship between (y, p) and R is bijective, as indicated in [70]. In this context, the notation R = f(y, p) is used.

The gaze vector g_c is defined in the camera coordinate system, and the transformation between g_c and g_h is defined by the head pose R:

$$g_c = R \cdot g_h \tag{3.2}$$

3.3 Eye-tracking Datasets

Within the examined research, the authors of the paper decided to create a custom dataset. The three fundamental principles outlined in the document, considered universal in eye-tracking dataset development, are as follows:

- **Extensive Coverage**: The dataset should encompass a wide range of head poses, eye gaze movements, and their combinations. By involving a significant number of participants, the aim is to ensure good performance across different subjects, accounting for variations in lighting, occlusions, and eyeglass reflections.
- **Device Independence**: A model trained on the dataset should be easily adaptable and deployable on other devices with different parameters for the camera and/or target screen.
- **3D Annotations**: Annotation of head pose and gaze vector in three-dimensional space enables the prediction of a 3D gaze vector rather than being limited to the intersection of gaze with the screen. This makes it possible to use the system to predict gaze intersection with any surrounding object.

3.3.1 Public Datasets

Within the context of gaze tracking research, the use of public datasets represents a crucial element for the development and evaluation of gaze vector estimation algorithms. Among the various datasets, those best suited for the thesis project task are as follows.

The Columbia dataset [76] consists of 5880 images depicting 56 individuals, ensuring diversity across five different head positions and 21 gaze directions. Participants were stably positioned facing a black background, using a chin rest, while a grid of dots was placed on a wall in front of them. Five camera positions, corresponding to each head position, were marked on the floor at a distance of two meters from the subject. Blurred images or situations where the subject closed their eyes were excluded from the data collection. The images have high resolution, 5184×3456 pixels.

EYEDIAP [77] comprises data from Microsoft Kinect and HD camera, synchronized with five LEDs, related to 16 individuals (12 males and four females) distributed over a total of 94 sessions. Users were instructed to sit in front of

the setup and fixate on various reference points, represented by both a three-dimensional ball in the scene and a circle that appeared either discretely (i.e., at random positions) or continuously (i.e., following a random 2-second trajectory) on the computer screen. Annotations regarding head position and eye tracking were generated through an algorithm, supplemented with manually annotated data for unreliable images (e.g., when the user closed their eyes), but only for a subset of sequences.

The UT Multiview dataset [78] comprises video sequences of 50 participants (35 males and 15 females) captured from eight different angles, totaling 160,000 images. Users were positioned 60 cm away from the screen using a chin rest and were instructed to focus on a circle appearing at random positions among 160 discrete cells. 2D and 3D facial landmark annotations are provided, along with verified gaze positions.

The MPIIGaze dataset [79] collects 213,659 images captured during daily laptop usage by 15 participants, over a period of more than three months. The images were acquired via the integrated camera and exhibit considerable variability in terms of appearance and lighting.

3.4 Neural Networks

3.5 Issues with Eye Tracking

The gaze tracking technique discussed, which relies on gaze vector regression, is deemed unsuitable for the current project's context due to several reasons.

Firstly, the authors' choice to employ two separate convolutional neural networks is a commonly observed practice in many research studies, believed to yield superior performance. However, it is crucial to highlight that this decision results in a considerable computational burden, exceeding the available computational resources.

Furthermore, apart from the computational challenge, the inherent complexity of the data and the sophisticated normalization processes present a significant hurdle. Examination of the datasets has unveiled notable variations among the collected data, necessitating a substantial amount of time to comprehend and effectively utilize them.

Lastly, the constraint of time must be emphasized as a critical factor. The scope of the current project demands significant time resources, far exceeding realistic limits for a thesis project. Given these constraints, it becomes imperative to explore more pragmatic alternatives to ensure the validity and feasibility of the thesis work.

Chapter 4

Gaze Classification

As outlined in Section 1.2, an alternative approach to gaze tracking, distinct from eye-tracking, is gaze classification. Unlike eye-tracking, gaze classification doesn't aim to pinpoint a continuous value; instead, it concentrates on identifying discrete directions. To further explore this concept, this chapter will scrutinize the study titled "Driver Gaze Zone Estimation using Convolutional Neural Networks: A General Framework and Ablative Analysis" [80]. This study offers a compelling example to grasp the core tenets of gaze classification while offering valuable insights.

4.1 Introduction

The project addresses a significant challenge: optimizing personalized estimation systems of gaze zones for drivers. Despite numerous research efforts to refine such systems, progress in making this capability universally applicable to diverse drivers, vehicles, perspectives, and contexts has been limited thus far. To tackle this challenge, the authors embraced an approach based on Convolutional Neural Networks (CNN), as depicted in Figure 4.1. These networks have demonstrated remarkable effectiveness in image classification tasks, object detection, and recognition. Particularly noteworthy is the success of CNNs in transfer learning. As emphasized by Oquab et al. [81], it has been observed that image representations learned through CNNs on extensively annotated datasets can be effectively transferred to other visual recognition tasks. Therefore, rather than starting from scratch, the authors adopted the transfer learning paradigm, fine-tuning four different networks previously trained to achieve state-of-the-art results on the ImageNet dataset [82].

This study examines seven distinct gaze zones (Figure 4.2): windshield, right side, left side, infotainment panel, central rearview mirror, tachometer, and the 'closed eyes' state, which frequently occurs when the driver blinks.

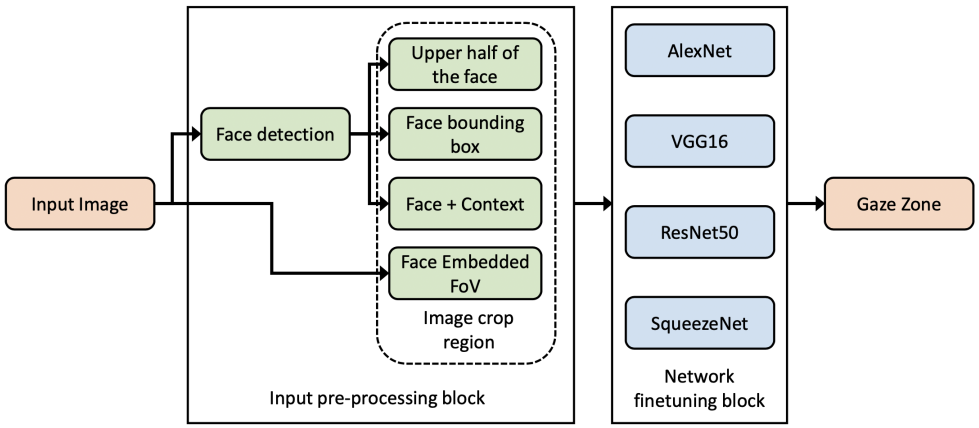


Figure 4.1: An overview of the proposed strategy for selecting the optimal architecture of Convolutional Neural Networks (CNN) and the most effective technique for image preprocessing in the context of gaze zone estimation [80]. The entire process consists of two phases: input preparation and network fine-tuning. During both the training and testing phases, only one of the four input preprocessing techniques and one of the four CNN architectures are selected exclusively. Source [80].

The key elements that distinguish this work are:

- A systematic analysis of ablative effects on different Convolutional Neural Network architectures and input strategies, aimed at improving the generalization of driver gaze zone estimation systems.
- Comparison of the CNN-based model with some of the most advanced approaches in the field.
- Creation of an extensive dataset of natural driving, characterized by significant variability.

4.2 Dataset

The effectiveness of neural networks in generalizing driver gaze zone estimation was assessed through evaluation on an extensive dataset of natural driving. This dataset comprised 11 sessions collected from 10 different subjects aboard two distinct vehicles, each with slight variations in camera settings and fields of view. The vehicles were equipped with internal and one external camera, all synchronized in time, capturing color videos at a resolution of 1280 x 720 pixels, and a rate of

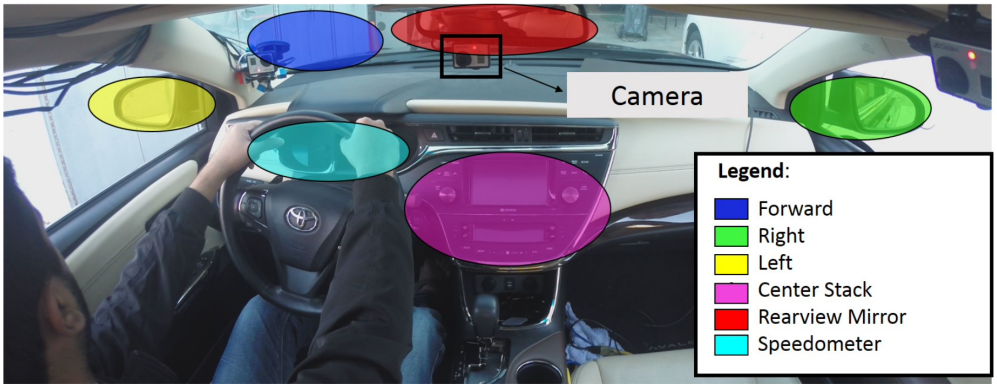


Figure 4.2: Representation of driver gaze zones considered in [80]. Source [80].

30 frames per second. The internal cameras, positioned near the rearview mirror and the A-pillar, were focused on the driver's face, while a human expert manually labeled seven different gaze zones.

Images were extracted from numerous driving "events," ensuring broad coverage of poses and pupil positions. Dataset balancing was achieved by undersampling frontal frames, and the division between training and testing sets considered driving sessions of 7 and 3 subjects, respectively, to enable robust evaluation across different drivers. The distribution of frames for each gaze zone is detailed in Table 4.1.

The images were captured during driving sessions characterized by variations in fields of view. The intrinsic diversity of the dataset plays a crucial role in developing models capable of effectively generalizing under heterogeneous conditions.

Gaze Zones	**Annotated frames**	**Training**	**Testing**
Forward	21522	3505	1023
Right	4216	3195	1021
Left	4751	3725	1022
Center Stack	4143	2831	1159
Rearview Mirror	4489	3533	956
Speedometer	4721	3580	1140
Eyes Closed	3673	2565	1093
Total	**47515**	**22934**	**7414**

Table 4.1: Number of annotated frames as well as frames used for training and testing from the dataset developed in [80].

4.3 Pre-processing

From Figure 4.3, it is evident that the authors considered four distinct approaches for preprocessing input images to CNNs during the training phase. In the first scenario, they utilized the integrated driver's field of view as input, delineating the area between the rearview mirrors. This decision enabled them to explore the feasibility of training the network directly from images, thus bypassing the face detection step. In the second approach, they employed the algorithm proposed by Yuen et al. [83] to detect the driver's face. For the third strategy, they extended the face bounding box to encompass context, allowing the network to learn features associated with the head's position relative to the surrounding environment. Lastly, in the fourth approach, only the upper half of the face was utilized as input. Extracted images were uniformly resized to 224x224 or 227x227 to fulfill the network's requirements.

Figure 4.3: Different cropped regions from the input image used for CNN training, with each cropped region colored for clear distinction. Source [80].

4.4 Neural Networks

The authors performed fine-tuning of four CNNs, originally trained on the ImageNet dataset (further details in Section 4.4.2). The options considered were:

a. AlexNet, proposed by Krizhevsky et al. [84];

b. VGG with 16 layers, introduced by Simonyan et al. [85];

c. ResNet with 50 layers, developed by He et al. [86];

d. SqueezeNet, presented by Iandola et al. [87].

Fine-tuning different networks aims to determine which best suits the task of gaze zone classification, providing valuable insights into architectural aspects such as depth, number of layers, kernel sizes, and model sizes, and their impact on this task.

In terms of network features, AlexNet is a CNN consisting of eight layers, incorporating five convolutional layers followed by two fully connected layers and a softmax layer. The initial convolutional layers utilize large kernels (11×11) with a stride of 5, followed by 5×5 kernels in the second layer and 3×3 kernels in subsequent layers. VGG16 comprises 16 layers, all featuring 3×3 convolutions and 2×2 pooling throughout the network. ResNet introduces skip connections and employs 7×7 convolutions in the initial layer, followed by 3×3 kernels in subsequent layers. SqueezeNet utilizes fire modules, combining 1×1 and 3×3 kernels. Due to its compact size, SqueezeNet is well-suited for implementation on FPGA and embedded systems. Both ResNet50 and SqueezeNet include a global average pooling layer at the end, but SqueezeNet integrates the softmax non-linearity directly after this layer, whereas ResNet50 incorporates a fully connected layer between pooling and softmax layers.

4.4.1 Transfer Learning

Transfer learning is a foundational strategy in machine learning, highlighted in [88], for its ability to leverage previously acquired knowledge in one task to enhance performance in a related task. Put simply, it enables a machine learning model to apply past learning experiences to a new context, leading to more efficient training on new tasks, particularly when dataset sizes are limited.

In the examined work, transfer learning has been successfully utilized to optimize the estimation of driver gaze zones. By employing CNNs pretrained on the ImageNet dataset, the acquired knowledge was effectively transferred to the specific task, notably enhancing the model's performance in estimating driver gaze zones across various scenarios and contexts.

4.4.2 ImageNet

ImageNet stands as one of the largest and most influential image datasets within machine learning and computer vision domains. This extensive database comprises over 14 million images, each meticulously annotated with object indications and bounding boxes delineating their contours. These objects span more than 20,000 distinct classes[1], with some classes, such as "soccer ball" or "strawberry," containing

[1]For the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a subset of 1000 non-overlapping categories is utilized.

hundreds of images each [82].

The significance of ImageNet lies in its role as a standard benchmark for evaluating image recognition algorithms. Its comprehensive image annotations facilitate the development of machine learning models capable of identifying and categorizing diverse objects and visual concepts. The annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC), conducted until 2017, has spurred notable advancements in the field, driving the widespread adoption of deep neural networks, particularly CNNs, for intricate image recognition tasks. Utilizing ImageNet as a foundational dataset has enabled the creation and assessment of sophisticated machine learning models, pivotal in the evolution of methodologies like transfer learning, wherein models pretrained on ImageNet are adapted to address specific computer vision tasks.

4.5 Results and Conclusions

The results outlined in Table 4.2 distinctly reveal two notable trends. Firstly, there is a discernible enhancement in the performance of all three networks when utilizing higher-resolution eye images during both training and testing phases, with optimal performance attained by providing solely the upper half of the face as input. Secondly, SqueezeNet consistently surpasses VGG16, which, in turn, outperforms ResNet50 across all cropped regions of the images. AlexNet exhibits poor performance, particularly when eyes constitute a small portion of the image. The optimal model emerges as a fine-tuned iteration of SqueezeNet, trained on upper half face images, achieving an accuracy of 95.18%, thereby showcasing the generalization capabilities of CNNs. Noteworthy is the subpar performance of AlexNet with face field of view images, attributed to the kernel size and stride in the first layer. Meanwhile, SqueezeNet and VGG16, equipped with smaller kernels, outshine ResNet50, whose slight accuracy decline may be attributed to the large kernel size and network depth.

Architecture	**Half Face**	**Face**	**Face+Context**	**Face Embedded FoV**
AlexNet	88.91	82.08	75.56	62.21
ResNet50	91.66	89.34	86.67	87.04
VGG16	93.36	92.74	91.21	88.92
SqueezeNet	**95.18**	**94.81**	**92.74**	**89.37**

Table 4.2: Ablation experiments with different CNNs and different cropped regions from images, referring to the work done in [80]. The average macro-accuracy obtained for each experiment is tabulated.

Gaze tracking techniques based on classification emerge as more feasible and

suitable for the context of this thesis project compared to the regression method that estimates the gaze vector. Several factors contribute to making classification a more reasonable and manageable choice.

Firstly, classification is less computationally burdensome than the regression method. The ability to estimate discrete directions allows for excellent performance even with only one camera, reducing model complexity and lowering computational load.

Secondly, the flexibility offered by using an integer number of classes, greater than or equal to 7, provides a clearer and more interpretable structure for result analysis. Each class distinctly represents an Area of Interest (AOI), enabling a detailed breakdown of the driver's gaze zones. This breakdown not only enhances the understanding of the information returned by the model but also facilitates more efficient correction of any errors or improvements in prediction accuracy.

Considering these aspects, the choice of gaze classification appears more suitable given the computational limitations, data complexity, and time constraints of the thesis project, while ensuring a valid and reasonable research methodology.

In conclusion, opting for gaze classification proves to be strategically advantageous in balancing effectiveness, manageability, and available resources, while also ensuring a valid and reasonable research methodology.

Chapter 5

Datasets

In Section 3.3, a comprehensive list of primary public datasets dedicated to gaze tracking via gaze vector regression was presented. Building upon this, the chapter aims to further this analysis by offering a comparable list of public datasets specifically tailored for gaze tracking using a gaze classification approach. The objective is to provide a thorough overview of crucial resources in this domain, emphasizing the unique characteristics of each dataset.

Subsequent to this examination, the viability and clarity of one or more identified datasets will be assessed, taking into account the specific requirements and objectives of the thesis project. The goal is to select datasets that can serve as a robust foundation for ongoing research, thereby contributing to the formulation of adopted methodologies and ensuring a precise and meaningful approach within the project's framework.

5.1 Public gaze classification datasets

Numerous open-source datasets are dedicated to analyzing drivers' visual behavior, captured both in stationary vehicles and while in motion in real-world scenarios. These datasets are accessible for download from open-source repositories or can be obtained upon request from the data owners, typically under agreements specifying non-commercial usage.

The following section presents a selection of these datasets along with concise descriptions of their key attributes.

The RS-DMV dataset [89] comprises ten grayscale face videos of drivers, recorded both indoors (simulator) and outdoors on university roads. DriveFace [90] categorizes the driver's head poses into three classes: right, frontal, and left. Brain4Cars [91] offers synchronized data from various sensors, including external and internal car videos, vehicle speed, and GPS coordinates, collected from ten drivers in natural

driving conditions over a period of up to two months. The DriveAHEAD dataset [92] is a head pose dataset incorporating depth and infrared images, utilizing a 3D motion capture sensor to track head position and orientation. The DMD dataset [93] is multimodal, comprising images from three cameras (RGB, IR, Depth) capturing face, body, and hand movements, along with information on head and hand pose and gaze frequency.

Certain datasets, such as LISA GAZE v2 [94], record data in stationary vehicles to mitigate risks associated with actual driving. Others, like DG-UNICAMP [95] and DGW [96], offer a diverse array of RGB, IR, and Depth images, encompassing various challenging lighting conditions. Notably, DGW poses classification challenges due to the presence of intertwined gaze classes.

After evaluating numerous initially considered datasets, only a subset proved suitable for the thesis project. Specifically, preference was given to datasets providing frames with diverse lighting conditions and a wide range of classes. The selection process was guided by a preference for a dataset that is representative of the European population, facilitating potential testing of the final model using frames acquired by us. The DG-UNICAMP dataset [95] was the sole available dataset meeting these criteria.

5.1.1 DG-Unicamp Driver Gaze Zone Dataset

The DG-Unicamp Driver Gaze Zone dataset [95] is a comprehensive repository of images encompassing data in color (RGB), infrared, and depth (RGB-D) modalities. These data were captured utilizing a camera positioned 60 cm away from the driver, situated above the tachometer.

In total, the dataset comprises approximately 230,830 frames, collected from 45 distinct drivers. For each driver, an extensive array of images is accessible, meticulously organized according to the driver's ID, as illustrated in Figure 5.1.

Every image in the dataset corresponds to a text file containing face detection details for a segment of the overall sample. The images are stored in two formats: BMP for color and infrared images, and YAML 1.0 for depth images. They are uncompressed and possess a resolution of 240 x 320 pixels. These images depict the driver focused on 19 key points in the frontal view of the car, as depicted in Figure 5.2, and adhere to a particular naming convention:

[driver_id]_[point]_[camera_type]_[frame_number].[bmp/yaml]

In the examination of the DG-Unicamp image dataset [95], a valid concern emerged regarding the relatively modest resolution of the images, fixed at 240 x 320 pixels. This resolution has the potential to diminish the clarity of crucial features, such as eyes and pupils, thereby posing a risk to the accurate estimation of the

```
Dataset DG-Unicamp
├── p044
│   ├── 44_10_depth_0.yaml
│   ├── 44_10_depth_aligned_0.yaml
│   └── ...
├── p042
│   ├── 42_10_ir1_0.bmp
│   ├── 42_10_ir2_0.bmp
│   └── ...
└── p030
    ├── 30_10_rgb_0.bmp
    └── ...
```

Figure 5.1: Diagram showing how the files are divided in the DG-Unicamp Dataset folder.

class among the 19 possibilities.

5.2 Custom Dataset

To address potential limitations regarding frame resolution in the previous dataset, a strategic decision was made to develop a new dataset with significantly improved resolution. This decision offers the necessary flexibility to adopt an alternative approach should the resolution of the initial dataset prove problematic, thus ensuring the robustness and effectiveness of the overall analysis.

Data acquisition occurred under static conditions within deactivated vehicles, with different times and days designated for each acquisition session. Involving a group of 30 individuals, some wearing glasses and others not, a single camera positioned above the odometer was utilized. This camera, a phone with a resolution of 1080x720 pixels and a frame rate of 30 frames per second (fps), facilitated detailed capture of the vehicle's interior.

Throughout the acquisition phase, drivers directed their focus towards nine specific areas: the left window, encompassing the rearview mirror; the area directly in front of the driver; the rearview mirror and the central front area; the area directly in front of the passenger seat; the right window, including the mirror; the glove compartment area; the radio system; and the odometer. Figure 5.3 distinctly illustrates these focus points, assigning them class numbers ranging from 21 to 29, with the final class representing the driver with closed eyes.

Before participating in the process, informed consent was obtained from the individuals, who graciously authorized the use of their images for the creation of a

Figure 5.2: In Figure A, the key points where the driver gazes during driving are highlighted, based on the DG-Unicamp dataset [95]. Point 19 is not displayed, indicating a moment when the driver has closed eyes. Figure B shows the driver's view, with the camera position above the tachometer. Figure C highlights the approximate distance of about 60 cm between the camera and the driver. Source [95].

dataset intended to train an artificial intelligence algorithm.

During filming, drivers were requested to assume a posture similar to that adopted during actual driving, allowing for natural variations in body positions and movements. For each class, a 30-second video was recorded for each driver. Subsequently, the videos were processed to extract 30 frames per second, thus forming the dataset. Figure 5.4 showcases examples of frames included in the dataset, representing the various classes of interest.

During the data collection process, the extracted frames were saved in .png format and named according to a standardized naming scheme:

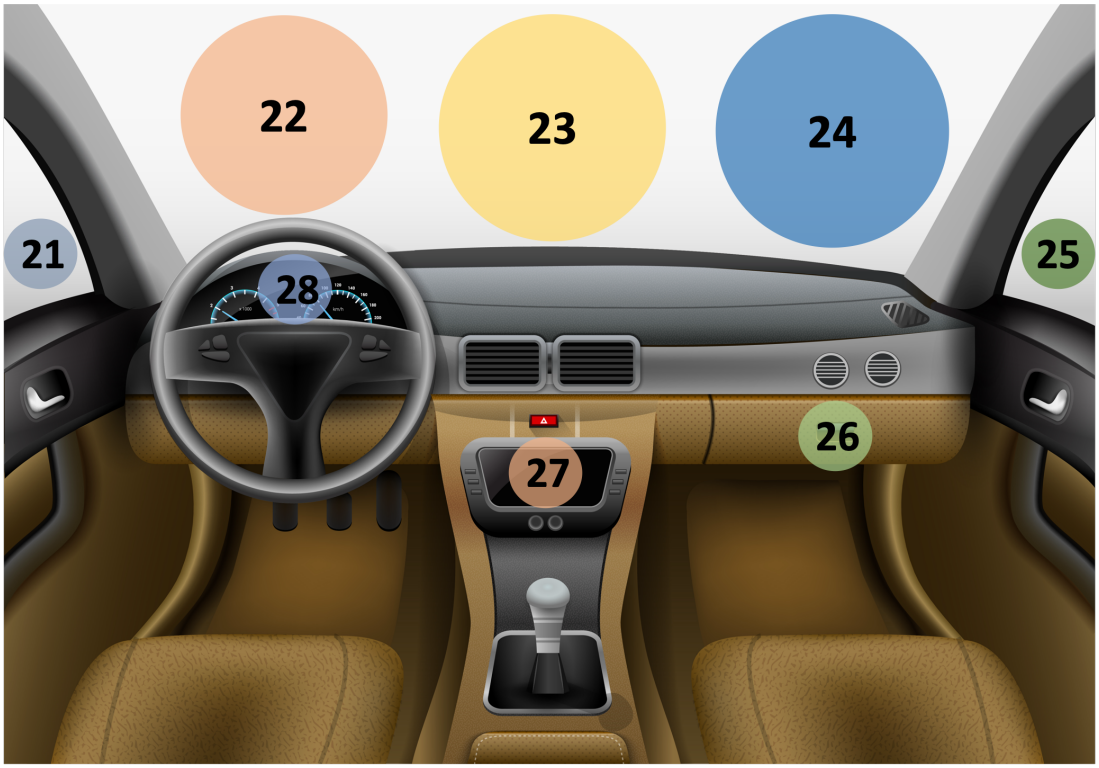


Figure 5.3: The illustration shows a car with overlaid numbers from 21 to 28, highlighting the focus points for the custom dataset. These numbers indicate where drivers gaze during recordings. It is important to note that class 29, corresponding to the condition of closed eyes, is not visible in the image.

[driver_id]_[point]_[camera_type]_[counter]_[frame_number].[png]

The pivotal concept here is the "counter," which assumes a critical role in file naming. This is particularly significant as the 30-second segments of video for each class originate from two distinct videos, each lasting 15 seconds. Consequently, the counter value fluctuates depending on the number of videos (1 or 2) from which the frames are extracted. The presence of the counter prevents overwrites, ensuring that frames extracted from the first video are not overwritten by those from the second. In essence, the counter guarantees uniqueness in identifying each frame, averting confusion and upholding the integrity of the collected data.

In a subsequent phase, every individual file was organized within folders named according to the driver's unique identifier (id_driver). This hierarchical structure furnishes an orderly and efficient approach for cataloging and retrieving data pertaining to each driver during subsequent analysis phases. The layout is outlined as depicted in Figure 5.5.

Figure 5.4: Representative examples of frames in the custom dataset. Letters from A to I correspond to specific classes as follows: A (Class 21), B (Class 22), C (Class 23), and so forth, indicating the numbering of classes associated with each letter in the figure.

```
Dataset_BT
├── p046
│   ├── p046_21_rgb_1_1.png
│   ├── p046_21_rgb_2_1.png
│   └── ...
├── p047
│   ├── p047_21_rgb_1_300.png
│   └── ...
└── p048
    ├── p048_21_rgb_2_400.png
    └── ...
```

Figure 5.5: Diagram showing how the files are divided in the Custom dataset folder.

The complete dataset comprises a total of 190,787 elements. However, it's imperative to underscore that frames associated with a particular driver are absent due to a camera settings error during acquisition. Consequently, the actual total is adjusted to 30 drivers, as delineated in Table 5.1. This table furnishes a

comprehensive breakdown of the number of frames for each class of each driver and, subsequently, the overall count of frames for each driver.

Number of frames

Drivers	**Classes**									**Total**
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	
p046	730	844	812	888	805	734	766	635	843	7039
p047	717	819	697	773	729	772	834	596	825	6744
p048	414	433	402	410	389	396	401	431	436	3703
p049	912	817	840	788	843	894	795	845	810	7526
p050	428	375	421	252	335	161	168	357	245	2733
p052	1006	822	778	713	663	757	273	431	806	6233
p053	557	418	421	294	365	446	336	442	413	3694
p054	1088	801	781	753	779	811	800	737	858	7390
p055	995	616	821	793	565	690	771	743	828	6804
p056	1025	808	810	837	851	800	812	812	762	7499
p057	945	789	846	792	810	800	804	837	920	7525
p058	742	809	828	803	546	834	810	806	812	6972
p059	954	283	562	794	863	563	860	793	840	6495
p060	835	809	850	753	597	777	630	878	829	6931
p061	862	817	809	828	814	816	792	879	768	7368
p062	432	852	814	835	526	701	829	829	741	6541
p063	825	939	817	904	863	849	948	734	906	7767
p064	962	885	952	795	910	860	785	830	877	7808
p065	893	886	828	897	909	884	809	844	903	7835
p066	626	810	613	721	824	778	850	804	881	6889
p067	570	414	403	379	359	409	391	436	424	3776
p068	475	310	405	418	365	416	419	437	407	3643
p069	853	854	694	414	886	904	699	844	844	6975
p070	878	845	804	777	823	576	926	935	792	7328
p071	885	723	855	692	787	830	491	822	846	6913
p072	922	851	822	797	803	564	809	858	848	7256
p073	438	405	346	574	461	354	455	422	440	3886
p074	448	487	394	456	557	469	404	458	456	4120
p075	878	884	852	905	823	865	878	948	846	7861
p076	747	898	795	780	849	842	819	890	931	7533
										190,787

Table 5.1: The table presents the quantity of frames collected for each class of every driver in the custom dataset. It is to be noted that the driver p051 has been excluded and therefore is not represented in the table.

Chapter 6

Methods

6.1 Data Preparation and Data Cleaning

6.1.1 DG-UNICAMP

As outlined in Section 5, this project introduced two distinct datasets aimed at exploring drivers' gaze directions to detect distraction situations. The first dataset, DG-UNICAMP, consists of 19 classes identifying the primary gaze directions. Additionally, a customized second dataset, featuring only 9 classes, was integrated. This decision was driven by the necessity to streamline analysis and reduce computational costs.

In alignment with this choice, the 19 classes of the public dataset were selectively grouped to match the 9 classes of the customized dataset, as depicted in Figure 6.1. This approach seeks to optimize analysis efficiency by focusing on key behaviors relevant to the context of driving distraction.

Following the class reorganization, a thorough examination of the dataset was carried out, eliminating frames characterized by extremely low brightness. The metric employed for this task was the Integrated Optical Density (IOD), closely linked to the concept of brightness. To clarify, in an 8-bit image (MxN) where all pixels are black, the IOD takes a value of 0. Conversely, if all pixels are white, the IOD reaches $(2^8 - 1) \times N \times M$, as demonstrated in Equation 6.1. Determining the threshold value below which frames were excluded involved a detailed analysis of the images, leading to the selection of an optimal value equal to $54 \times N \times M$, considering M=240 and N=320. This decision resulted in a notable reduction in dataset dimensions. Moreover, to ensure the utilization of all images in the dataset during the training phase, preprocessing (as discussed in Section 6.3.1) was conducted in advance on all frames. This procedure facilitated the elimination of frames where the Mediapipe framework failed to detect the driver's face. Following these two data cleaning processes, entire drivers were completely removed, while in

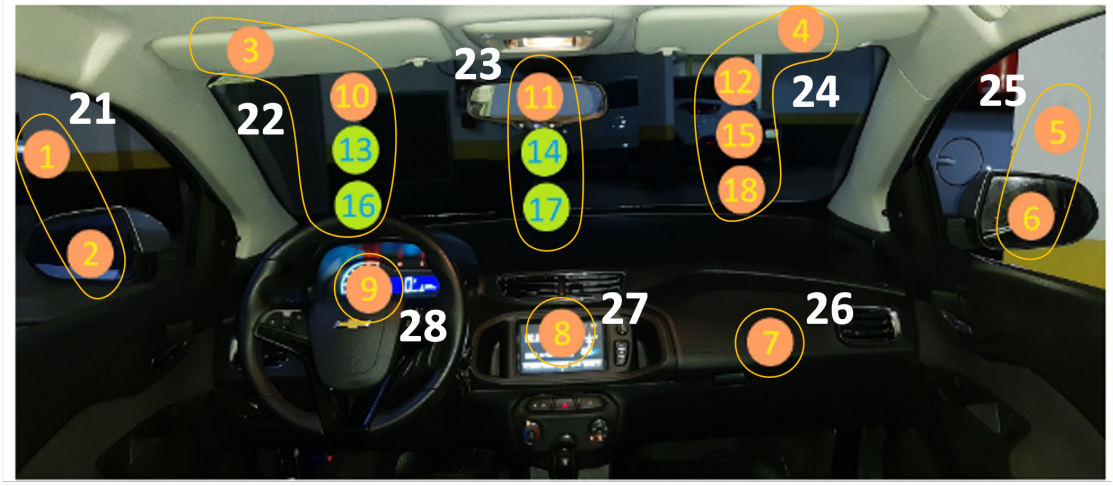


Figure 6.1: The figure clearly depicts the subdivision of the 19 classes of the DG-UNICAMP dataset, distinguishing them with green and orange colors. The original divisions have been merged via yellow curves, assigning them a new class. Precisely, the resulting new classes are 9, ranging from 21 to 29. Note that classes 19 and 29 are excluded, as they represent the driver's state with closed eyes.

some cases, only specific classes were entirely eliminated.

$$IOD = \int_x \int_y a(x,y)dxdy = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} b[m,n] \tag{6.1}$$

6.1.2 Custom Dataset

The custom dataset, as detailed in Section 5.2, was internally generated within the company during the thesis research, incorporating several optimizations compared to the DG-UNICAMP dataset. Specifically, images were exclusively captured during morning and afternoon hours, thereby circumventing the presence of entirely black frames encountered in the DG-UNICAMP dataset due to nighttime acquisitions. Divided into 9 classes, the custom dataset obviates the necessity for further grouping. Consistent with the approach employed during the cleaning analysis of DG-UNICAMP, preemptive preprocessing was executed to discard any frames where the Mediapipe framework fails to detect the driver's face. For these reasons, unlike the public dataset, situations where data for entire drivers or parts of them are missing do not occur in this case.

6.2 Dataset Division

The division of datasets into training, validation, and test sets is a fundamental practice in the field of machine learning. This approach aims to objectively evaluate the performance of a model while ensuring its ability to generalize well to unseen data. Each division serves a distinct role in the model development process:

1. Training Set: This subset of the dataset is used to train the model. During this phase, the model learns the patterns and relationships present in the data, optimizing its parameters to fit the training set.

2. Validation Set: Following training, the model is assessed on a separate dataset known as the validation set. This set aids in tuning the model's parameters and selecting the best architecture, preventing overfitting to the training set. Validation helps optimize the model's performance without compromising its generalization ability.

3. Test Set: Once the model has been trained and optimized based on the validation set, it is evaluated on a completely independent test set. This set consists of data that the model has not encountered during training or validation. By assessing the model's performance on this set, an accurate estimation of its real generalization capabilities can be obtained.

In the specific context of datasets comprising drivers, it is crucial to conduct a data split based on individual drivers. This split aims to mitigate the risk of the model becoming overly specialized on a single driver, instead fostering its capacity to generalize effectively to unknown drivers. Dividing the data by drivers enables an assessment of the model's ability to handle diverse driving styles while maintaining high performance across different individuals. This approach enhances confidence in the model's generalization in real-world scenarios, free from the constraints of specific idiosyncrasies associated with a single driver.

The two datasets have been divided as illustrated in Tables 6.1 and 6.2.

Dataset DG-UNICAMP (9-classes)	
	# Drivers
Training set	26
Validation set	11
Test set	4

Table 6.1: Summary table showing the number of drivers present in each dataset generated from the DG-UNICAMP dataset for training the 9-class models.

Custom Dataset (9-classes)	
	# **Drivers**
Training set	24
Validation set	3
Test set	3

Table 6.2: Summary table showing the number of drivers present in each dataset generated from the custom dataset for training the 9-class models.

6.2.1 DG-UNICAMP

The DG-UNICAMP dataset boasts a substantial number of frames, which, if processed by an artificial intelligence model, would incur excessively high computational times and costs. Consequently, it was deemed necessary to proceed with reducing the dataset's dimensions.

To ensure acceptable computational times, 10 frames were extracted from each of the 19 original classes for every driver. However, this selection inevitably led to an imbalance in the new dataset. As depicted in Table 6.3, which illustrates the number of frames for each class concerning each driver in the test set, there is a noticeable overrepresentation in classes 22 and 24. This phenomenon arises from these classes resulting from more frequent mergers compared to others, as elucidated in Figure 6.1. Additionally, it's notable that some drivers lack frames entirely for certain class types due to reasons elaborated in Section 6.1.1.

To rectify this imbalance, a two-phase balancing process was implemented:

1. Initially, the number of frames extracted from each class for every driver was increased to 40. This adjustment aligns with the quantity of frames extracted for class 22, which is the most represented class (calculated by multiplying 10 frames per class by 4 merged classes).

2. Subsequently, cases where some classes had no available frames for certain drivers (0 frames) were addressed. For each class, the number of frames for other drivers was augmented to equalize the total number of frames with that of the most represented class.

These adjustments, implemented uniformly across both the Training and Validation sets, are clearly illustrated in Table 6.4. The table displays the number of frames for each class relative to each driver in the Test set following the balancing process.

The total number of frames for each dataset is reported in Table 6.5.

Test set (DG-UNICAMP) - after									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
p029	20	40	30	40	20	10	10	10	10
p024	20	40	30	40	20	10	10	10	0
p019	20	40	30	40	20	10	10	10	10
p013	20	40	0	40	20	10	0	0	0

Table 6.3: The table displays the number of frames for each class in relation to each driver in the test set of the DG-UNICAMP dataset. Please note that the table reflects the original distribution, before any balancing procedures were applied.

Test set (DG-UNICAMP) - before									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
p029	40	40	53	40	40	40	53	53	80
p024	40	40	53	40	40	40	53	53	0
p019	40	40	53	40	40	40	53	53	80
p013	40	40	0	40	40	40	0	0	0

Table 6.4: The table shows the number of frames for each class relative to each driver in the test set of the DG-UNICAMP dataset. Please note that the table represents the data distribution after the balancing procedure has been applied.

Dataset DG-UNICAMP (9-classes)	
	# **Frames**
Training set	9407
Validation set	3960
Test set	1446

Table 6.5: Summary table showing the number of frames present in each dataset generated from the DG-UNICAMP dataset for training the 9-class models..

6.2.2 Custom Dataset

The overall frame count in the custom dataset is substantial, prompting a selection process to trim it down to 40 frames per class for each driver, as detailed in Table 6.6. This selection involved organizing the frames chronologically for each driver within the same class and extracting those that were farthest apart. Such an approach guarantees the neural network's training with diverse images, facilitating effective generalization by exploring all possible combinations within specific classes.

Test set (dataset personalizzato)									
	21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
p053	40	40	40	40	40	40	40	40	40
p055	40	40	40	40	40	40	40	40	40
p065	40	40	40	40	40	40	40	40	40

Table 6.6: The table displays the number of frames for each class relative to each driver in the test set of the custom dataset. Please note that the table represents the data distribution after the dataset reduction procedure has been applied.

The total number of frames for each dataset is reported in Table 6.7.

6.2.3 Dataset for Bi-Class Models

For the datasets used in the bi-class models, distinguishing classes 23/24 and 26/27, a driver-based subdivision was also adopted. The same drivers present in the just divided datasets, both for DG-UNICAMP and the custom dataset, were retained, selecting only the frames related to the two classes to be classified.

To ensure a total frame count comparable to that of the datasets for the 9-class models shown in Tables 6.5 and 6.7, a greater number of frames were extracted for each of the 2 classes for every driver. Specifically, 180 frames were extracted for each of the 2 classes for every driver. However, the final result (see Tables 6.8 and 6.9) differs slightly from that shown in Tables 6.5 and 6.7, as the total number

Dataset personalizzato (9-classes)	
	# **Frames**
Training set	8640
Validation set	1080
Test set	1080

Table 6.7: Summary table showing the number of frames present in each dataset generated from the custom dataset for training the 9-class models.

of extracted frames is lower than necessary due to the absence of some frames required for a specific class in the original datasets.

DG-UNICAMP (2-classes)	
	# **Frames**
Training set	9200
Validation set	3852
Test set	1356

Table 6.8: Summary table showing the number of frames present in each dataset generated from the DG-UNICAMP dataset for training the bi-class models.

6.3 Pipeline

As depicted in Figures 6.2 A and B, two distinct pipelines were followed during the thesis project, both sharing some common characteristics.

In both pipelines, the primary objective is to estimate the direction towards which the driver is looking among the 9 possible directions, as outlined in Section 6.1.1. To achieve this goal, it is necessary to train artificial intelligence models capable of discriminating between the possible classes when given an input image.

When training CNNs using images as input, it is often essential to perform a pre-processing phase to make the images compatible with the model in question. Therefore, the first phase, pre-processing, is shared identically by both pipelines. Subsequently, the images are inputted into the model, which processes them by estimating the learned class membership during the training phase. It is in this phase that the two pipelines differentiate. In the first pipeline, a single neural network is trained to estimate the 9 main directions, as shown in Figure 6.2 A. In the second pipeline, however, 3 distinct neural networks are trained: the first is tasked with estimating 7 out of 9 gaze directions, merging classes 23/24 and 26/27. Subsequently, if the first network identifies an image's class as one of the merged

Custom Dataset (2-classes)	
	# **Frames**
Training set	8607
Validation set	1080
Test set	1080

Table 6.9: Summary table displaying the number of frames present in each dataset generated from the custom dataset for training the bi-class models.

classes, two additional binary neural networks are activated, specifically dedicated to discriminating classes 24/25 and 26/27, as illustrated in Figure 6.2 B.

The decision to implement the second pipeline was derived from an analysis of various studies in the literature, such as [80]. During this analysis, it emerged that many of these works do not consider classes 24 and 27. By implementing only pipeline A, there could be estimation errors by the model, which might confuse class 24 and 27 with nearby and very similar classes, namely 23 and 26 respectively. This additional approach could ensure more precise and reliable classification, especially in situations where the distinctions between certain classes might be blurred or unclear.

6.3.1 Pre-processing

Three distinct approaches were adopted for the pre-processing of inputs to the CNN, aiming to assess the impact on the model's performance. In the first approach (Figure 6.3 A, D), the original image was used, thus preserving all the surrounding context. This choice aims to understand if the network can train effectively and achieve good performance without the need to preprocess the image before inputting it into the CNN. In the second approach (Figure 6.3 B, E), the driver's face crop was employed using Mediapipe [97]. This decision is motivated by the desire to focus the network's attention exclusively on the facial details, deemed crucial for understanding the gaze direction. In the third approach (Figure 6.3 C, F), only the upper part of the driver's face was used, also obtained through Mediapipe [97], and subsequently extracting 60% of the window from the top. This choice was inspired by [80], where they demonstrate that this approach leads to superior performance compared to any other type of pre-processing. All extracted images were uniformly resized to 224x224, in accordance with the network's requirements, as illustrated in Figure 6.4.

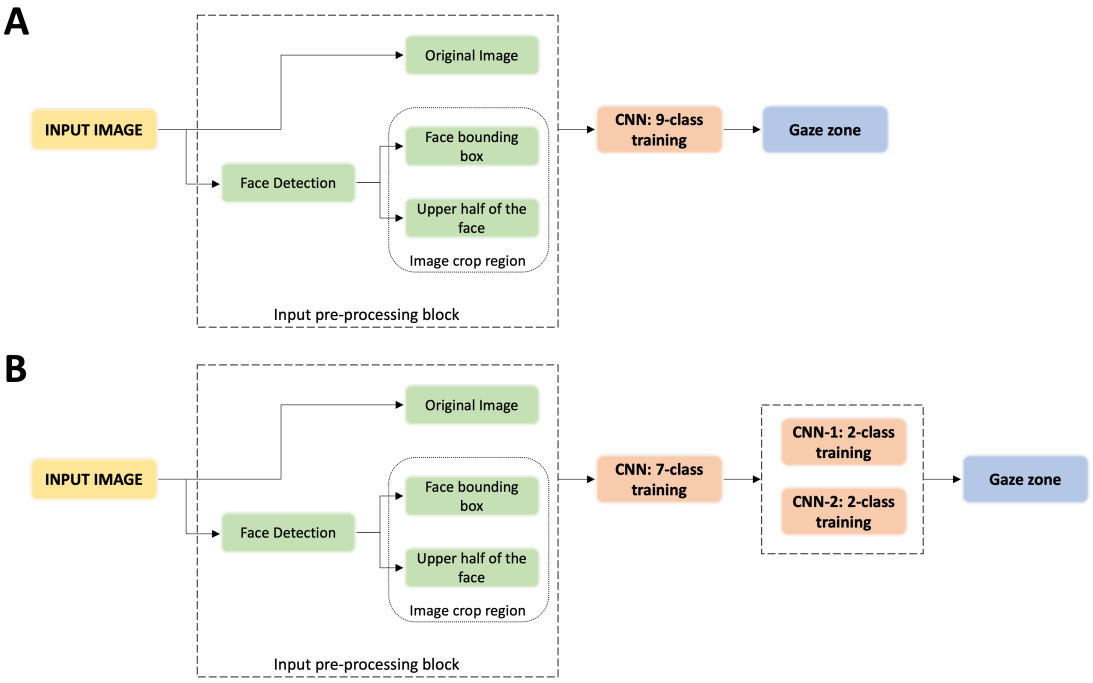


Figure 6.2: Illustration of the two pipelines followed in the thesis project. Pipeline A involves the use of a single neural network to estimate the 9 main gaze directions, while Pipeline B utilizes three distinct neural networks: the first to estimate 7 out of 9 directions, and the other two to specifically discriminate the merged classes 23/24 and 26/27.

Figure 6.3: The figure illustrates various pre-processing methodologies applied to both images from the custom dataset (A, B, C) and those from the DG-UNICAMP dataset (D, E, F), both intended for the CNN. The input images from the custom dataset maintain a resolution of 1280x720 pixels, while those from DG-UNICAMP are 240x320 pixels. Images A and D represent the originals without any pre-processing; images B and E feature the face cropped using Mediapipe [97] (see Section 6.3.1), while images C and F show a type three pre-processing, with the upper face area cropped by 60%. Subsequently, all images are uniformly resized to 224x224 pixels. The difference in resolution between the images from the custom dataset and those from the DG-UNICAMP dataset is evident following pre-processing.

Face Detection

The face detection phase constitutes the initial crucial step in implementing preprocessing of types 2 and 3. In scientific literature, one widely utilized library for this purpose is Dlib. This library relies on the classic Histogram of Oriented Gradients (HOG) method, coupled with a linear classifier, an image pyramid, and a sliding window detection scheme. Such object detectors are generally versatile and capable of identifying various types of semi-rigid objects, including human faces. Indeed, Dlib offers the capability to estimate 68 face landmarks, as clearly highlighted in Figure 6.5 B. This feature proves fundamental in the image processing phase.

However, for this project's context, a new technology has been selected, namely the Mediapipe framework developed by Google. Mediapipe employs a lightweight model for detecting single or multiple faces in selfie-like images from smartphone

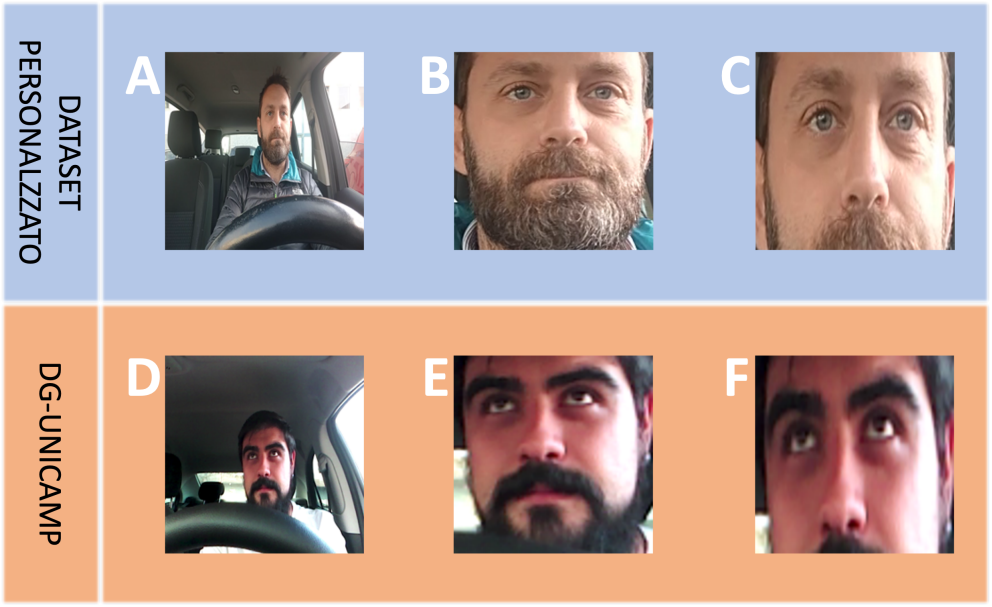


Figure 6.4: The figure represents images from the custom dataset (A, B, C) and the DG-UNICAMP dataset (D, E, F) after the resizing process.

cameras or webcams. This model is optimized for images captured by the phone's front-facing camera at close range. The model architecture is based on a convolutional neural network technique called Single Shot Detector (SSD) [97], integrated with a custom encoder.

Unlike Dlib, the Mediapipe framework is capable of estimating 468 face landmarks, as clearly illustrated in Figure 6.5 A. This feature represents a significant increase compared to Dlib's 68 landmarks, allowing for extremely precise and detailed results during the face recognition phase. The vast number of landmarks provided by Mediapipe allows for an even richer and more detailed representation of facial anatomical variations, enhancing the face detection system's sensitivity and accuracy.

The visual comparison in Figure 6.5 clearly emphasizes that, unlike Dlib, Mediapipe not only outlines the main features of the face but also provides a more detailed estimation by incorporating additional "specific" or "internal" landmarks. Thus, Mediapipe surpasses simple external face delineation by extending to the identification of specific points within the facial structure.

The inclusion of more "specific" landmarks offers a richer and more detailed anatomical representation of the driver's face. This aspect holds particular significance in advanced face recognition scenarios as it enables the capture and analysis of unique facial features with greater precision, thereby enhancing the overall accuracy of the implemented face recognition system.

Figure 6.5: Figure A depicts the image of the driver in the car after processing with Mediapipe, while Figure B shows the image obtained with Dlib.

In essence, Mediapipe emerges as a distinct solution due to its capacity to furnish more comprehensive information through landmark estimation, thereby making a substantial contribution in the image processing phase and within the specific context of the ongoing project.

6.3.2 Neural Networks

Network Finetuning Block

For both pipelines, an approach based on five different neural networks was adopted, initially trained on the ImageNet dataset [82]. The options considered include:

- ResNet18
- EfficientNet B0
- SqueezeNet 1.1
- VGG16
- MobileNet v2

Each of these neural networks was subsequently subjected to a fine-tuning process. This means that instead of training them from scratch, their weights, pretrained on ImageNet, were adapted to the specific task of the thesis project:

predicting the direction in which the driver is looking, classifying among nine possible categories.

The five networks exhibit different characteristics. ResNet18 [98] employs a deep residual architecture to facilitate deep learning. EfficientNet B0 [99] stands out for its optimal balance between depth, width, and resolution. SqueezeNet 1.1 [87] features a compact architecture, while VGG16 [100] adopts a more traditional network structure with numerous layers. MobileNet v2 [101] is known for its computational efficiency, making it ideal for devices with limited resources. Table 6.10 illustrates the main characteristics of the networks, including the advantages and disadvantages of each.

The use of these different neural networks aims to explore and evaluate the performance of each in relation to the specificity of the driver's gaze direction prediction problem. This versatile approach allows for comparing and selecting the network that best suits the study's needs, taking into account the diverse characteristics and complexities of each architecture.

Training

For the first presented pipeline, across all explored architectures, it was crucial to replace the last layer of the network, initially comprised of 1000 neurons, by introducing a fully connected layer consisting of 9 neurons, followed by the addition of a softmax layer. The training phase was divided into two distinct parts. Initially, only the newly added layer underwent specific training for a predetermined number of epochs, aimed at instructing it to classify the 9 classes. Throughout this phase, the network's layers, except for the newly added one, remained frozen. Subsequently, the model underwent further training, with all the network weights, including those of the newly inserted layer, being unfrozen. Figures 6.6 A and B depict the two distinct training phases.

The learning rate for the initial training phase was set to 1×10^{-4} for all examined networks, closely monitoring both loss and accuracy throughout training and validation phases. Whenever fluctuations in the loss function were observed, the learning rate was automatically decreased.

During the second training phase, a lower learning rate of 1×10^{-5} was set for all networks, with continuous observation of loss and accuracy during both training and validation. Similar to the preceding phase, any fluctuations in the loss function led to an automatic reduction in the learning rate.

However, an exception was made for the two-class classification models. For the initial part, the learning rate was set to 1×10^{-5} and for the subsequent part, it was lowered to 1×10^{-6}. This decision was influenced by the fact that the training started from the 7-class model, which already had proficient recognition capabilities for faces and primary orientations.

Architecture	Structure	Parameters	Advantage	Disadvantage
Resnet 18	Feed-forward network with residual connection	11.4 M	High depth without vanishing gradient	Higher computational complexity
VGG16	Sequential structure with very small convolution filters (3x3)	134.7 M	Simple and interpretable	Very heavy in terms of computation
EfficientNet B0	Network scaled according to the compound scaling technique	5.3M	Optimized for depth, width, and resolution	May require higher training resources
MobileNet v2	Network based on an inverted residual structure	4M	Executable on mobile and embedded devices with limited computational resources. Much faster than traditional convolutional neural networks	Optimized for specific tasks. Lower accuracy due to high speed and efficiency. Suitable for limited datasets
SqueezeNet 1.1	Compact structure thanks to the use of fire modules	1M	Small number of parameters	Possible loss of accuracy due to compression

Table 6.10: Summary table of the 5 convolutional neural networks (CNNs) used in the thesis project. The table provides an overview of the structure of each network, the number of parameters used, and their respective advantages and disadvantages.

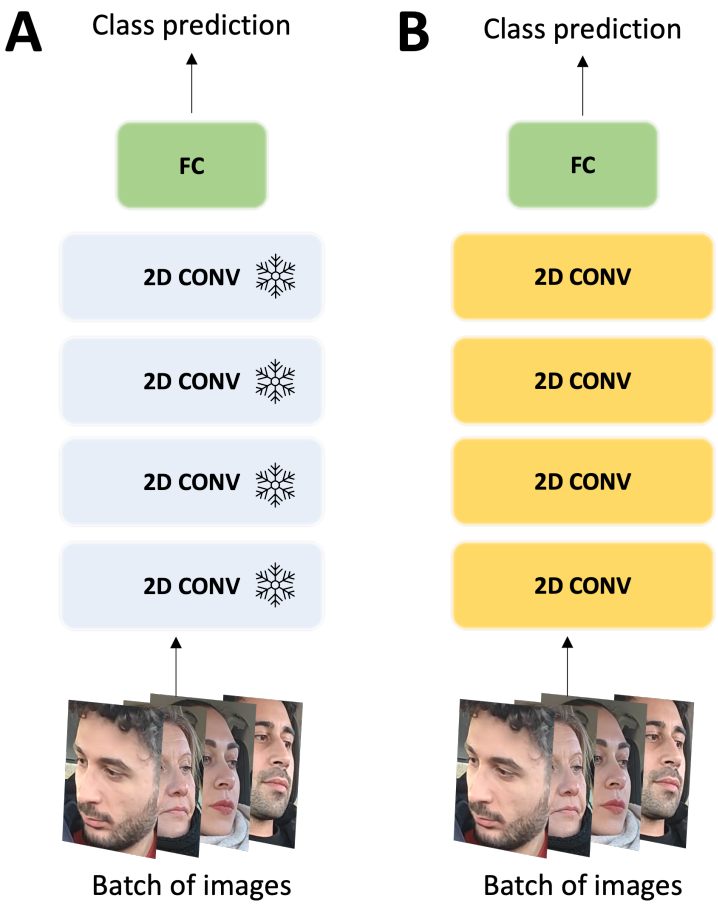


Figure 6.6: Both figures depict the configuration of a convolutional neural network. Figure A is designed to outline the initial phase of training, characterized by the complete freezing of all the weights of the network except for the last layer dedicated to classification. Instead, Figure B is intended to illustrate the subsequent phase of training, in which all the weights of the network are unfrozen.

The adopted loss function is the Cross-Entropy Loss [102], widely utilized in literature to address both multi-class and binary classification tasks.

All networks underwent a fine-tuning process, lasting 10 epochs for the first training and 20 epochs for the second, using the mini-batch gradient descent method with adaptive learning rates. Batch sizes were determined to respect GPU memory constraints, resulting in batch sizes of 64 and 32 respectively. The Adam optimization algorithm introduced by Kingma and Ba [103] was employed. Data augmentation techniques, such as image flipping or rotation, were not applied to avoid potential label alterations or the generation of unrealistic images for guiding

purposes. Pixel intensity modification was excluded since the dataset already exhibited a wide variety in lighting conditions. All experiments were conducted using PyTorch [104] on computational resources provided by Google Colab [105], specifically utilizing a Tesla T4 GPU.

In the context of the second pipeline, where three distinct neural networks are trained, the training process is identical to the one described earlier. The main difference lies in the number of neurons in the added layer: in the first neural network, they become 7, while in the other two, they are set to 2. It is important to highlight that the binary-class networks underwent a fine-tuning process starting from the networks previously trained on the seven classes.

Detailed information regarding the network training parameters is provided in Table 6.11. Any exceptions will be addressed in Chapter 7, dedicated to the results.

Network training parameters		
Loss function		Cross Entropy
Optimizer		Adam
Batch		64 - 32
	First Train	**Second Train**
# Epoch	10	20
Learning rate	1×10^{-4} / 1×10^{-5}	1×10^{-5} / 1×10^{-6}

Table 6.11: Summary of network parameters

Evaluation Metrics

The evaluation of the experiments presented in the previous sections primarily relied on two metrics.

Initially, a confusion matrix was generated based on the individual models' results on the test set. Subsequently, the Macro-average accuracy was computed to provide an overall assessment of the models' performance, as described by the formula:

$$\text{Macro-average accuracy} = \frac{1}{N} \sum_{i=1}^{N} \frac{(True\ \ Positive)_i}{(Total\ \ Population)_i} \tag{6.2}$$

where, N is the total number of classes.

The second metric adopted is the F1-score, which requires separate calculation for each class. To achieve this, the multi-class confusion matrix is divided as illustrated in Figure 6.7. Once the values of True Positives (TP), True Negatives (TN), False Negatives (FN), and False Positives (FP) are identified, the F1-score can be computed. This metric represents the harmonic mean between precision

and recall, assigning equal weight to both false positives and false negatives, and is calculated according to the formula:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6.3}$$

Where:

- Precision is calculated as $\frac{TP}{TP+FP}$.
- Recall is calculated as $\frac{TP}{TP+FN}$.

The F1-score is considered a secondary but crucial measure for a detailed evaluation of the model's performance. Specifically, when the overall accuracy among models is similar, the F1-score of individual classes becomes essential for guiding the selection of the best model.

The same evaluation methods apply to binary-class confusion matrices as well. In this case, a single accuracy value will be obtained, accompanied by a single F1-score value.

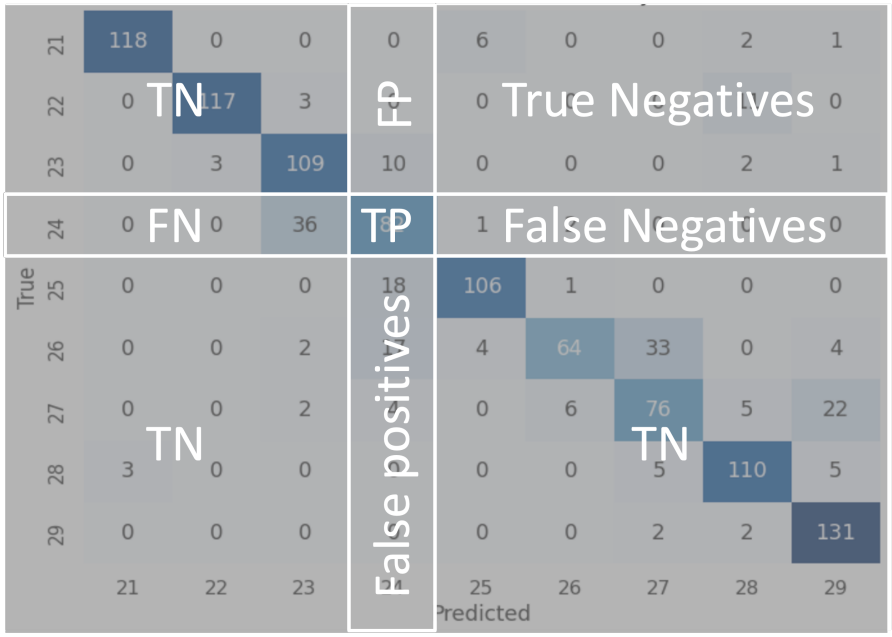


Figure 6.7: Multi-class confusion matrix

Chapter 7

Results

7.1 Analysis of Datasets

In the scope of this preliminary analysis, two datasets were evaluated, DG-Unicamp and a custom dataset, through the training of a convolutional neural network ResNet-18. The performance of the generated models was assessed using three different image preprocessing approaches: original image, face cropping, and only half of the face. As depicted in Tables 7.1 and 7.2, a notable performance difference between the two datasets is observed.

DG-UNICAMP			
Architecture	**Original Image**	**Face**	**Half Face**
Resnet 18	0.54	0.63	0.66

Table 7.1: Table showing the results of micro-average accuracy on the test set of the DG-Unicamp dataset, evaluated through three different image preprocessing approaches.

Dataset Personalizzato			
Architecture	**Original Image**	**Face**	**Half Face**
Resnet 18	0.74	0.86	0.87

Table 7.2: Table showing the results of micro-average accuracy on the test set of the custom dataset, evaluated through three different image preprocessing approaches.

In particular, training the ResNet-18 model on the DG-Unicamp dataset resulted in overfitting during both the initial and subsequent training phases, as highlighted

in Figures 7.1 A and B. This outcome is likely attributable to the low resolution of the images in this dataset. For instance, during face cropping, the resulting images have significantly low resolutions, typically around 82x82 pixels. Since the neural network requires input images with a size of 224x224 pixels, the resizing process via linear interpolation leads to a considerable loss of details, clearly indicating a limitation in the use of this dataset.

Consequently, it can be concluded that the use of the DG-Unicamp dataset should be discarded for future work steps, as its low resolution and the consequent loss of details significantly compromise the performance of the neural network.

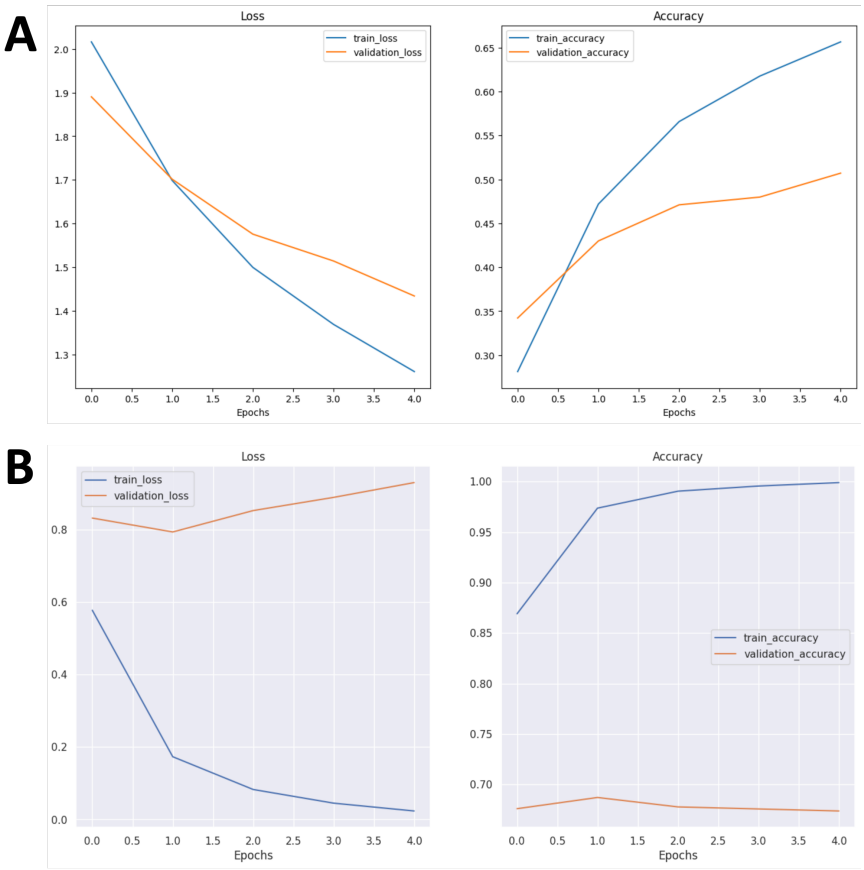


Figure 7.1: The figures display the trend of cross-entropy loss and accuracy during the training process of the ResNet-18 model on the DG-Unicamp dataset, utilizing the *half face* preprocessing. Figure A shows the results of the first training phase, while Figure B shows those of the second phase, following the unfreezing of all model weights.

7.2 Analysis of Network Architectures and Different Image Cropping Regions

7.2.1 Pipeline A

The Table 7.3 presents the Macro-average accuracy obtained on the test set for 15 different combinations of neural networks and image cropping regions for the 9-class classification task. Two clear trends emerge from the table. Firstly, it's evident that all networks, except for MobileNet v2, achieve better results when fed with only the upper half of the face. Secondly, the ResNet18 and EfficientNet B0 architectures consistently outperform VGG16, which, in turn, outperforms SqueezeNet 1.1 across all different image cropping regions. The only exception is observed in training with MobileNet v2 using face cropping, where an accuracy of 0.90 is reached, surpassing the maximum value obtained with ResNet18 and EfficientNet B0 using only the upper half of the face as pre-processing.

Architecture	Original Image	Face	Half Face
Resnet 18	0.74	0.86	**0.87**
EfficientNet B0	0.79	0.86	**0.87**
SqueezeNet 1.1	0.67	0.81	0.82
VGG 16	0.79	0.83	0.85
MobileNet v2	0.68	**0.90**	0.86

Table 7.3: Experiments of ablation conducted using various CNNs and various image cropping regions for the 9-class classification task. The macro-average accuracy obtained for each experiment is reported in tabular form.

In Figures 7.2, 7.3, and 7.4, the confusion matrices, F1-score values, computation time in milliseconds on GPU, and heatmaps are illustrated for the EfficientNet B0 and ResNet18 models, using the upper half of the face as input images. Conversely, in Figures 7.5 and 7.6, the same graphs are presented for MobileNet v2, using the entire face as input images.

It is specified that the time in milliseconds refers to the average of individual inferences over the entire test set.

The heatmaps provide a visualization of activation patterns in the last convolutional layer of the model, allowing to identify the areas of the image that had the greatest impact on the model's classification decisions.

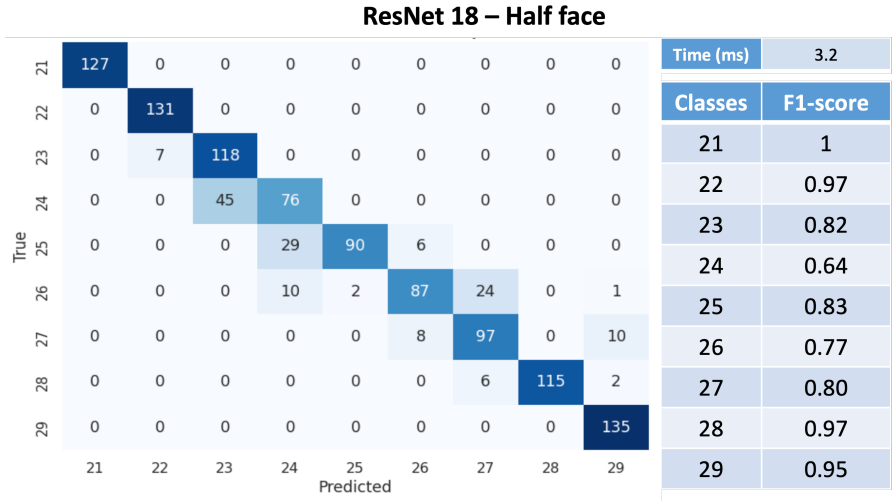


Figure 7.2: The figure depicts the 9x9 confusion matrix for the ResNet18 model with half-face preprocessing. On the right side of the figure, the corresponding F1-score values for each class are reported.

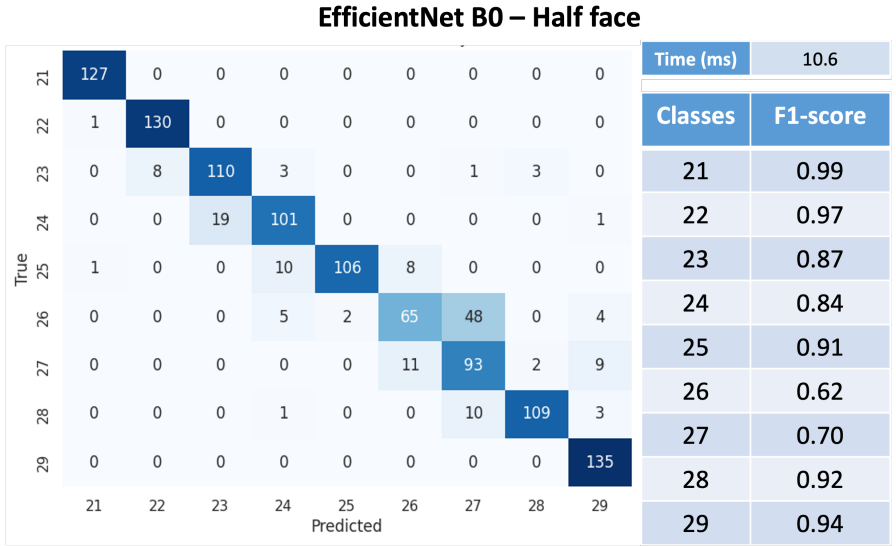


Figure 7.3: The figure displays the 9x9 confusion matrix for the EfficientNet B0 model with half-face crop preprocessing. On the right side of the figure, the corresponding F1-score values for each class are reported.

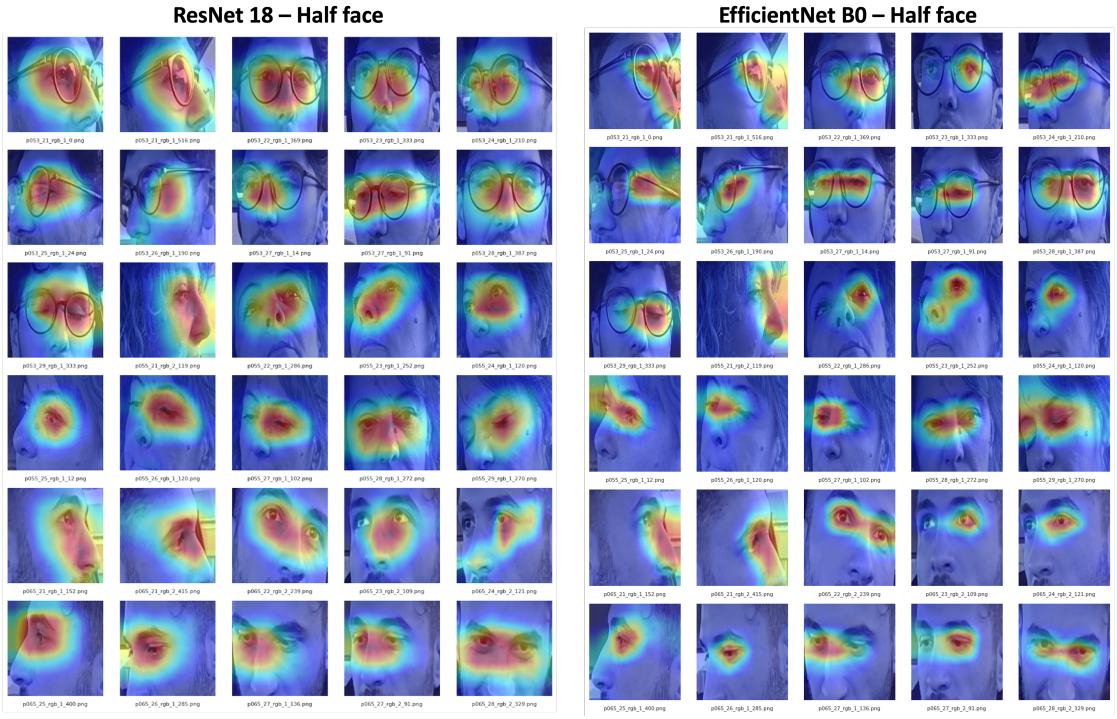
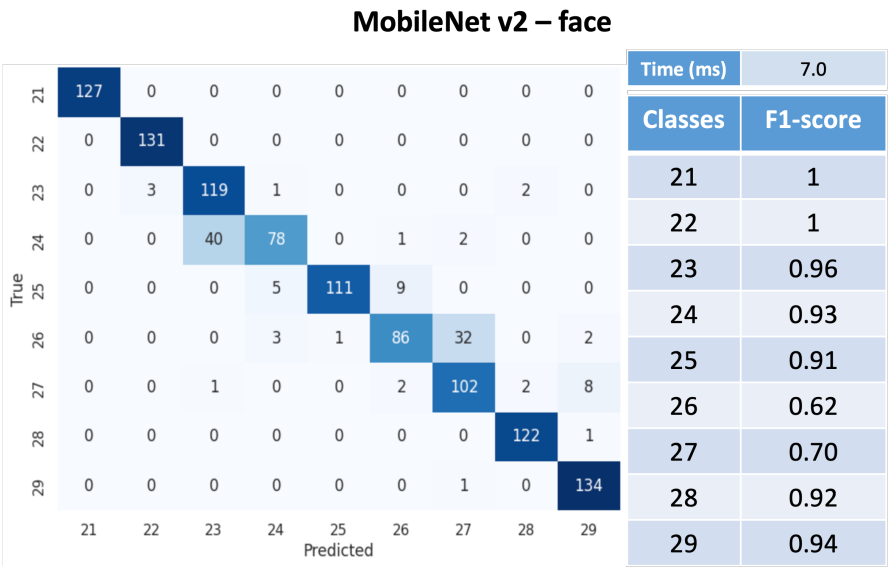


Figure 7.4: The figure illustrates the comparison between the heatmaps generated by the ResNet18 and EfficientNet B0 models (baseline performance in Figure 7.3) using 30 frames from the test set.



Time (ms)	7.0
Classes	**F1-score**
21	1
22	1
23	0.96
24	0.93
25	0.91
26	0.62
27	0.70
28	0.92
29	0.94

Figure 7.5: The figure presents the 9x9 confusion matrix for the MobileNet v2 model with face crop preprocessing. On the right side of the figure, the corresponding F1-score values for each class are reported.

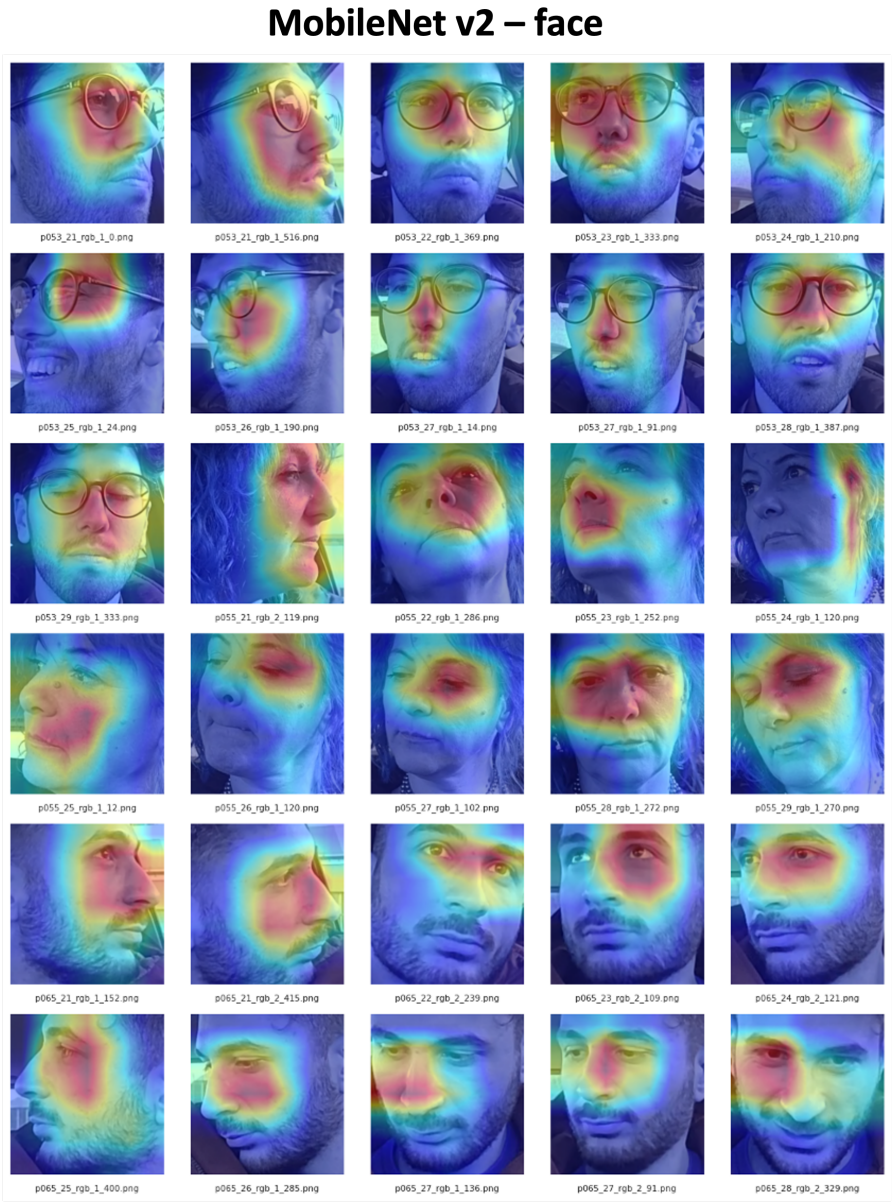


Figure 7.6: The figure displays the heatmaps generated by the MobileNet v2 model (baseline performance in Figure 7.5) using 30 frames from the test set.

7.2.2 Pipeline B

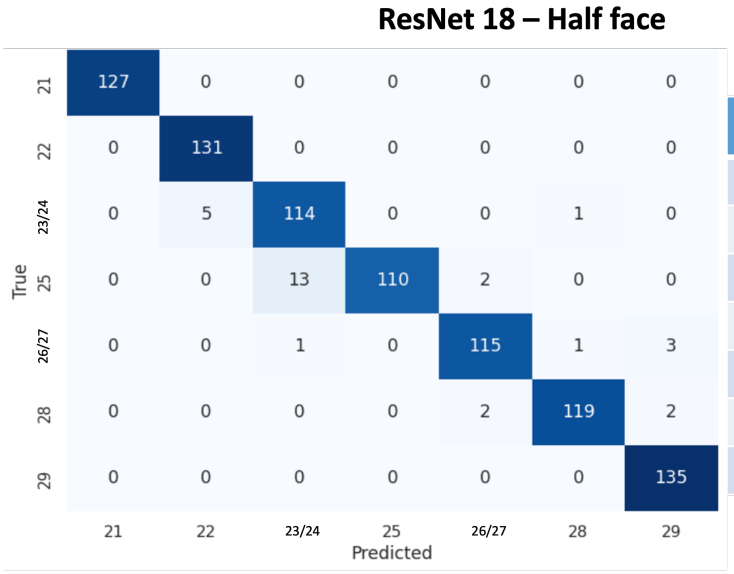
7-class Model

The Table 7.4 presents the Macro-average accuracy obtained on the test set for 15 different combinations of neural networks and image cropping regions for the 7-class classification task. Clear improvements in performance are observed compared to those obtained during inference on the same test set for the recognition of 9 classes. In particular, it is noted that ResNet18 with half face crop preprocessing records a percentage increase in performance of 11.5%, reaching an accuracy of 97%. This result highlights the usefulness of merging classes 23/24 and 26/27, emphasizing how the 9-class model achieves lower performance precisely due to the confusion in classification between these classes. Also noteworthy are the performances obtained by MobileNet v2 with face crop preprocessing in this case.

Architecture	**Original Image**	**Face**	**Half Face**
Resnet 18	0.82	0.93	**0.97**
EfficientNet B0	0.85	0.94	0.92
SqueezeNet 1.1	0.72	0.93	0.91
VGG 16	0.79	0.89	0.92
MobileNet v2	0.76	**0.96**	0.94

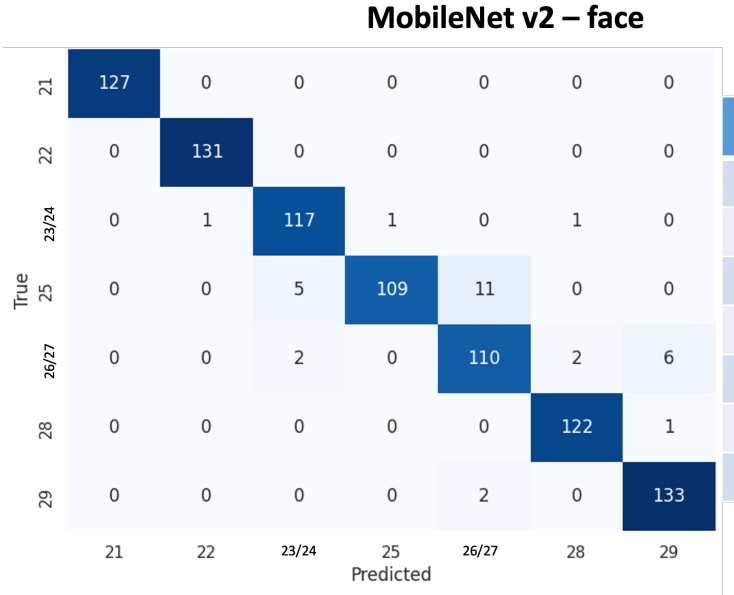
Table 7.4: Ablation experiments were conducted using various CNNs and different image cropping regions for the 7-class classification task. The macro-average accuracy obtained for each experiment is reported in tabular form.

In Figures 7.7 and 7.9, the confusion matrices, F1-score values, and heatmaps for the ResNet18 model are illustrated, using the upper half of the face as input images. Conversely, in Figures 7.8 and 7.10, the same graphs are presented for MobileNet v2, using the entire face as input images.



Classes	F1-score
21	1
22	0.98
23/24	0.92
25	0.93
26/27	0.96
28	0.98
29	0.98

Figure 7.7: The figure displays the 7x7 confusion matrix for the ResNet-18 model with half-face crop preprocessing. On the right side of the figure, the corresponding F1-score values for each class are reported.



Classes	F1-score
21	1
22	1
23/24	0.96
25	0.93
26/27	0.91
28	0.98
29	0.97

Figure 7.8: The figure presents the 7x7 confusion matrix for the MobileNet v2 model with face crop preprocessing. On the right side of the figure, the corresponding F1-score values for each class are reported.

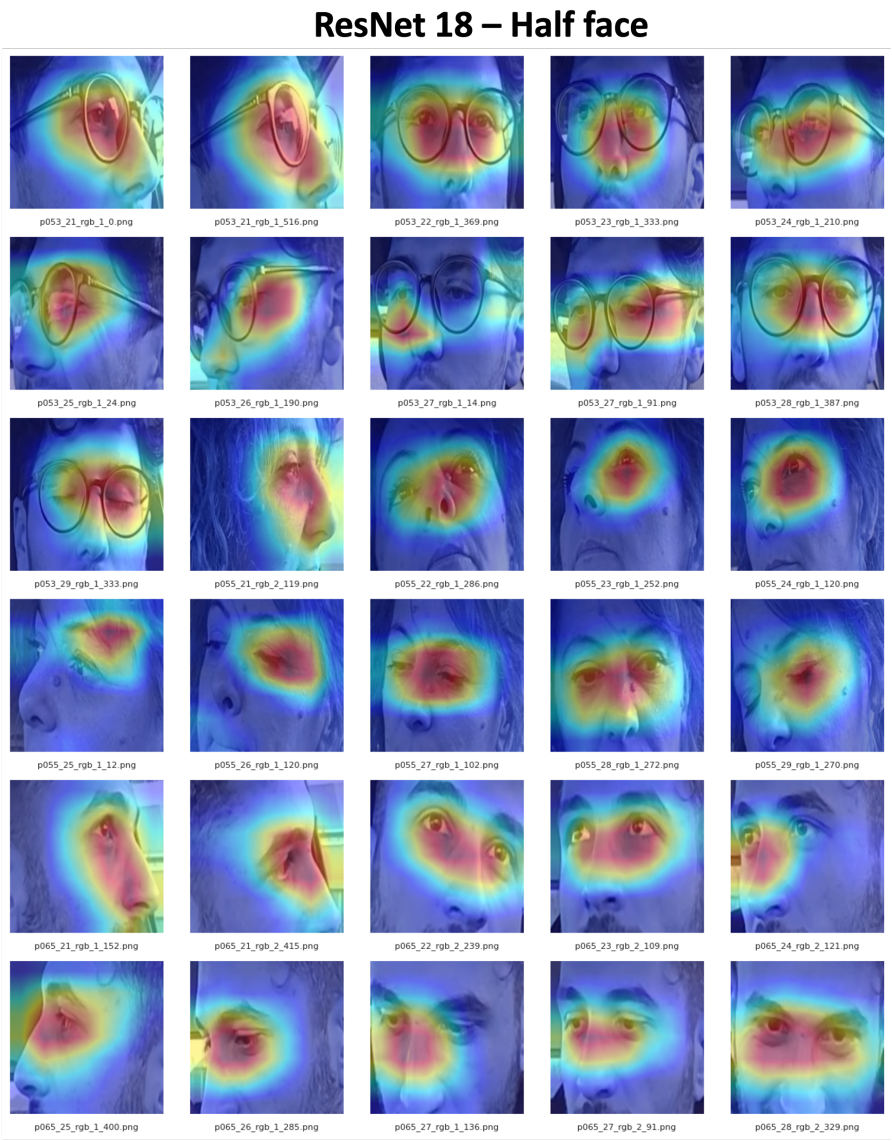


Figure 7.9: The figure displays the heatmaps generated by the ResNet-18 model (baseline performance in Figure 7.7) using 30 frames from the test set.

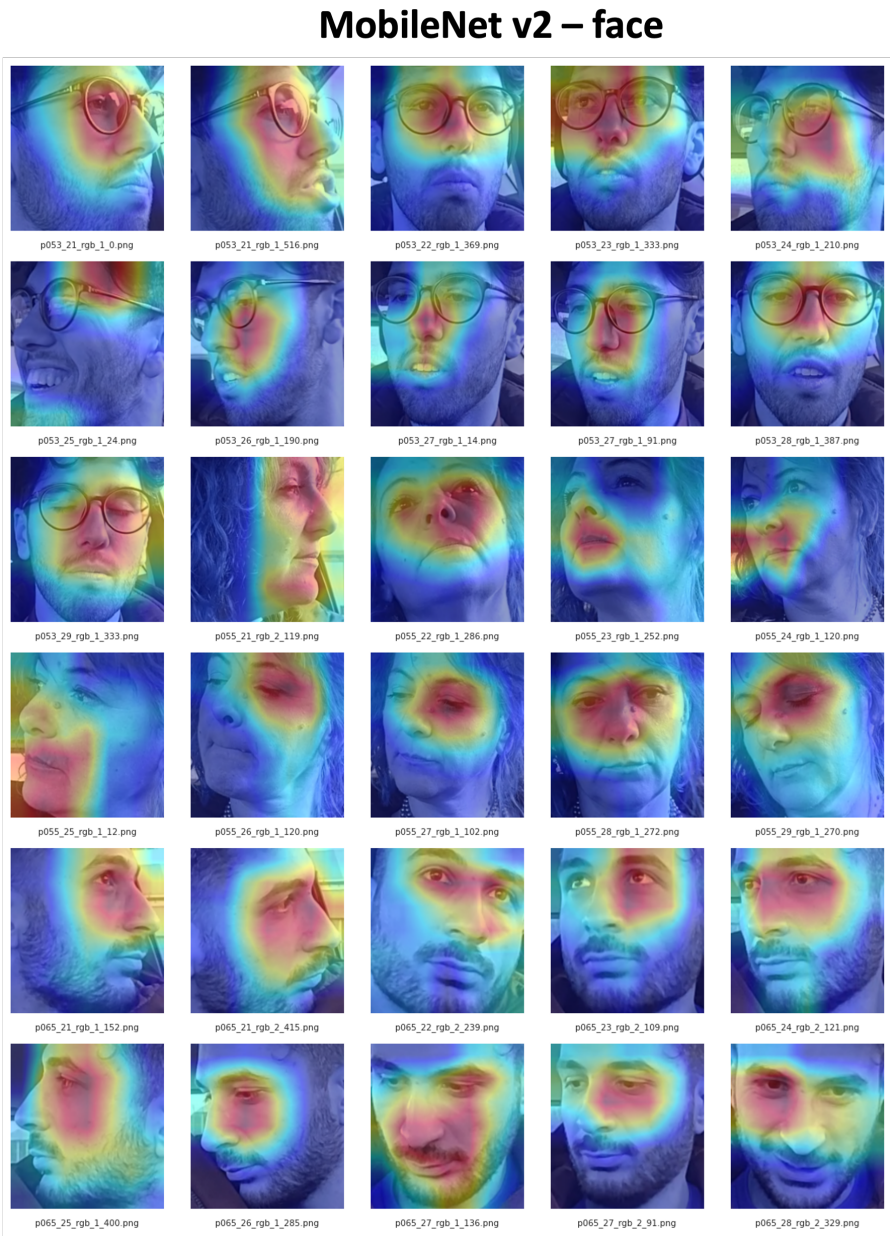


Figure 7.10: The figure depicts the heatmaps generated by the MobileNet v2 model (baseline performance in Figure 7.8) using 30 frames from the test set.

Two-class Model

Ablation experiments were also conducted on binary classifiers, both for distinguishing between classes 23/24 and for distinguishing between classes 26/27. It is important to emphasize that the binary models were trained based on the models previously trained for the classification of the 7 classes. To achieve this, the last layer was removed and replaced with a new one dedicated to binary classification, following the procedure described in Section 6.3.2. The results of these experiments are reported in Tables 3 and 4.

Architecture	**Half Face**		**Face**	
	Accuracy	**F1-score**	**Accuracy**	**F1-score**
Resnet 18	**0.89**	**0.90**	0.85	0.83
EfficientNet B0	0.89	0.89	**0.91**	**0.92**
SqueezeNet 1.1	0.77	0.71	0.70	0.71
VGG 16	0.84	0.85	0.89	0.88
MobileNet v2	0.86	0.82	0.85	0.84

Table 7.5: Ablation experiments were conducted using different CNN architectures and various image cropping regions for the task of binary classification (23/24). The macro-average accuracy and F1-score values obtained for each experiment are reported in tabular form.

Architecture	**Half Face**		**Face**	
	Accuracy	**F1-score**	**Accuracy**	**F1-score**
Resnet 18	**0.91**	**0.89**	**0.92**	**0.90**
EfficientNet B0	0.82	0.78	0.83	0.79
SqueezeNet 1.1	0.79	0.74	0.71	0.72
VGG 16	**0.91**	**0.89**	**0.94**	**0.95**
MobileNet v2	0.86	0.82	0.87	0.83

Table 7.6: Ablation experiments were conducted using different CNN architectures and various image cropping regions for the task of binary classification (26/27). The macro-average accuracy and F1-score values obtained for each experiment are reported in tabular form.

In Figure 7.11, the confusion matrices and the corresponding evaluation metrics of EfficientNet b0 models are shown, which achieved the best performance in the binary classification of classes 23/24. This includes both half face crop and face crop preprocessing. Figure 7.12 instead depicts the confusion matrices and evaluation metrics related to Resnet18 and VGG16 models, also with the best performance in

the binary classification of classes 26/27, both with half face crop and face crop preprocessing.

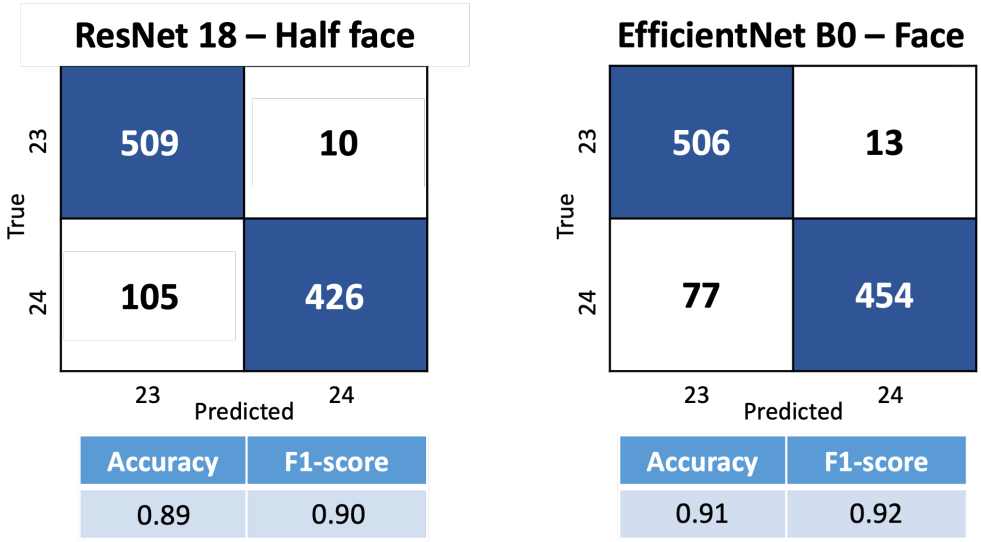


Figure 7.11: The figure displays the 2x2 confusion matrices of models with the highest performance in classifying between classes 23 and 24. The models utilize two types of pre-processing: half face crop and face crop. Below each confusion matrix, the accuracy and F1-score values are provided.

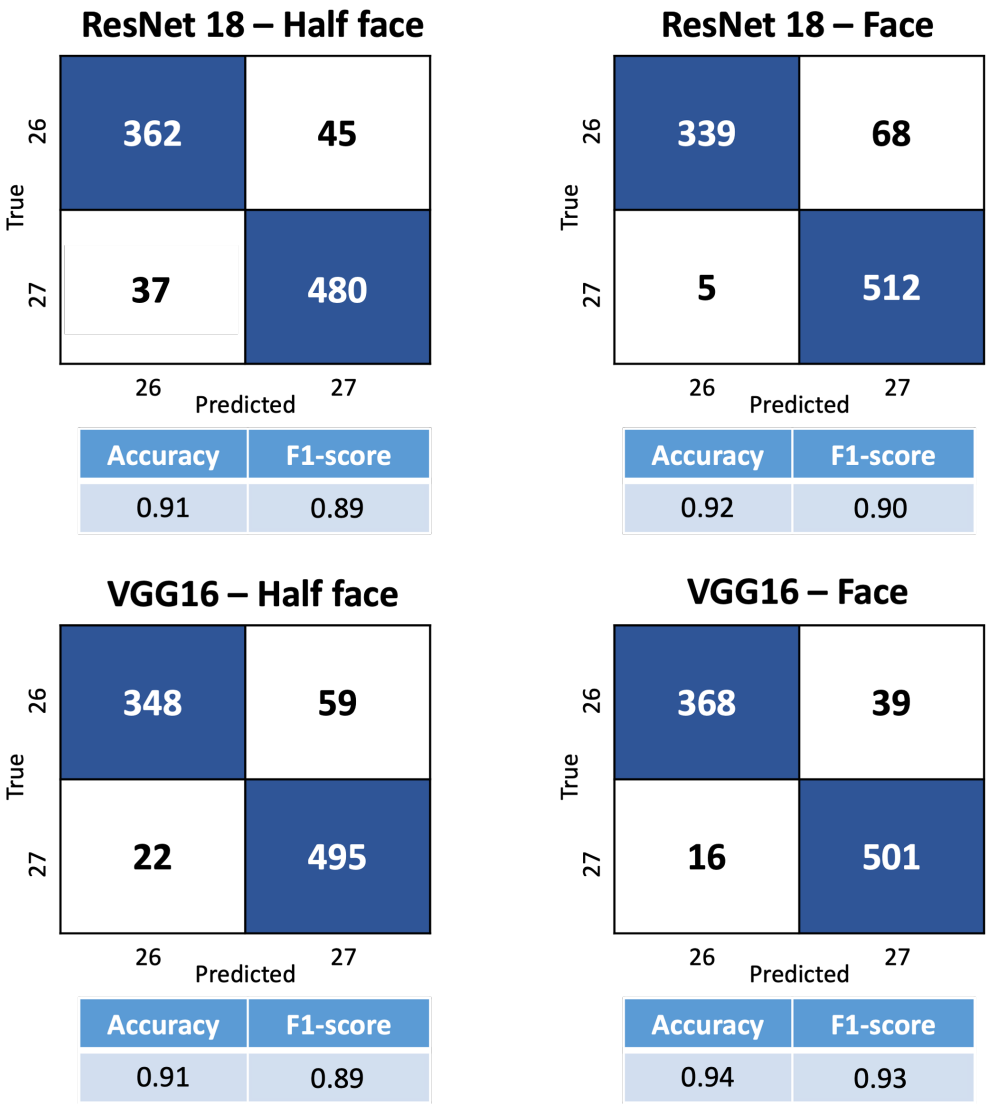


Figure 7.12: The figure displays the 2x2 confusion matrices of models with the highest performance in classifying between classes 26 and 27. The models utilize two types of pre-processing: half face crop and face crop. Below each confusion matrix, the accuracy and F1-score values are provided.

Model 722

Subsequently, the models with the best performance for the 7-class classification (Table 7.4) and the 2-class classification (Tables 7.5 and 7.6) were selected. These three models were combined to perform inference on the test set and classify into 9 classes. It is important to note that this approach could lead to an increase in computational time compared to direct classification with models specifically designed for 9 classes, as the images will have to pass through multiple models in series. In Table 7.7 and 7.8, the results obtained using two different types of preprocessing are reported: cropping of the half-face and full face, respectively.

Models			**Time gpu (ms)**	**Accuracy**
7 classes	**23/24**	**26/27**		
Resnet-18	Resnet-18	Resnet-18	9.3	0.91
Resnet-18	Resnet-18	VGG16	14.8	0.90

Table 7.7: Table reporting accuracy values and computational time in milliseconds relative to the combination of the best classification models presented in Tables 7.4, 7.5, and 7.6. The models considered utilize the cropped region of the upper half of the face as input.

Models			**Time gpu (ms)**	**Accuracy**
7 classes	**23/24**	**26/27**		
MobileNet v2	EfficientNet B0	Resnet-18	21.7	0.91
MobileNet v2	EfficientNet B0	VGG16	18.4	0.91

Table 7.8: Table reporting accuracy values and computational time in milliseconds relative to the combination of the best classification models presented in Tables 7.4, 7.5, and 7.6. The models considered utilize the entire face region as input.

Figures 7.13 and 7.14 show the confusion matrix, F1-score scores, and GPU computation time of the models listed in Table 7.7, which achieved the best performance using half-face cropping as image preprocessing. Similarly, Figures 7.15 and 7.16 present the same graphs related to the models listed in Table 7.8, using instead full-face cropping as image preprocessing.

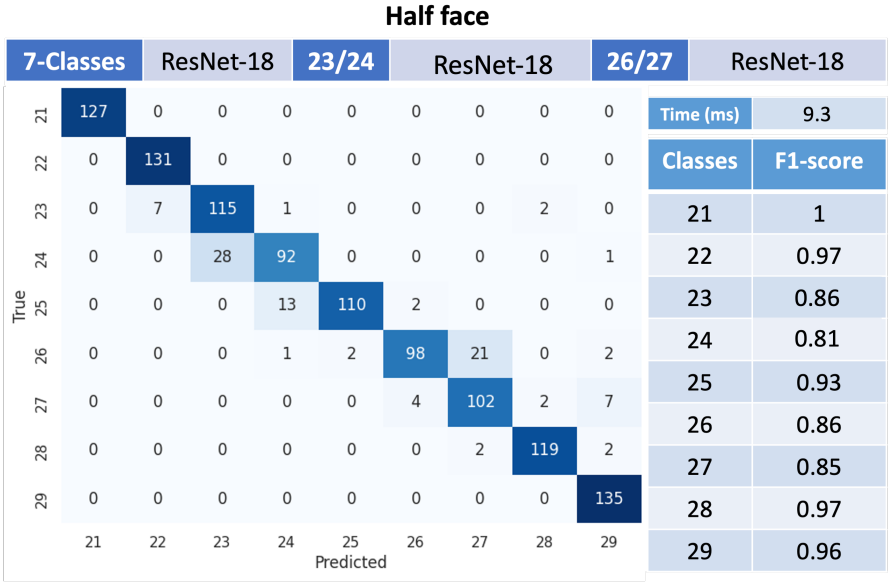


Figure 7.13: The figure depicts the 9x9 confusion matrix of the three CNN models listed above, which utilized a half-face crop preprocessing. Specifically, the ResNet-18 model was used for the classification of the 7 primary classes, followed by EfficientNet B0 to distinguish between classes 23/24 and again ResNet-18 to distinguish between classes 26/27. The F1-score scores for each class are indicated on the right.

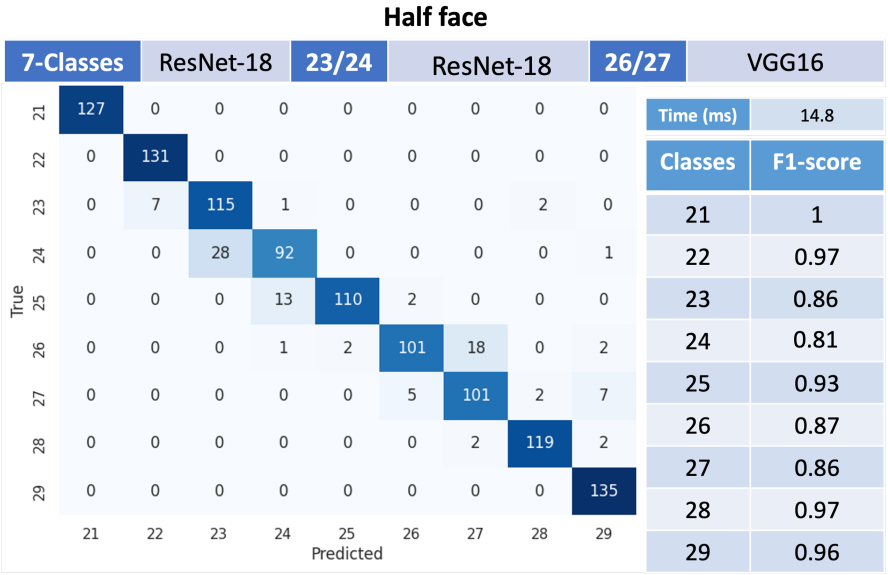


Figure 7.14: The figure displays the 9x9 confusion matrix of the three CNN models listed above, which utilized a half-face crop preprocessing. Specifically, the ResNet-18 model was used for the classification of the 7 primary classes, followed by EfficientNet B0 to distinguish between classes 23/24 and VGG16 to distinguish between classes 26/27. The F1-score scores for each class are indicated on the right.

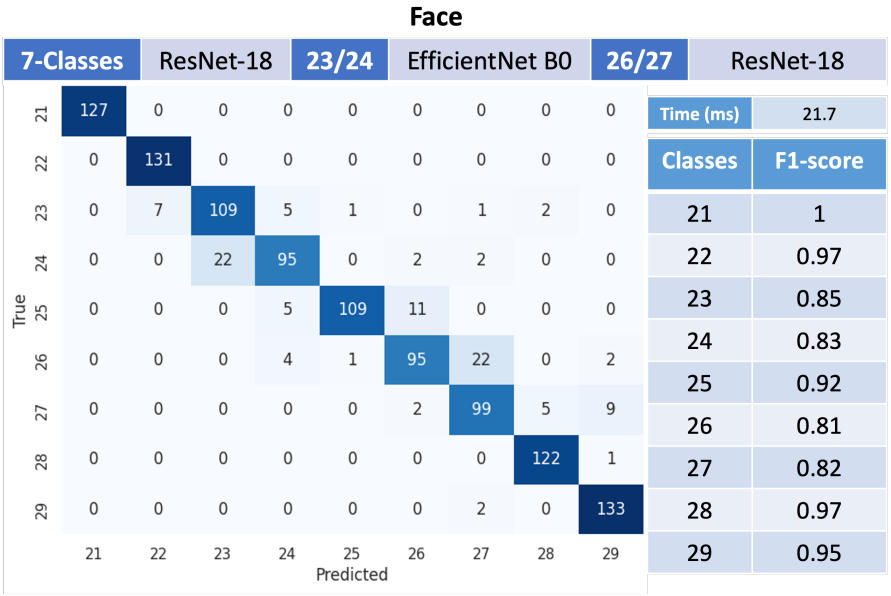


Figure 7.15: The figure displays the 9x9 confusion matrix of the three CNN models listed above, which utilized a face crop preprocessing. Specifically, the ResNet-18 model was used for the classification of the 7 primary classes, followed by EfficientNet B0 to distinguish between classes 23/24 and ResNet-18 to distinguish between classes 26/27. The F1-score scores for each class are indicated on the right.

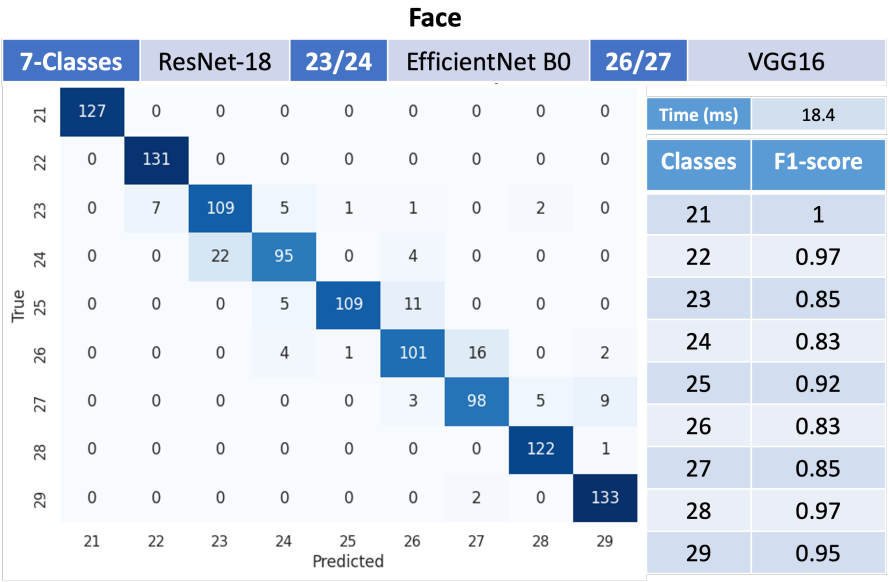


Figure 7.16: The figure illustrates the 9x9 confusion matrix of the three CNN models listed above, which employed a face crop preprocessing. Specifically, the ResNet-18 model was utilized for the classification of the 7 primary classes, followed by EfficientNet B0 to distinguish between classes 23/24 and VGG16 to distinguish between classes 26/27. The F1-score scores for each class are indicated on the right.

7.2.3 Model selection

Half Face crop

From the analysis of the results obtained on models trained with half-face crop preprocessing, and by comparing Tables 7.3 (Pipeline A) and 7.7 (Pipeline B), it emerges that the cascade approach adopted in Pipeline B leads to a significant improvement in model accuracy compared to Pipeline A. In particular, there is a 4.6% increase in accuracy using the ResNet18 model (7-class classification model) + ResNet18 (23/24-class classification model) + ResNet18 (26/27-class classification model). The F1-score values unequivocally confirm an improvement in the cascade approach, with values never lower than 0.80 (refer to Figures 7.2 and 7.14).

Although the average computation time to perform inference on a single frame increases from 3.2 ms in the one-shot model to 9.3 ms in the second approach, this increase does not compromise the possibility of real-time gaze tracking. Considering that the camera's frame rate is 30 frames per second, it is sufficient for the computation time to remain below a certain limit, such as 33 ms, for the model to function smoothly in real-time.

The maximum inference time varies depending on the application's specifications and the context in which the model is used. However, a general rule is that inference should occur within the time of a single video frame to ensure real-time operation. For example, if the camera's frame rate is 30 frames per second, the maximum inference time should be less than approximately 33 milliseconds (considering 1000 milliseconds divided by 30 frames). In many gaze tracking applications, it is preferable to have even shorter inference times, such as 20 milliseconds or less, to allow for a safety margin and ensure a quick and smooth system response.

Face crop

The comparison between models trained with Face crop preprocessing leads to different conclusions. Analyzing Tables 7.3 (Pipeline A) and 7.8 (Pipeline B), there is only a marginal increase of 1 percentage point in accuracy when adopting the cascade approach compared to Pipeline A, accompanied by a computational time increase to 18.4 ms. For this reason, among the two types, Pipeline A is preferred, where the best MobileNet v2 model achieves an accuracy of 90

However, comparing the best 722 model in Figure 7.5 (with Half Face crop preprocessing) with the best one-shot model in Figure 7.14 (with Face crop preprocessing), it is evident that the latter exhibits F1-score values lower than 0.85, even as low as 0.62, unlike the 722 model, which consistently exceeds this threshold, as mentioned earlier. This suggests that the cascade Half Face crop model generalizes better than any model presented in the previous paragraphs, thus representing our state of the art.

Chapter 8

Conclusion

This thesis has delved into the pressing issue of road accidents, placing particular emphasis on driver distraction as a pivotal factor in such incidents. Through thorough analysis of statistics and relevant sources, it has become evident that despite some progress, the number of casualties from road accidents remains unacceptably high, and current measures fall short of achieving the safety targets set for the upcoming decade.

Advanced Driver Assistance Technologies (ADAS), notably Advanced Driver Assistance Systems, have been identified as crucial resources for enhancing road safety. However, it has become clear that visual driver distraction stands as one of the major risk factors, highlighting the critical role that ADAS technologies can play in mitigating this issue.

The implementation of Directive 2019/2144 in the European Union, mandating the standard inclusion of certain ADAS systems in new vehicles, marks a significant stride towards bolstering road safety. Among these systems, the Driver Monitoring System (DMS) holds particular relevance for monitoring driver distraction.

The research and development phase of this thesis focused on the deployment of advanced software for real-time gaze tracking within the vehicle cabin, with the aim of refining driver distraction detection systems. Through analysis of diverse datasets and training of artificial intelligence models, significant outcomes were achieved, underscoring the promise of the proposed methodologies.

The approach involving pre-processing images via cropping the upper half of the face, coupled with the utilization of three cascade models for gaze classification, has demonstrated superior efficacy compared to alternative combinations. The latter encompassed variations in image pre-processing and the employment of one-shot classification models.

Among the three models utilized, the first played a pivotal role in identifying seven of the nine fundamental classes, concentrating on the primary directions of the driver's gaze. The subsequent two models adeptly tackled more intricate

classification challenges previously unexplored in research.

The deployment of these cascade models represents a notable stride forward in driver gaze monitoring and distraction mitigation during driving. The capability to accurately and promptly discern between various gaze directions significantly contributes to road safety, curtailing the risk of distraction-induced accidents.

In conclusion, the findings of this research furnish a robust groundwork for the development of advanced driver assistance systems, thereby fostering safer and more conscientious driving practices.

Chapter 9

Limitations and Future Developments

The limitations of this thesis work encompass several challenges. Firstly, there is a risk of constraints associated with the eye-tracking technology itself, which may not be entirely accurate or swift in all situations, particularly under variable lighting conditions or with partially obscured faces. Additionally, the results obtained might be specific to the context and datasets used, not readily generalizable to other situations or groups of drivers. The complexity of driver distraction, influenced by multiple factors beyond gaze, is not fully addressed by the developed software. Finally, the efficacy of the software could hinge on integration with existing ADAS systems and their market penetration.

Regarding future developments, various directions can be contemplated. It is imperative to enhance the accuracy and robustness of the gaze-tracking software, perhaps through the adoption of more advanced methodologies or the incorporation of data from additional sensors. Moreover, expanding the software's capability to detect not only visual distraction but also other types of distractions is essential. Testing and refining the software in a variety of real-world driving conditions are crucial to ensuring its effectiveness across diverse scenarios. Lastly, integrating the software with more advanced artificial intelligence systems could enable a more dynamic and personalized response to driving conditions and driver distraction.

In conclusion, while this work represents a significant step in driver distraction monitoring, there are still numerous opportunities to enhance the technology and make roads safer for all users.

Bibliography

[1] World Health Organization. *Global Status Report on Road Safety 2023*. Licence: CC BY-NC-SA 3.0 IGO. Geneva: World Health Organization, 2023 (cit. on p. 1).

[2] European Commission. *Drive Distraction 2015*. 2015. URL: https://road-safety.transport.ec.europa.eu/system/files/2021-07/ersosynthesis2015-driverdistraction25_en.pdf (cit. on p. 1).

[3] Adnan Shaout, Dominic Colella, and S. Awad. «Advanced Driver Assistance Systems - Past, present and future». In: *2011 Seventh International Computer Engineering Conference (ICENCO'2011)*. 2011, pp. 72–82. DOI: 10.1109/ICENCO.2011.6153935 (cit. on p. 1).

[4] Filippo Baldisserotto, Krzysztof Krejtz, and Izabela Krejtz. «A Review of Eye Tracking in Advanced Driver Assistance Systems: An Adaptive Multi-Modal Eye Tracking Interface Solution». In: *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications*. ETRA '23. Tubingen, Germany: Association for Computing Machinery, 2023. DOI: 10.1145/3588015.3589512. URL: https://doi.org/10.1145/3588015.3589512 (cit. on p. 1).

[5] Parlamento Europeo e Consiglio dell'Unione Europea. *Regolamento (UE) 2019/2144*. 2019. URL: https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32019R2144 (cit. on p. 1).

[6] *The General Safety Regulations (GSR) and Driver Monitoring Systems (DMS)*. URL: https://smarteye.se/blog/the-general-safety-regulations-gsr-and-driver-monitoring-systems-dms/ (cit. on p. 2).

[7] *Automotive DMS (Driver Monitoring System) Research Report, 2019-2020*. 2020 (cit. on p. 2).

[8] Orval Hobart Mowrer, Theodore Cedric Ruch, and Neal E. Miller. «THE CORNEO-RETINAL POTENTIAL DIFFERENCE AS THE BASIS OF THE GALVANOMETRIC METHOD OF RECORDING EYE MOVEMENTS». In: *American Journal of Physiology* 114 (1935), pp. 423–428.

URL: https://api.semanticscholar.org/CorpusID:100208278 (cit. on p. 7).

[9] David B Carr and Priyanka Grover. «The Role of Eye Tracking Technology in Assessing Older Driver Safety». In: *Geriatrics (Basel, Switzerland)* 5.2 (2020), p. 36. DOI: 10.3390/geriatrics5020036. URL: https://pubmed.ncbi.nlm.nih.gov/32517336/ (cit. on p. 7).

[10] Eugene Levin, Anna Banaszek, Jessica McCarty, and Aleksander Zarnowski. «Attention driven approach for geospatial data update and fusion». In: May 2015 (cit. on p. 7).

[11] Santhoshikka R, Laranya R, and Harshavarthini C. «Eye Tracking and Its Applications». In: *IARJSET* 8 (Aug. 2021). DOI: 10.17148/IARJSET.2021.8824 (cit. on pp. 7–9).

[12] Hr Chennamma and Xiaohui Yuan. «A Survey on Eye-Gaze Tracking Techniques». In: *Indian Journal of Computer Science and Engineering* 4 (Dec. 2013) (cit. on p. 7).

[13] Pavan Kumar Sharma and Pranamesh Chakraborty. «A Review of Driver Gaze Estimation and Application in Gaze Behavior Understanding». In: *arXiv preprint arXiv:2307.01470* (2023). DOI: 10.48550/arXiv.2307.01470. arXiv: 2307.01470 [cs.CV]. URL: https://doi.org/10.48550/arXiv.2307.01470 (cit. on p. 7).

[14] Katarzyna Harezlak and Pawel Kasprowski. «Application of eye tracking in medicine: A survey, research issues and challenges». In: *Computerized Medical Imaging and Graphics* 65 (2018). Advances in Biomedical Image Processing, pp. 176–190. ISSN: 0895-6111. DOI: https://doi.org/10.1016/j.compmedimag.2017.04.006. URL: https://www.sciencedirect.com/science/article/pii/S0895611117300435 (cit. on p. 8).

[15] Gavin Sim and Raymond Bond. «Eye tracking in Child Computer Interaction: Challenges and opportunities». In: *International Journal of Child-Computer Interaction* 30 (2021), p. 100345. ISSN: 2212-8689. DOI: https://doi.org/10.1016/j.ijcci.2021.100345. URL: https://www.sciencedirect.com/science/article/pii/S2212868921000593 (cit. on p. 8).

[16] H. Ashraf, M. H. Sodergren, N. Merali, G. Mylonas, H. Singh, and A. Darzi. «Eye-Tracking Technology in Medical Education: A Systematic Review». In: *Medical Teacher* 40.1 (2018), pp. 62–69. DOI: 10.1080/0142159X.2017.1391373. URL: https://doi.org/10.1080/0142159X.2017.1391373 (cit. on p. 8).

[17] Dongheng Li, Jason Babcock, and Derrick J. Parkhurst. «openEyes: a low-cost head-mounted eye-tracking solution». In: *The Human Computer Interaction Program* () (cit. on p. 8).

[18] Matteo Cognolato, Manfredo Atzori, and Henning Müller. «Head-mounted eye gaze tracking devices: An overview of modern devices and recent advances». In: *Journal of Rehabilitation and Assistive Technologies Engineering* 5 (2018). PMID: 31191938, p. 2055668318773991. DOI: 10.1177/2055668318773991. eprint: https://doi.org/10.1177/2055668318773991. URL: https://doi.org/10.1177/2055668318773991 (cit. on p. 8).

[19] Heiko Drewes and Albrecht Schmidt. «Interacting with the Computer Using Gaze Gestures». In: *Human-Computer Interaction – INTERACT 2007*. Ed. by Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 475–488 (cit. on p. 8).

[20] Yuto Tamura and Kentaro Takemura. «Estimating Point-of-Gaze using Smooth Pursuit Eye Movements without Implicit and Explicit User-Calibration». In: *ACM Symposium on Eye Tracking Research and Applications* (2020). URL: https://api.semanticscholar.org/CorpusID:218831129 (cit. on p. 8).

[21] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. «Webgazer: Scalable Webcam Eye Tracking Using User Interactions». In: AAAI Press, 2016. ISBN: 9781577357704 (cit. on p. 9).

[22] Katarzyna Wisiecka, Krzysztof Krejtz, Izabela Krejtz, Damian Sromek, Adam Cellary, Beata Lewandowska, and Andrew Duchowski. «Comparison of Webcam and Remote Eye Tracking». In: New York, NY, USA: Association for Computing Machinery, 2022. ISBN: 9781450392525. DOI: 10.1145/3517031.3529615. URL: https://doi.org/10.1145/3517031.3529615 (cit. on p. 9).

[23] Pramodini Punde, Mukti Jadhav, and Ramesh Manza. «A study of eye tracking technology and its applications». In: Oct. 2017, pp. 86–90. DOI: 10.1109/ICISIM.2017.8122153 (cit. on p. 9).

[24] *Tobii Pro Glasses 3*. URL: https://www.tobii.com/products/eye-trackers/wearables/tobii-pro-glasses-3 (cit. on p. 10).

[25] *Obsbot Tiny 2 4K PTZ Webcam*. URL: https://www.cameranu.nl/nl/p3343554/obsbot-tiny-2-4k-ptz-webcam (cit. on p. 10).

[26] Nicholas J Wade, Benjamin W Tatler, and Dieter Heller. «Dodge-Ing the Issue: Dodge, Javal, Hering, and the Measurement of Saccades in Eye-Movement Research». In: *Perception* 32.7 (2003). PMID: 12974565, pp. 793–804. DOI: 10.1068/p3470. URL: https://doi.org/10.1068/p3470 (cit. on p. 11).

[27] Edmund Burke Huey. *The Psychology and Pedagogy of Reading*. New York, NY, USA: The Macmillan Company, 1908 (cit. on p. 11).

[28] Robert J. Jacob and Kathy S. Karn. «Eye tracking in human-computer interaction and usability research: Ready to deliver the promises». In: *The Mind's Eye*. Amsterdam, The Netherlands: Elsevier, 2003, pp. 573–605 (cit. on p. 11).

[29] William Sims Bainbridge. *Berkshire Encyclopedia of Human-Computer Interaction*. Vol. 1. Great Barrington, MA, USA: Berkshire Publishing Group LLC, 2004 (cit. on p. 11).

[30] Raymond Dodge and Thomas S. Cline. «The angle velocity of eye movements». In: *Psychological Review* 8 (1901), p. 145 (cit. on p. 12).

[31] Charles H. Judd, Clark N. McAllister, and William Steele. «General introduction to a series of studies of eye movements by means of kinetoscopic photographs». In: *Psychological Review Monographs* 7 (1905), pp. 1–16 (cit. on p. 12).

[32] Walter R. Miles and Edward Shen. «Photographic recording of eye movements in the reading of Chinese in vertical and horizontal axes: Method and preliminary results». In: *Journal of Experimental Psychology* 8 (1925), p. 344. DOI: 10.1037/h0070937 (cit. on p. 12).

[33] H.T. Moore and A.R. Gilliland. «The Measurement of Aggressiveness». In: *Journal of Applied Psychology* 5 (1921), p. 97. DOI: 10.1037/h0071483 (cit. on p. 12).

[34] Hamilton Hartridge and Leslie Thomson. «Methods of investigating eye movements». In: *British Journal of Ophthalmology* 32 (1948), p. 581. DOI: 10.1136/bjo.32.9.581 (cit. on p. 12).

[35] A.L. Yarbus. «Eye movements during perception of complex objects». In: *Eye Movements and Vision*. Berlin, Germany: Springer, 1967, pp. 171–211 (cit. on p. 12).

[36] D. Noton and L. Stark. «Scanpaths in saccadic eye movements while viewing and recognizing patterns». In: *Vision Research* 11 (1971). IN3–IN8, pp. 929–942. DOI: 10.1016/0042-6989(71)90078-3 (cit. on p. 12).

[37] J. Merchant, R. Morrissette, and J.L. Porterfield. «Remote measurement of eye direction allowing subject motion over one cubic foot of space». In: *IEEE Transactions on Biomedical Engineering* 4 (1974), pp. 309–317. DOI: 10.1109/TBME.1974.324969 (cit. on p. 12).

[38] T.N. Cornsweet and H.D. Crane. «Accurate two-dimensional eye tracker using first and fourth Purkinje images». In: *Journal of the Optical Society of America* 63 (1973), pp. 921–928. DOI: 10.1364/JOSA.63.000921 (cit. on p. 12).

[39] D. Cazzato, M. Leo, C. Distante, and H. Voos. «When I Look into Your Eyes: A Survey on Computer Vision Contributions for Human Gaze Estimation and Tracking». In: *Sensors* 20 (2020), p. 3739. DOI: 10.3390/s20133739. URL: https://doi.org/10.3390/s20133739 (cit. on pp. 12–14).

[40] T.E. Hutchinson, K.P. White, W.N. Martin, K.C. Reichert, and L.A. Frey. «Human-computer interaction using eye-gaze input». In: *IEEE Transactions on Systems, Man, and Cybernetics* 19 (1989), pp. 1527–1534. DOI: 10.1109/21.44046 (cit. on p. 12).

[41] X. Xiong, Z. Liu, Q. Cai, and Z. Zhang. «Eye gaze tracking using an RGBD camera: A comparison with a RGB solution». In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. New York, NY, USA, Sept. 2014, pp. 1113–1121. DOI: 10.1145/2638728.2641556 (cit. on p. 12).

[42] B.C. Kim and E.C. Lee. «3D Eye-Tracking Method Using HD Face Model of Kinect v2». In: *Advanced Multimedia and Ubiquitous Engineering*. Berlin, Germany: Springer, 2016, pp. 235–242. DOI: 10.1007/978-3-319-29778-1_25 (cit. on p. 12).

[43] Z. Ye, Y. Li, A. Fathi, Y. Han, A. Rozga, G.D. Abowd, and J.M. Rehg. «Detecting eye contact using wearable eye-tracking glasses». In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. Pittsburgh, PA, USA, Sept. 2012, pp. 699–704. DOI: 10.1145/2370216.2370346 (cit. on p. 13).

[44] D.C. Richardson and M.J. Spivey. «Eye Tracking: Characteristics and Methods». In: *Encyclopedia of Biomaterials and Biomedical Engineering* 3 (2004), pp. 1028–1042 (cit. on p. 13).

[45] Z. Sharafi, Z. Soh, and Y.G. Guéhéneuc. «A Systematic Literature Review on the Usage of Eye-Tracking in Software Engineering». In: *Information and Software Technology* 67 (2015), pp. 79–107. DOI: 10.1016/j.infsof.2015.06.005 (cit. on p. 13).

[46] R.G. Lupu and F. Ungureanu. «A Survey of Eye Tracking Methods and Applications». In: *Bulletin of the Institute of Politehnica Din Iasi Autom. Control Comput. Sci. Sect.* 3 (2013), pp. 72–86 (cit. on p. 13).

[47] P. Kasprowski and J. Ober. «Eye Movements in Biometrics». In: *International Workshop on Biometric Authentication*. Berlin, Germany: Springer, 2004, pp. 248–258 (cit. on p. 13).

[48] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford, UK: OUP Oxford, 2011 (cit. on p. 13).

[49] K. Rayner. «Eye Movements in Reading and Information Processing: 20 Years of Research». In: *Psychological Bulletin* 124 (1998), p. 372. DOI: 10.1037/0033-2909.124.3.372 (cit. on p. 13).

[50] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. «It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation». In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 2299–2308. DOI: 10.1109/CVPRW.2017.284 (cit. on p. 13).

[51] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. «Eye Tracking for Everyone». In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2016, pp. 2176–2184. DOI: 10.1109/CVPR.2016.239. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.239 (cit. on p. 13).

[52] J. Chen and Q. Ji. «3D Gaze Estimation with a Single Camera without IR Illumination». In: *Proceedings of the 2008 19th International Conference on Pattern Recognition*. Tampa, FL, USA, Dec. 2008, pp. 1–4 (cit. on p. 14).

[53] N.A. Dodgson. «Variation and Extrema of Human Interpupillary Distance». In: *Stereoscopic Displays and Virtual Reality Systems XI*. Vol. 5291. International Society for Optics and Photonics. Bellingham, WA, USA, 2004, pp. 36–46 (cit. on p. 14).

[54] D. Cazzato, F. Dominio, R. Manduchi, and S.M. Castro. «Real-time Gaze Estimation via Pupil Center Tracking». In: *Paladyn, Journal of Behavioral Robotics* 9 (2018), pp. 6–18. DOI: 10.1515/pjbr-2018-0002. URL: https://www.degruyter.com/document/doi/10.1515/pjbr-2018-0002.pdf (cit. on p. 14).

[55] W. Maio, J. Chen, and Q. Ji. «Constraint-based Gaze Estimation without Active Calibration». In: *Proceedings of the Face and Gesture 2011*. Santa Barbara, CA, USA, Mar. 2011, pp. 627–631 (cit. on p. 14).

[56] X. Xiong, Z. Liu, Q. Cai, and Z. Zhang. «Eye gaze tracking using an RGBD camera: A comparison with an RGB solution». In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. New York, NY, USA, Sept. 2014, pp. 1113–1121 (cit. on p. 14).

[57] D. Cazzato, A. Evangelista, M. Leo, P. Carcagnì, and C. Distante. «A low-cost and calibration-free gaze estimator for soft biometrics: An explorative study». In: *Pattern Recognition Letters* 82 (2016), pp. 196–206. DOI: 10.1016/j.patrec.2016.06.018 (cit. on pp. 14, 15).

[58] D. Vaughan, T. Asbury, and P. Riordan-Eva. *General Ophthalmology*. Stamford, CT, USA: Appleton & Lange, 1995 (cit. on p. 14).

[59] A. Strupczewski, B. Czuprynski, J. Naruniec, and K. Mucha. «Geometric Eye Gaze Tracking». In: *VISIGRAPP (3: VISAPP)*. Setúbal, Portugal: SCITEPRESS, 2016, pp. 446–457 (cit. on p. 14).

[60] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. «Adaptive linear regression for appearance-based gaze estimation». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (2014), pp. 2033–2046. DOI: 10.1109/TPAMI.2014.2305703 (cit. on p. 15).

[61] R. Valenti, J. Staiano, N. Sebe, and T. Gevers. «Webcam-based visual gaze estimation». In: *International Conference on Image Analysis and Processing*. Springer. 2009, pp. 662–671 (cit. on p. 15).

[62] O. Williams, A. Blake, and R. Cipolla. «Sparse and Semi-supervised Visual Mapping with the S^ 3GP». In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 1. 2006, pp. 230–237. DOI: 10.1109/CVPR.2006.288 (cit. on p. 15).

[63] F. Lu, T. Okabe, Y. Sugano, and Y. Sato. «A Head Pose-Free Approach for Appearance-Based Gaze Estimation». In: *British Machine Vision Conference (BMVC)*. (Accessed on 1 July 2020). 2011, pp. 1–11. URL: http://www.bmva.org/bmvc/2011/proceedings/paper126/paper126.pdf (cit. on p. 15).

[64] S. Park, A. Spurr, and O. Hilliges. «Deep pictorial gaze estimation». In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 721–738 (cit. on p. 15).

[65] F. Martinez, A. Carbone, and E. Pissaloux. «Gaze estimation using local features and non-linear regression». In: *Proceedings of the 2012 19th IEEE International Conference on Image Processing*. IEEE. Orlando, FL, USA, Sept. 2012, pp. 1961–1964 (cit. on p. 15).

[66] Q. Huang, A. Veeraraghavan, and A. Sabharwal. «TabletGaze: Dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets». In: *Machine Vision and Applications* 28 (2017), pp. 445–461. DOI: 10.1007/s00138-017-0894-2 (cit. on p. 15).

[67] K. A. Funes-Mora and J. M. Odobez. «Gaze estimation in the 3D space using RGB-D sensors». In: *International Journal of Computer Vision* 118 (2016), pp. 194–216. DOI: 10.1007/s11263-015-0881-x (cit. on p. 15).

[68] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. «Deep learning». In: *Nature* 521 (2015), pp. 436–444 (cit. on p. 16).

[69] Blaise Noris, J. Bradley Keller, and Aude Billard. «A wearable gaze tracking system for children in unconstrained environments». In: *Comput. Vis. Image Underst.* 115 (2011), pp. 476–486 (cit. on p. 16).

[70] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. «Learning-by-synthesis for appearance-based 3D gaze estimation». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. Columbus, OH, USA, June 2014, pp. 1821–1828 (cit. on pp. 16, 20, 21).

[71] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. «Appearance-based gaze estimation in the wild». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. Boston, MA, USA, June 2015, pp. 4511–4520 (cit. on pp. 16, 20).

[72] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. «Eyetracking for everyone». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, NV, USA, June 2016, pp. 2176–2184 (cit. on p. 16).

[73] Weipeng Zhu and Haotian Deng. «Monocular free-head 3D gaze tracking with deep learning and geometry constraints». In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE. Venice, Italy, Oct. 2017, pp. 3143–3152 (cit. on pp. 16, 19–21).

[74] Cristina Palmero, Jose Selva, Mohammad Ali Bagheri, and Sergio Escalera. «Recurrent CNN for 3D gaze estimation using appearance and shape cues». In: *arXiv preprint arXiv:1805.03064* (2018) (cit. on p. 17).

[75] Guan Liu, Yue Yu, Karla Andrea Fernandez Mora, and Jean-Marc Odobez. «A Differential Approach for Gaze Estimation». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019) (cit. on p. 17).

[76] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar. «Gazelocking: Passive Eye Contact Detection for Human-Object Interaction». In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. St. Andrews, UK, Oct. 2013, pp. 271–280 (cit. on p. 22).

[77] K. A. Funes Mora, F. Monay, and J. M. Odobez. «Eyediap: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras». In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. Denver, CO, USA, Mar. 2014, pp. 255–258 (cit. on p. 22).

[78] Y. Sugano, Y. Matsushita, and Y. Sato. «Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA, June 2014, pp. 1821–1828 (cit. on p. 23).

[79] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. «Appearance-Based Gaze Estimation in the Wild». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, USA, June 2015, pp. 4511–4520 (cit. on p. 23).

[80] Sourabh Vora, Akshay Rangesh, and Mohan Trivedi. «Driver Gaze Zone Estimation Using Convolutional Neural Networks: A General Framework and Ablative Analysis». In: *IEEE Transactions on Intelligent Vehicles* PP (Feb. 2018). DOI: 10.1109/TIV.2018.2843120 (cit. on pp. 25–28, 30, 48).

[81] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. «Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1717–1724 (cit. on p. 25).

[82] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. «ImageNet: A Large-Scale Hierarchical Image Database». In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255 (cit. on pp. 25, 30, 52).

[83] K. Yuen, S. Martin, and M. M. Trivedi. «Looking at Faces in a Vehicle: A Deep CNN Based Approach and Evaluation». In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE. 2016, pp. 649–654 (cit. on p. 28).

[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton. «Imagenet Classification with Deep Convolutional Neural Networks». In: *Advances in Neural Information Processing Systems*. 2012, pp. 1097–1105 (cit. on p. 28).

[85] K. Simonyan and A. Zisserman. «Very Deep Convolutional Networks for Large-Scale Image Recognition». In: *CoRR* abs/1409.1556 (2014) (cit. on p. 28).

[86] 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 (cit. on p. 28).

[87] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. «Squeezenet: Alexnet-Level Accuracy with 50x Fewer Parameters and 0.5 MB Model Size». In: *arXiv preprint arXiv:1602.07360* (2016) (cit. on pp. 28, 53).

[88] A. Hosna, E. Merry, J. Gyalmo, et al. «Transfer Learning: A Friendly Introduction». In: *Journal of Big Data* 9.102 (2022). DOI: 10.1186/s40537-022-00652-w (cit. on p. 29).

[89] J. Nuevo, L. M. Bergasa, and P. Jiménez. «Rsmat: Robust simultaneous modeling and tracking». In: *Pattern Recognition Letters* 31 (2010), pp. 2455–2463 (cit. on p. 33).

[90] K. Diaz-Chito, A. Hernández-Sabaté, and A. M. López. «A reduced feature set for driver head pose estimation». In: *Applied Soft Computing* 45 (2016), pp. 98–107 (cit. on p. 33).

[91] A. Jain, H. S. Koppula, S. Soh, B. Raghavan, A. Singh, and A. Saxena. «Brain4cars: Car that knows before you do via sensory-fusion deep learning architecture». In: *arXiv preprint arXiv:1601.00740* (2016) (cit. on p. 33).

[92] A. Schwarz, M. Haurilet, M. Martinez, and R. Stiefelhagen. «Driveahead-a large-scale driver head pose dataset». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 1–10 (cit. on p. 34).

[93] J. D. Ortega, N. Kose, P. Cañas, M.-A. Chao, A. Unnervik, M. Nieto, O. Otaegui, and L. Salgado. «Dmd: A large-scale multi-modal driver monitoring dataset for attention and alertness analysis». In: *European Conference on Computer Vision*. Springer. 2020, pp. 387–405 (cit. on p. 34).

[94] A. Rangesh, B. Zhang, and M. M. Trivedi. «Driver gaze estimation in the real world: Overcoming the eyeglass challenge». In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 1054–1059 (cit. on p. 34).

[95] R. F. Ribeiro and P. D. Costa. «Driver gaze zone dataset with depth data». In: *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE. 2019, pp. 1–5 (cit. on pp. 34, 36).

[96] S. Ghosh, A. Dhall, G. Sharma, S. Gupta, and N. Sebe. «Speak2label: Using domain knowledge for creating a large scale driver gaze zone estimation dataset». In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2896–2905 (cit. on p. 34).

[97] W. Liu and et al. «SSD: Single Shot MultiBox Detector». In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9905. Lecture Notes in Computer Science. Springer, Cham, 2016. DOI: 10.1007/978-3-319-46448-0_2 (cit. on pp. 48, 50, 51).

[98] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV] (cit. on p. 53).

[99] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. arXiv: 1905.11946 [cs.LG] (cit. on p. 53).

[100] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV] (cit. on p. 53).

[101] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. 2019. arXiv: 1801.04381 [cs.CV] (cit. on p. 53).

[102] Anqi Mao, Mehryar Mohri, and Yutao Zhong. *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*. 2023. arXiv: 2304.07288 [cs.LG] (cit. on p. 55).

[103] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG] (cit. on p. 55).

[104] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs.LG] (cit. on p. 56).

[105] Tiago Carneiro, Raul Victor Medeiros Da NóBrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C. De Albuquerque, and Pedro Pedrosa Rebouças Filho. «Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications». In: *IEEE Access* 6 (2018), pp. 61677–61685. DOI: 10.1109/ACCESS.2018.2874767 (cit. on p. 56).

Acknowledgements

I wish to express my deep gratitude to all the people who have contributed to the success of this thesis.

First and foremost, I would like to thank Prof. Massimo Salvi, my academic advisor, for being a valuable and constant guide throughout these months, providing his invaluable support and offering crucial advice during my thesis journey.

I thank Brain Technologies, and especially Luca, for offering me the opportunity to collaborate and for their valuable support during my research. I sincerely appreciate the warm welcome and stimulating work environment they have created.

I want to dedicate a special thank you to my parents, Ilario and Celeste, and my brother Andrea. Your love, support, and sacrifices have made every success of mine possible. Without you, I could not have reached this milestone. I am infinitely grateful for everything you have done for me.

I thank my grandmother Rosa, my second mother. With her unconditional love, she has always encouraged me to do my best and has had infinite faith in my abilities. Thank you, Grandma, for teaching me the pure values of love, humility, and kindness.

I would like to sincerely thank Gloria, my girlfriend. She has been my rock, my support, and my source of inspiration at every moment. When I didn't believe in myself, she always saw my potential and encouraged me wholeheartedly. Thank you for everything you have done and continue to do for me. Your presence in my life is an immense value that I will never cease to recognize.

I wish to extend warm thanks to all my aunts, uncles, and cousins. Your presence has added warmth and joy to my life. Thank you for sharing special moments with

me and making family a place of love and mutual support.

I thank my former roommates and now great friends, Michele, Gabriele, and Alessio. Their understanding, constant support, and pleasant cohabitation have made this experience even more significant and memorable.

I thank Antonio, Vito, and Giovanni, valuable colleagues met during my university journey. With them, I have shared not only classes and exams but also moments of intense work and mutual support.

Thanks to my dear lifelong friends, Filippo M., Francesco, Giulia, Filippo C. Every moment spent together in our city has been a unique experience, full of joy, fun, and carefreeness.

Lastly, I want to dedicate a deep and sincere thank you to Anna Maria, an extraordinary support figure who has guided my path since I was almost a child. Her constant encouragement, shared wisdom, and kindness have shaped the person I am today. Without her valuable contribution, I would never have come this far. Her impact on my life is indelible, and her example will stay with me forever. Thank you from the bottom of my heart, Anna Maria, for believing in me and for being a constant source of inspiration and guidance.

Italiano:

Desidero esprimere la mia profonda gratitudine a tutte le persone che hanno contribuito al successo di questa tesi.

Innanzitutto, vorrei ringraziare il Prof. Massimo Salvi, il mio tutor accademico, per essere stato una guida preziosa e costante durante tutti questi mesi, fornendo il suo prezioso supporto e offrendo consigli fondamentali nel corso del mio percorso di tesi.

Ringrazio l'azienda Brain Technologies, e in particolare Luca, per avermi offerto l'opportunità di collaborare e per il loro prezioso sostegno durante la mia ricerca. Apprezzo sinceramente l'accoglienza calorosa e l'ambiente di lavoro stimolante che hanno creato.

Desidero dedicare un ringraziamento speciale ai miei genitori, Ilario e Celeste, e a mio fratello Andrea. Il vostro amore, il vostro sostegno e i vostri sacrifici hanno

reso possibile ogni mio successo. Senza di voi, non avrei potuto raggiungere questo traguardo. Vi sono infinitamente grato per tutto quello che avete fatto per me.

Ringrazio la mia nonna Rosa, la mia seconda mamma. Con il suo amore incondizionato, mi ha sempre incoraggiato a dare il meglio di me stesso e ha riposto una fiducia infinita nelle mie capacità. Grazie Nonna per avermi insegnato i valori puri di amore, umiltà e gentilezza.

Vorrei ringraziare di cuore Gloria, la mia ragazza. Lei è stata la mia roccia, il mio sostegno e la mia fonte di ispirazione in ogni momento. Quando io stesso non credevo in me, lei ha sempre visto il mio potenziale e mi ha incoraggiato senza riserve. Grazie per tutto ciò che hai fatto e continui a fare per me. La tua presenza nella mia vita è un valore immenso che non smetterò mai di riconoscere.

Desidero estendere un caloroso ringraziamento a tutti i miei zii e cugini. La vostra presenza ha aggiunto calore e gioia alla mia vita. Grazie per aver condiviso con me momenti speciali e aver reso la famiglia un luogo di amore e supporto reciproco.

Ringrazio i miei ex coinquilini e ormai grandi amici, Michele, Gabriele e Alessio. La loro comprensione, il loro costante supporto e la piacevole convivenza hanno reso questa esperienza ancora più significativa e memorabile.

Ringrazio Antonio, Vito e Giovanni, preziosi colleghi incontrati durante il percorso universitario. Con loro ho condiviso non solo le lezioni e gli esami, ma anche momenti di intenso lavoro e sostegno reciproco.

Grazie ai miei cari amici di sempre, Filippo M., Francesco, Giulia, Filippo C. Ogni momento trascorso insieme nella nostra città è stato un'esperienza unica, ricca di gioia, divertimento e spensieratezza.

Infine, desidero dedicare un ringraziamento profondo e sincero ad Anna Maria, una figura di straordinario sostegno che ha guidato il mio cammino sin da quando ero quasi un bambino. Il suo costante incoraggiamento, la saggezza condivisa e la gentilezza hanno plasmato la persona che sono oggi. Senza il suo prezioso contributo, non sarei mai arrivato dove sono ora. Il suo impatto sulla mia vita è incancellabile e il suo esempio rimarrà con me per sempre. Grazie di cuore, Anna Maria, per aver creduto in me e per essere stata una fonte costante di ispirazione e guida.