An AI based Safe Driving Support System using Two Dashboard Cameras

March 2024

Doctor of Engineering

Ulziibayar Sonom-Ochir

Department of Information Science and Intelligent Systems
Graduate School of Advanced Technology and Science,
Tokushima University
Tokushima, Japan

Abstract

Driving support systems are of paramount importance in the modern era due to their ability to reduce human errors and improve overall road safety that can save lives by reducing and preventing accidents. They also address the rising issue of distracted driving, optimize traffic flow, meeting consumer preferences for enhanced vehicle features, and align with regulatory efforts to make driving safer and more efficient.

Detecting driver distraction promptly is imperative for enhancing road safety. While various methodologies and technologies have been explored to address this issue, we present an innovative, cost-efficient, non-intrusive, and lightweight Safe Driving Support System (SDSS) that utilizes dual dashboard cameras. In addition to conventional driver's gaze tracking, our system considers other broader aspects, including monitoring the road environment and pedestrian safety. Our study comprises two primary modules: distracted driver detection and pedestrian safety.

Driver's distraction detection: This module evaluates distraction by analyzing the driver's gaze direction and the position of pedestrians on the road. It consists of two parts with parallel procedures. The first is to estimate the direction of the driver's gaze, and the second is to detect the pedestrian and determine their position. In the first part, the system receives the video captured through the driver monitoring camera and then defines the gaze region the driver is looking at. Through extensive experimentation, we investigated how different camera positions affect gaze estimation. Moreover, we explored strategies that use appearance-based solutions, including a combination of gaze and head features, domain adaptation solutions to enhance gaze mapping's robustness to various drivers and environments and several camera positions. From these strategies, OpenFace with SVM classifier (using gaze angle, head position_R, head rotation_R, and eye position WO-Z features) using camera position 2, outperformed others, achieving an 85.6% accuracy rate for the Strictly Correct Estimation Rate (SCER) and a 98.7% accuracy rate for the Loosely Correct Estimation Rate (LCER). Notably, we also employed unsupervised domain adaptation through a conditional Generative Adversarial Network (GAN) to ensure accurate gaze mapping across diverse drivers and environments. The domain adaptation approach used showed an average Strictly Correct Estimation Rate (SCER) accuracy of 81.38% and 93.53%, along with a Loosely Correct Estimation Rate (LCER) accuracy of 96.69% and 98.9% for the two different strategies, respectively. These results demonstrate the effectiveness of our method in adapting to different domains.

Additionally, we achieved an average SCER accuracy of 85.00% and 94.84%, along with LCER accuracy of 98.80% and 99.23% for the two strategies, respectively. This showcases the adaptability of our approach to handle various environments and even different camera positions for the same driver, indicating potential self-calibration capabilities. Simultaneously, the second procedure receives video from the front-view camera to identify pedestrian activity. By combining the data from all road users, we can evaluate the driver's distraction level.

Pedestrian Safety: This module assesses the risk level of pedestrians based on road lane lines and pedestrian's relative positions using the video feed from the front-view camera. To determine pedestrian's safety, we divided the road into sections based on the level of risk to the pedestrians, including high-risk, risky, and safe regions based on the lane lines. Our pedestrian safety module relies on two procedures: lane line detection and pedestrian distance and position detection. We have experimented with pedestrian distance and position detection procedures using methods such as the optic flow method and Deep learning methods. This integration enables the system to provide real-time feedback on potential hazards in the driver's vicinity. Moreover, recognizing the paramount importance of pedestrian safety, we have introduced a dedicated Pedestrian Safety part. This module demonstrated promising results, with an average lane line recognition accuracy of 95.79% and a pedestrian distance and position detection accuracy of 86.45%.

Together, these modules offer an early detection and mitigation solution for the leading causes of accidents: driver distraction and pedestrian risk. In summary, our Safe Driving Support System provides a comprehensive and cost-effective approach to enhance road safety by addressing driver distraction and pedestrian safety. The results from our experiments demonstrate the system's effectiveness in detecting and mitigating potential hazards on the road, contributing to a safer transportation environment and the prevention of accidents.

Keywords

Safe Driver Support Systems, visual distraction, gaze mapping, moving object, gaze regions, gaze estimation, driving environment, visual attention, pedestrian safety.

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] **S. Ulziibayar**, S. Karungaru, K. Terada, and A. Altangerel,
Detection of Driver's Visual Distraction using Dual Cameras
International Journal of Innovative Computing, Information and Control, Vol.18, No.05, (Oct 2022), 1445-1461.

[**Paper II**] **S. Ulziibayar**, S. Karungaru, K. Terada, and A. Altangerel,
Domain Adaptation for Driver's Gaze Mapping for Different Drivers and New Environments
International Journal of Advances in Intelligent Informatics, Vol.10, No.02, (Feb 2024), 2442-6571.

[**Paper III**] **S. Ulziibayar**, S. Karungaru, K. Terada, and A. Altangerel,
Appearance-based Drivers Gaze Mapping using Dash Camera
International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS2022), Vol.23, No.12, (Jan 2023), 1-5.

[**Paper IV**] **S. Ulziibayar**, S. Karungaru, K. Terada, and A. Altangerel,
Dashboard-camera-based Safe Driving Support System
The 8th IEEE/ACIS International Conference on Big Data, Cloud Computing, and Data Science Engineering (BCD 2023), (Dec 2023).

Acknowledgment

I want to express my heartfelt gratitude to Professor Dr. Stephen Karungaru for his unwavering support during my doctoral studies. His valuable advice, motivation, and guidance have been instrumental in my research and thesis writing. I am deeply grateful to him for all the things I have learned from him in the last three years. Without his guidance and advice, I would not have been able to complete this study.

I also want to extend my gratitude to Professor Dr. Kenji Terada from the Department of Information Science and Intelligent Systems at Tokushima University, as well as to Dr. Akinori Tsuji, for their advice, support and assistance in keeping me on track with my progress.

I am also thankful to Professor Altangerel Ayush from the School of Information and Communication Technology, Mongolian University of Science and Technology for giving me the opportunity to pursue a Ph.D in Japan.

Furthermore, I want to express my sincere gratitude to the organizations behind the Higher Engineering Education Development Project (MJEED). Their support has allowed me to pursue a doctoral program in Japan and provided me with full financial assistance. Lastly, I want to thank all the members of the B1 lab (STIR-AI) who have studied with me. I am grateful for the wonderful three years we spent together.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Related works

1.1 Introduction and objectives

The World Health Organization reports that traffic accidents are one of the top eight causes of death, resulting in over 1.3 million fatalities worldwide each year. Hence, it is crucial to monitor the driving process, evaluate driver distraction levels, and offer warnings or support to the driver [1]. Safe driving support systems are of paramount importance in the modern era due to their ability to save lives by preventing accidents, reducing human errors, and improving overall road safety. They also address the rising issue of distracted driving, optimize traffic flow, meet consumer preferences for enhanced vehicle features, and align with regulatory efforts to make driving safer and more efficient. As the prevalence of serious accidents attributed to driver distraction continues to escalate, the development of a robust Safe Driving Support System (SDSS) is imperative for enhancing road safety.

In the pursuit of safer roadways and reduced traffic accidents, this thesis endeavors to develop an innovative, cost-efficient, and non-intrusive SDSS harnessing the capabilities of dual dashboard cameras. Our research comprises two parts, focusing on driver distraction detection and pedestrian safety, both integral components of our proposed safe driving support system. In the modern world, road safety remains a crucial concern, with driver distraction and pedestrian-related incidents being significant contributors to traffic accidents. To address these challenges, our thesis seeks to contribute a novel approach that leverages advanced technology and computer vision, providing a comprehensive solution for enhancing road safety.

1.2 Related works

A plethora of studies have been conducted to identify and prevent driver distractions, which could lead to potential accidents. These studies are categorized based on the method used to determine the safe driving process, with three primary types of data employed to detect distracted drivers. Generally, three primary types of data are used to detect distracted drivers. The first method involves physiological data, such as electrocardiograms and electroencephalograms, which monitor the driver's heart rate and brain activity, respectively. Detecting distractions through EEG-based Brain-Computer Interfaces (BCIs) has been proposed as a promising solution. This type of research introduces an automatic framework that incorporates BCIs and a realistic driving simulator for detecting distractions [2][3][4][5]. The second type involves vehicle control data, including pedal positions and steering wheel movements, which provide insight into the driver's behavior and response time. The primary objective of this type of research is to evaluate the effectiveness of leading supervised learning classification algorithms in detecting driving states. This is a critical issue for comprehending the driver's mood or driving habits through the use of sensor data from the CAN (Control Area Network) bus [6] [7]. Lastly, the third type uses visual data, such as eye and body movements, as well as images or videos of the driver's facial expressions [8], to assess the driver's level of distraction. While all three types have been explored, the majority of research has focused on visual data, which provides a comprehensive overview of the driver's behavior while on the road.

Our study also concentrates on identifying distracted drivers using visual data. Although driving activities are largely dependent on the driver, all road users must be considered when determining safe driving conditions. Therefore, we considered road environment, pedestrians, and other road users as variables to determine driver's distraction. Within two primary modules—driver's distraction detection and pedestrian safety—we highlight various related works about driver gaze mapping, pedestrian detection, and pedestrian distance and relative position measurement in this section.

Driver's gaze mapping: A lot of studies have focused on the issue of the driver's gaze estimation task. These related studies can be classified into hardware-based and appearance-based methods. The hardware-based studies often use additional equipment to determine the driver's visual attention. Several approaches based on using wearable devices [9][10] monitored the driver's visual attention. Mizuno et al. conducted a system of visual attention detection using a gaze tracker and a vehicle-mounted device [11]. Also, Wang et al. conducted the driver's gaze tracking system using a dual camera [12]. Although these approaches are effective and robust, they are intrusive, costly, and unsuitable for real application. Also, this type of system is difficult to use and fatigues the driver. Appearance-based gaze mapping aims to predict the driver's gaze direction from their visual appearance. These types of studies can be classified into methods considering both eye and head orientation and methods that consider only head orientation and eye gaze. The following studies conduct driver visual attention-based eye gaze [13][14]. Xiao and Feng [13] used a pupil-based gaze mapping estimator using the Haar classifier. To identify facial features, such as both eyes, the corners of the lips, and the bounding box of the face, Smith et al. [14] conducted color and intensity. They estimated the head orientation and gaze direction using these features. However, if the driver is talking or wearing glasses, this approach sometimes fails to detect facial features [14]. Eye direction estimation can be made using the above methods while considering eye gaze. However, it is not always possible to detect eye direction inside a vehicle environment since the driver's eye blink, a considerable head rotation, and sunshine reflections on eyeglasses can all obscure the eye region. On the other hand, many researchers estimated gaze direction by using head orientation [15][16][17]. This method determines the direction the driver is looking at using only the head orientation, such as the left and right borders and the center of the head, without using detailed facial features [15]. Considering only the head position when estimating the driver's gaze direction has many advantages, such as less dependence on training data and no need for extensive training. It is also an advantage that it is possible to detect gaze direction when the method cannot determine detailed face features. Although this method has many advantages, it has lower accuracy in head orientation estimation.

Therefore, the gaze estimate task requires simultaneous consideration of both head position and eye gaze for inference. Some studies used eye gaze and head orientation to determine the gaze direction. To identify the pupils, nose bottom, and pupil glints, Kaminski et al. analyzed the intensity, shape, and size features. They estimated continuous head orientation and gaze direction based on these features and an anthropomorphic model [18]. Also, Naqvi et al. [19] used three deep CNNs that use the images of the left eye, right eye, and full face, and combine the outputs by these three CNNs for gaze estimation. This study demonstrated excellent performance but required complicated initialization. Although the above studies used a combination of head position and eye gaze features, it is still less possible to draw inferences using either feature alone. The use of this state-of-the-art of gaze estimation method on real vehicles is restricted by the fact that they are intrusive and expensive.

Moreover, one of the problems of this research field is the insufficient training dataset. Although there are several open-source datasets, most are designed for specific environments. This means that real driving environment training data is rare. Camera calibration is also important in this field, and other estimations are made depending on the camera setting. Therefore, no matter how well gaze estimation is done, the performance may decrease depending on the camera settings. To address these challenges, techniques for domain adaptation are employed to mitigate the negative effects of domain shifting, allowing the model to be applicable across different domains and environments using small dataset. Wang et al. utilize an appearance discriminator and head pose classifier to achieve domain adaptation by adversarial learning [20]. Meanwhile, Cheng et al. proposed to enhance cross-domain performance without target domain data by eliminating gaze-irrelevant features [21]. More recently, Bao et al. have proposed a rotation-enhanced unsupervised domain adaptation technique for the problem of the lack of access to target domain labels in real-world situations [22]. While the aforementioned methodologies have demonstrated notable advancements in enhancing gaze-related tasks, it is imperative to distinguish between gaze mapping and gaze estimation. Gaze estimation primarily involves determining the direction in which a person's gaze is focused, typically relying on technologies like eye-tracking to pinpoint the location of the eyes and infer the point of focus. Essentially, gaze estimation answers the question of 'where' the eyes are directed. On the other hand, gaze mapping extends beyond mere estimation, aiming to provide a comprehensive and spatial representation of the entire gaze behavior. Gaze mapping encompasses not only the predefined regions but also the dynamic patterns, head and body movements, and interactions of the gaze within a given environment. It seeks to create a detailed map or model that reflects how the individual's gaze traverses and engages with different elements in their surroundings. In the context of driver behavior, gaze mapping becomes particularly crucial for understanding not just the instantaneous points of focus but also the broader context of how the driver visually navigates through complex outdoor environments. This includes considerations for factors such as scanning the road, monitoring mirrors, and responding to dynamic stimuli. Despite significant progress in gaze estimation, achieving accurate and robust gaze mapping, especially in the challenging conditions presented by outdoor driving environments and without additional devices, remains a formidable task in the field of research and development.

In this thesis, we present several strategies that use appearance-based (a combination of gaze and head features) and domain adaptation solutions to improve gaze mapping's robustness to different drivers and new environments.

Pedestrian detection: using dashboard cameras is a critical topic in computer vision with various applications in advanced driver assistance systems, surveillance, safety systems, and advanced robotics. It is an essential component in both of our modules. The primary function of pedestrian detection is to locate pedestrians, determine their distance, and assess the risk level. Numerous researchers have studied pedestrian detection using dashboard cameras in recent years. Unlike general pedestrian detection, this study aims to detect pedestrians in a dynamic background from a moving camera. Based on the method used, the existing studies can be classified into different categories.

Holistic detection: is designed to detect pedestrians in images by scanning the entire frame. This approach can accurately detect humans in a static image without requiring any motion information. Various research works employ different features to detect pedestrians, such as the use of global features like edge template in [23] and local features like the histogram of oriented gradients descriptors in [24]. However, this approach has some drawbacks, as it can be easily affected by background clutter and occlusions. Nonetheless, many research works focus on modifying or extending this approach for pedestrian detection, with [25] being the most notable example, which employed optical flow and Histogram of oriented gradients.

Part-based detection: Pedestrian detection can be done using part-based approaches that utilize collections of pedestrian parts. The first step in this method is to derive part hypotheses by learning local features such as edgelet [26] and orientation features [27]. Then, these part hypotheses are combined to form the best assembly of pedestrian hypotheses. Although effective, part detection is a challenging process that requires careful consideration.

Motion-based detection: In research on pedestrian detection using onboard cameras, motion-based detection is ineffective under conditions such as fixed camerasand stationary lighting [28].

Optic flow methods: including the Lucas-Kanade method is a classical optical flow-based approach, which can be quite fast and computationally efficient, especially when implemented in a simple form. It operates at the pixel level and estimates the motion of objects based on the gradient information in the image. While it can work well for simple tracking tasks, it may not be as accurate or robust as deep learning-based methods. It might struggle with complex scenes, occlusions, and changes in lighting conditions. The Lucas-Kanade method may be suitable for real-time applications where speed is critical, and we can make certain assumptions about the scene's simplicity and the motion of objects. However, it may not be the best choice for high-precision pedestrian detection in challenging environments.

Deep learning methods: Deep learning methods have revolutionized pedestrian detection since Girshick et al [29]. proposed Region-based CNN (R-CNN) in 2014. The techniques based on deep learning can be broadly classified into two categories. The first one is a two-stage processing method, including RCNN 2014 [29] Mask RCNN 2017 [30] Fast RCNN [31] and Faster RCNN [32]. This method generates regional suggestion boxes for potential objects, followed by predictions on these boxes. The second one is a one-stage processing method, including YOLO (You Only Look Once) [33], YOLOv2 [34], YOLOv3 [35], SSD [36], RetinaNet [37], DIOU [38], YOLOv4 [39] and YOLOv5. This method directly returns the object area on the feature map and gives the final prediction result. In other words, this method is real-time object detection framework that divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell simultaneously.

Among these methods, YOLO models, especially YOLOv3 and YOLOv4, have been used for pedestrian detection. These models are designed for real-time object detection and can process frames or images very quickly, making them suitable for real-time pedestrian detection. YOLOv4 and similar deep learning-based models tend to offer superior accuracy and robustness in pedestrian detection tasks, especially in complex scenarios with occlusions, different lighting conditions, and varying poses. They are commonly used in applications where detection accuracy is crucial, such as autonomous vehicles and surveillance systems.

Pedestrian detection technology has made great progress from the original traditional machine learning to the deep neural network. However, the performance of recognition is still insufficient in conditions such as pedestrians at long distances and noisy background environments. As part of the thesis, we have tested several methods from the above-mentioned approaches that are the most efficient and suitable for our system and compared their effectiveness in determining safety by detecting pedestrians.

1.3 Thesis structure

The thesis is organized as follows. As mentioned before, this thesis consists of two main parts, driver's distraction detection and pedestrian safety. In the subsequent sections, we will present our proposed safe driving support system, highlighting its innovative features, cost-effectiveness, and non-intrusive design. Additionally, we will detail the methodology for our two core studies, which form the basis of our research. Chapter 2 outlines the structure of these modules and the implemented strategies, while Chapter 3 delves into our experiments and findings.Our overarching objective is to offer groundbreaking solutions that mitigate driver distraction and improve pedestrian safety, thus fostering safer roads for all.

Chapter 2

Proposed System Structure

2.1 Overview of proposed system

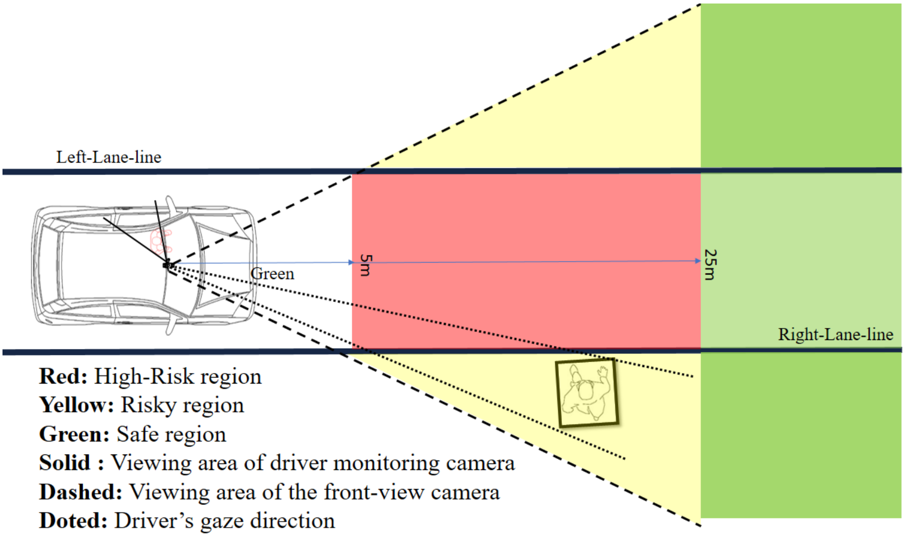

Left-Lane-line
Green
5m
25m
Right-Lane-line
Red: High-Risk region
Yellow: Risky region
Green: Safe region
Solid : Viewing area of driver monitoring camera
Dashed: Viewing area of the front-view camera
Doted: Driver's gaze direction

Figure 2.1: Overview of the proposed system

Our system consists of two modules: driver's distraction detection and pedestrian safety. The overview is shown in Figure 2.1. The first module analyzes the video stream from the driver monitoring camera and the front-view camera, determining the level of the driver's distraction. To accomplish this, it receives the video captured through the driver monitoring camera and then defines the gaze region the driver is looking at. We explored several strategies for this task. The following sections will detail each of these strategies. Simultaneously, the module receives video from the front-view camera to identify pedestrian activity. By combining the data from all road users, we can evaluate the driver's distraction level.

The second module analyzes the video feed from the front-view camera, determining the pedestrian's safety. To determine pedestrian safety, we divided the road into sections based on the level of risk for pedestrians, including High-Risk, Risky, and Safe regions, as shown in Figure 2.1. We then assessed the level of risk based on where pedestrians were located on the road.

As a result, by combining driver's distraction evaluation and pedestrian safety processing, the modules offer support for safe driving. The following sections will detail each of these modules and the tools used.

2.2 Driver's distraction detection module

In this section, we describe the challenges of the driver's distraction detection module. This module consists of two parts with parallel procedures: gaze mapping and pedestrian detection. The module analyzes the video stream from the driver monitoring camera and the front-view camera to determine the level of the driver's distraction. The first procedure, gaze mapping, is to estimate the direction of the driver's gaze, and the second procedure, pedestrian detection, is to detect the pedestrian and determine his position. This first procedure consisted of two steps: the facial feature extraction step and the gaze region classification step, as shown in Figure 2.2. The facial feature extraction step involves extracting relevant facial features from images that have gone through the video stream. Finally, the gaze region classification step predicts one of the predefined gaze regions using these features. In other words, to accomplish this task, the gaze mapping procedure receives the video captured through the driver monitoring camera and then defines the gaze region the driver is looking at. We explored several strategies for this task.

The driver distraction module consists of two procedures, with detecting pedestrians being the second one. Our system not only monitors driving situations but also keeps a close watch on the road conditions outside through a front-view camera. The second procedure receives video from the front-view camera to identify the relative position of the pedestrian. By combining the data from all road users, we can evaluate the driver's distraction level, shown in Figure 2.2. We considered three levels of distraction of the driver.

- *Safe State:* If the driver's gaze region is THE SAME as the one

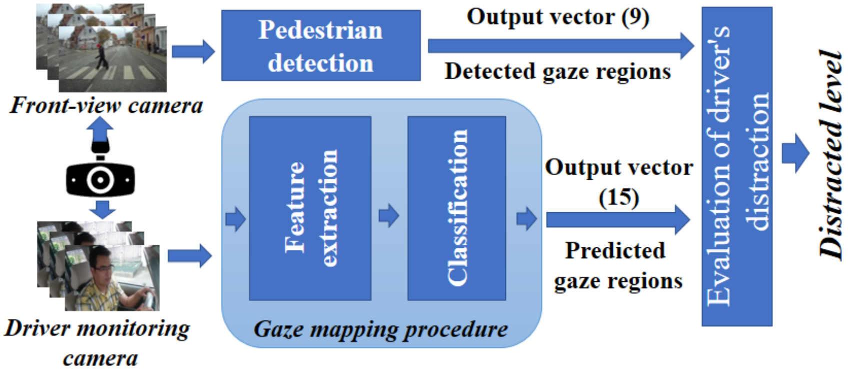


Figure 2.2: Structure of the driver's distraction detection module

the pedestrian is detected in, it is considered a "Safe" state.

- *Risky State:* If the driver's gaze region is NOT THE SAME as the one the pedestrian is detected in, BUT it is a neighboring gaze region, it is considered a "Risky" state.
- *Distracted state:* If the driver's gaze region is THE SAME as the one the pedestrian is detected in, and NOT a neighbor region EITHER, it is considered a "Distracted" state.

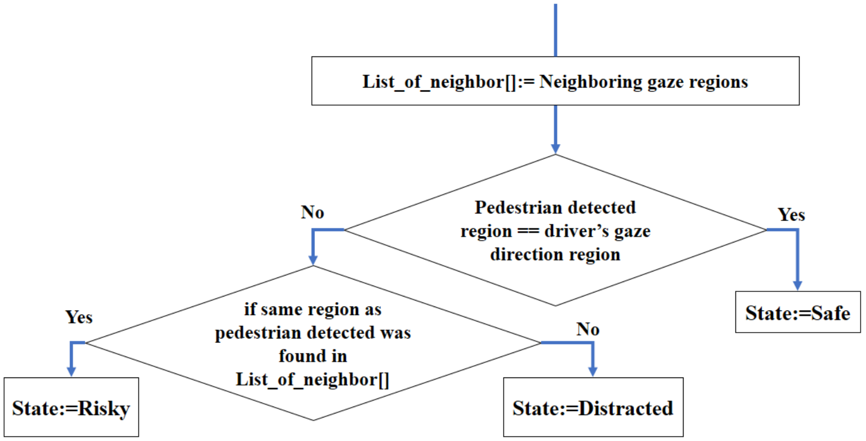

List_of_neighbor[]:= Neighboring gaze regions
Pedestrian detected region == driver's gaze direction region
No
Yes
State:=Safe
if same region as pedestrian detected was found in List_of_neighbor[]
Yes
No
State:=Risky
State:=Distracted

Figure 2.3: Flowchart of evaluation of the distraction level of the driver

Therefore, we can evaluate the driver's distraction level based on the driver's gaze direction and the pedestrian's positions, which are the outputs of the two procedures above. Our algorithm, as illustrated in Figure 2.3, identifies the levels of distraction.

In this section, we present the structure of the two procedures, driver's gaze mapping and pedestrian detection. Using these procedures, we can assess the level of distraction for the driver. As we mentioned, we implemented different strategies of gaze mapping. In the following sections, we provide more detailed experimentation and comparison of the results for each strategy of gaze mapping and pedestrian detection module.

2.2.1 Gaze mapping

In this section, we present the structure of the driver's gaze mapping. The gaze mapping procedure receives the video captured through the driver monitoring camera and then defines the gaze region the driver is looking at. We investigated different methods, such as using the MobileNet model, OpenFace with SVM classifier, and the domain adaptation method for gaze mapping. Using gaze mapping, we can assess the level of distraction for the driver, as demonstrated in Figure 2.3. This procedure comprises of two steps: the facial feature extraction step and the gaze region classification step, as shown in Figure 2.2. The facial feature extraction step involves extracting relevant facial features from images that have gone through the video stream. Finally, the gaze region classification step predicts one of the predefined gaze regions using these features.

In the following sub-sections, we provide a more detailed structure and comparison of the results for each method of gaze mapping.

2.2.1.1 Gaze mapping using MobileNet model

The first method for gaze mapping used the MobileNet deep learning method. Recently, many good deep-learning methods have been used for gaze mapping, and these studies have shown high accuracy. However, despite the high accuracy of the test data, it is very sensitive to the slight movement of the camera in the driving environment. Also, most of these methods are still expensive and difficult to use in real-time environments.

Therefore, we aimed to develop a method that is comparable to deep neural methods, and more robust with the slight movement of the camera. So, we experimented with the deep learning method and our proposed methods on the gaze mapping module to compare the training data and the real-time environments. In this study, we wanted to

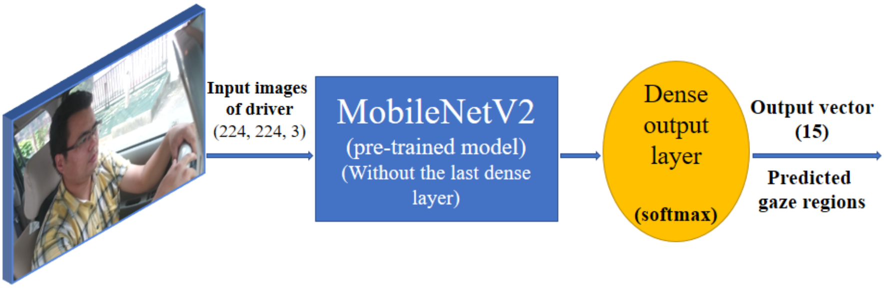


Figure 2.4: Gaze mapping using MobileNet model

demonstrate that our proposed methods are comparable to the accuracy of the deep neural method and more robust with the slight movement of the camera in the real driving scenario. Of the deep learning methods, the MobileNet model was appropriate for our study, because MobileNet needs very little computation power to run. This makes it a perfect fit for mobile devices, light systems, and computers to run without GPUs. Also, MobileNet significantly has a lower number of parameters in the deep neural network. This results in more lightweight deep neural networks. Being lightweight enables high execution speed, that is best suited for our system. We used a pre-trained MobilenetV2 [40] model without the last dense layer for the gaze mapping module. We then added a dense layer with 15 predefined gaze regions as shown in Figure 2.4. On the DGM dataset, we trained the model using several different strategies, namely the four different fine-tuning and transfer learning strategies.

2.2.1.2 Gaze mapping using OpenFace with SVM classifier

Our next proposed method involves utilizing the OpenFace methodology in conjunction with an SVM classifier for gaze mapping. Appearance-based methods use facial features to detect the driver's gaze direction. These types of studies can be classified into methods considering both eye and head orientation and methods that consider only head orientation and eye gaze. Recent gaze mapping studies indicated that consideration of both head position and eye gaze can benefit performance. Pushing this idea further, we propose an appearance-based method that, uses a combination of head position and face features. Determining face and head features from images is one of the challenges of gaze mapping. We chose the OpenFace 2.0 toolkit [41] for the feature extraction task because of its robustness and performance. The Open Face outperforms all of the baselines in both of the experiments of head position and eye gaze estimation. Specifically, the performance of gaze estimation and head pose estimation was 3.2 and 9.1 measured in the mean absolute degree error. It demonstrates state-of-the-art performance [41]. The Open Face provides face detection and extracts 68 facial features including eye gaze and head position features. Therefore, we used OpenFace, which provides gaze angles and head position features, for the gaze mapping task. The gaze angle, head position, rotation, and eye position features are recognized by analyzing the driver's face from the driver monitoring camera using the OpenFace toolkit. Also, in the gaze estimation task, we selected the SVM classifier as the classifier to classify 15 predefined regions. Our proposed system is a real-time system, so the performance speed must be high.

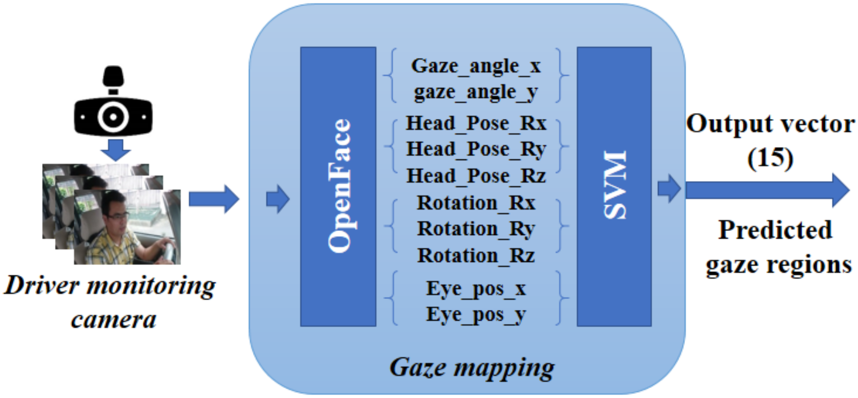


Figure 2.5: Gaze mapping using OpenFace with SVM classifier

In terms of performance speed, the SVM classifier is much faster than other classifiers, which might be more appropriate for our task with the OpenFace toolkit, Figure 2.5. We chose the SVM classifier based on the following:

- ***Performance speed:*** Our proposed system is a real-time system, so the performance speed must be high. In terms of performance speed, the SVM classifier was much faster than other classifiers [42].
- ***The accuracy of classifying gaze mapping:*** The accuracy of classifying gaze mapping was sufficiently high in mean overall

accuracy. This is shown in [42] and other studies [43] [44]. In these studies, the SVM classifier exhibited superior results to the neural network method and random forest in terms of overall accuracy and robustness. Therefore, we used the SVM algorithm to implement the classification of gaze mapping. In addition, SVM is better at classifying the extraction data of the OpenFace, as can be seen from the study of Rill Garcia et al. [44].

- ***Amount of our data:*** The OpenFace toolkit was used to extract gaze direction and head direction features from the DGM dataset. Then, we trained the SVM classifier on this dataset. Our dataset is relatively small, with few samples, making it more suitable for SVM classifiers.

We tuned the hyper-parameters to train the SVM classifier using GridSearchCV [45] from the Scikit Learn library. GridSearchCV helps to combine an estimator with a grid search preamble to tune hyper-parameters such as kernel, C, and gamma. To determine the value of parameters C and gamma for searching for the best value, we set C from 0.1 to 100 and gamma from 0.0001 to 10. According to GridSearchCV, the most appropriate parameters for the dataset extracted from the OpenFace toolkit were defined as C=10, gamma: 0.1, kernel: 'rbf'.

The DGM dataset which is more detailed in Section 3, was used to extract facial features using the OpenFace toolkit. We conducted the training of the SVM classifier using several different strategies using the features extracted by OpenFace. The features we used are as follows:

- Gaze angle: $gaze\_angle\_x, gaze\_angle\_y$,
- Head position_T: $Head\_Pose\_Tx, Head\_Pose\_Ty, Head\_Pose\_Tz$,
- Head position_R: $Head\_Pose\_Rx, Head\_Pose\_Ry, Head\_Pose\_Rz$,
- Head rotation_T: p_Tx, p_Ty,
- Head rotation_R: p_Rx, p_Ry, p_Rz,
- Eye_Position: $gaze\_0\_x, gaze\_0\_y, gaze\_0\_z$, and
- A combination of the features in binary, triple, and quadruple to determine how they affect the estimation of gaze mapping.

Therefore, we trained the SVM classifier on the dataset consisting of facial features and corresponding gaze region labels. During the execution of the gaze mapping, as shown in Figure 2.5, the facial features extracted by OpenFace are fed into the pre-trained SVM classifier, and then the pre-trained SVM classifier predicts one of the pre-defined 15 gaze regions corresponding to these features.

2.2.1.3 Gaze mapping using domain adaptation

Previous studies and proposed methods have shown a common challenge where performance tends to decline when dealing with different drivers and environments. This can be attributed to several factors such as domain disparities, insufficient data for the target driver, environmental influences, and different camera positions. Although deep learning and convolutional neural networks perform well on learned data, the results are not satisfactory for different car environments, camera positions, and domains. To overcome these challenges, domain adaptation techniques are utilized to minimize the negative effects of domain shifting, enabling the model to be applicable across different domains and environments. The proposed method has three steps, pre-processing, facial feature extraction, and gaze region classification as shown in Figure 2.6. During training, we tried two pre-processing strategies for the input feature extraction step. The first strategy involved using an image of the driver's full appearance and the environment. This allowed us to skip the face detection and face bounding box & crop step and directly train the feature extraction from the input images, as shown by line A, in Figure 2.6. In the second strategy, we specifically detected the driver's face and used it as input for the feature extraction step, as shown by line B, in Figure 2.6. The facial feature extraction step involves extracting relevant facial features from images of pre-processing step. Finally, the gaze region classification step predicts one of the predefined gaze regions using these features.

At first, we will provide a detailed description of gaze mapping using domain adaptation, including the principles of the base model, the algorithmic steps, and the mathematical aspects. This method's main theoretical underpinning is that the model is designed to address challenges related to domain shift, leveraging adversarial training and transfer learning principles for unsupervised domain adaptation in gaze mapping (DGM dataset to Columbia Cave-DB). To provide a detailed explanation, we begin by selecting the components of the proposed model structure. First, we choose a discriminative base model, as we assume that when adapting a model from a source domain to a target domain, the discriminative aspects of the model are more important than the generative aspects. Then, the choice between shared and

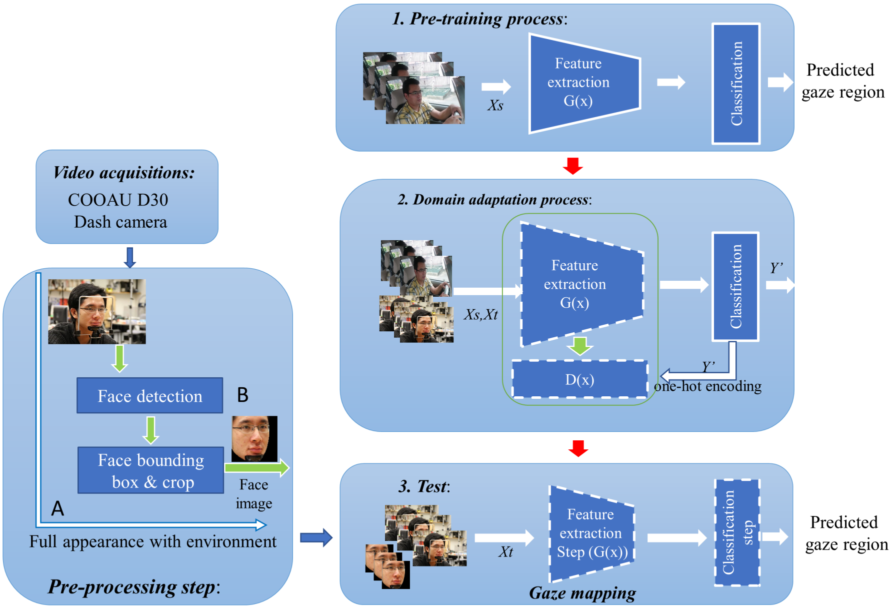


Figure 2.6: Structure of the proposed gaze mapping using domain-adaptation method

unshared weights depends on the nature of the adaptation problem. If the domains are expected to have similar characteristics, shared weights might be more appropriate. In our case, the target and source domains are quite different depending on the participating drivers' environment and facial appearance. Hence, separate sets of model parameters are used for the source and target domains. This is because unshared weights allow the model to adapt more flexibly to domain-specific characteristics, which is important when there is a significant domain shift. Therefore, we chose an unsupervised domain adaptation method with unshared weights. Moreover, adversarial loss is another important component of our proposed model. It is a crucial component of unsupervised domain adaptation, particularly in methods that leverage domain adversarial training. We used separate sets of model parameters for the source and target domains, and therefore, we chose the GAN loss as the adversarial loss for our case. By combining unshared weight and GAN loss, we assumed that the model could adapt to the specific features present in each domain while minimizing the domain shift through adversarial training.

Next, we will describe the training process of the feature extraction and classification steps, as illustrated in Figure 2.7. The training aims to

Algorithm 1: Training procedure

```
Data: Source domain images with labels, Target domain images
Result: Feature extractor network $G$, Classifier network $C$
Initialize and pre-training:
$G \leftarrow$ ResNet18 network with modified final layers (13 neurons + softmax);
$D \leftarrow$ MLP discriminator with input dimension $G$ output dimension, hidden dimension 128, and output dimension 1;
$C \leftarrow$ MLP classifier with input dimension $G$ output dimension, hidden dimension 64, and output dimension 13;
for each epoch do
    for each image $x$ in source domain do
        Compute features $f \leftarrow G(x)$;
        Encode domain label $y \leftarrow Y(x)$;
        Calculate loss based on $f$ and ground truth labels;
        Back-propagate to update $G$ parameters $\theta$;
Feature Extraction and Adversarial Training:
Freeze pre-trained feature extractor parameters $\theta$;
for each epoch do
    for each image $x$ in source and target domains do
        Compute features $f \leftarrow G(x)$;
        Encode domain label $y \leftarrow Y(x)$;
        Train discriminator $D$ to minimize loss;
        Train generator $G$ to minimize loss;
Joint Fine-tuning and Classifier Training:
Un-Freeze feature extractor parameters $\theta$;
for fine-tune epochs do
    Sample a batch from source and target domains;
    Extract features: $f_s \leftarrow G(\text{batch}_s)$, $f_t \leftarrow G(\text{batch}_t)$;
    Compute classification loss: $L_{\text{cls}} \leftarrow C(f_s).\text{loss}(\text{batch}_s)$;
    Compute domain adaptation loss: $L_{\text{adapt}} \leftarrow \text{MMD}(f_s, f_t)$;
    Combined loss: $L \leftarrow L_{\text{cls}} + \lambda * L_{\text{adapt}}$;
    Back-propagate to update both $G$ and $C$ parameters;
Output:
Fine-tuned feature extractor $G$ with domain adaptation;
Classifier $C$ trained on source domain and fine-tuned by target domain;
```

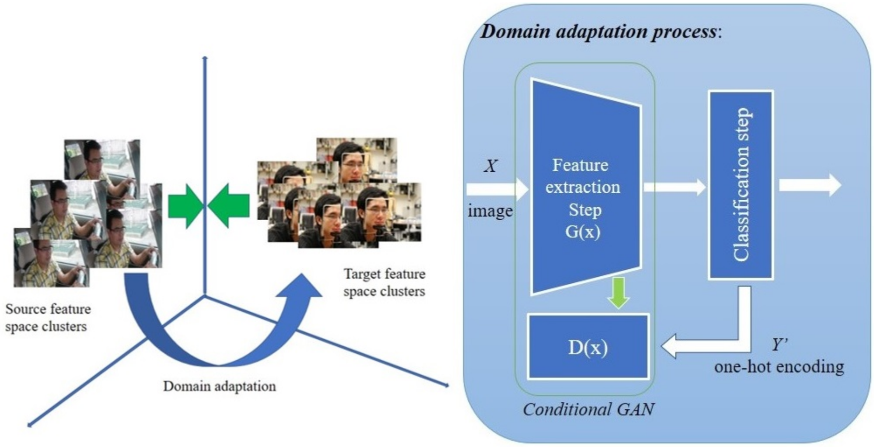

Source feature space clusters
Target feature space clusters
Domain adaptation
Domain adaptation process:
X
image
Feature extraction Step G(x)
Classification step
D(x)
Y'
one-hot encoding
Conditional GAN

Figure 2.7: An overview of domain adaptation

achieve unsupervised domain adaptation for gaze mapping, specifically from the DGM dataset to Columbia Cave-DB. When images are from different distributions, a feature extractor maps them to different clusters in the feature space. To bring these clusters closer together, a conditional generative adversarial network (CGAN) [46] is used. In detail, we utilized the ResNet18 model [47] as the backbone model. To modify the network, we replaced its final layer with a new fully connected layer consisting of 13 neurons. Additionally, a softmax layer was included on top of it. The model uses a feature extractor as a generator G(x), where x represents the input image, and an external multi-layer perceptron acts as a discriminator D(x), which determines whether the extracted feature is from the source or target domain. This classification is represented through one-hot encoding, Y(x).

During each epoch, the discriminator is optimized first, to minimize the difference between D(G(x)) and Y(x) for all x in both domains. The generator is then optimized to confuse the discriminator, to minimize the difference between D(G(x)) and Y', where Y' represents the one-hot encoding for the source domain and x is from the target domain. This process maps images from the target domain to a cluster that is closer to the cluster in the source domain's feature space. The feature extractor parameters are frozen, and the classifier is trained on the source domain. Since the feature extractor is generalized, training on the source domain can enhance performance on the target domain. Furthermore, we will explain the proposed model in terms of process. One important aspect of our method is the adversarial training procedure. We will provide a step-by-step algorithm that gives a mathematical overview of the key components and processes involved in unsupervised domain adaptation for gaze mapping, as described in Algorithm 1.

2.2.2 Pedestrian detection

The driver's distraction detection module incorporates pedestrian detection as its second procedure. In addition to monitoring driving conditions, the module employs front-view cameras to observe the road ahead. If it detects a moving object, such as a pedestrian, it pinpoints the exact predefined windshield region where the object is situated, as shown in Figure 2.2. This is crucial since distracted drivers may fail to notice pedestrians or other moving objects on the road. We implemented and tested two approaches within this procedure.

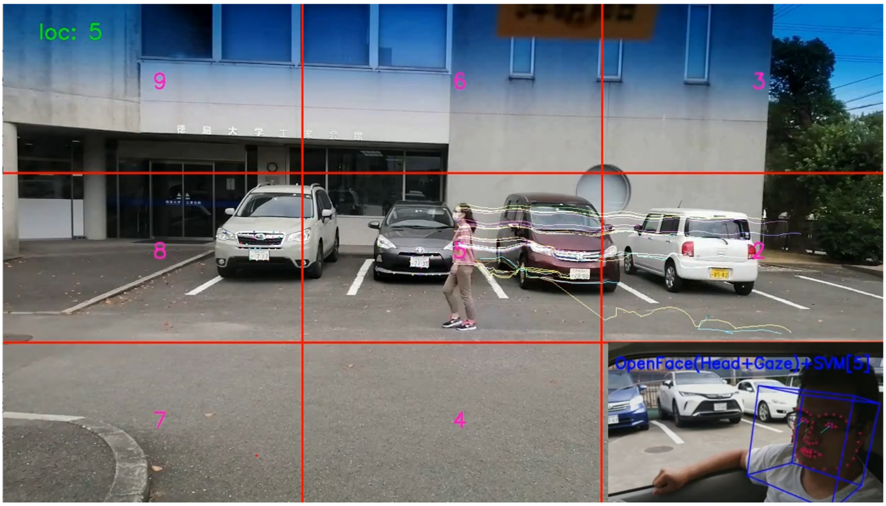


Figure 2.8: Scene of the implementation of Lucas-Kanade dense method

First, we chose the Lucas-Kanade dense optical flow method for this task because of its speed and efficiency, particularly in its fundamental form. The Lucas-Kanade dense optical flow method is a technique used in computer vision to estimate motion in a sequence of images. It's primarily used for tracking motion, but it can also be applied to detect moving objects like pedestrians. We implemented the Lucas-Kanade dense optical flow algorithm to compute the dense optical flow between consecutive frames. This algorithm estimates the motion vector for each pixel in the image. The implementation is shown in Figure 2.8. However, even though the Lucas-Kanade method shows decent performance for pedestrian detection tasks, it may have limitations in accurately detecting pedestrians, especially in challenging scenarios like occlusions, varying lighting conditions, or complex backgrounds. Therefore, a complete pedestrian detection system often integrates multiple methods and techniques to achieve robust and accurate results. More advanced approaches like deep learning-based object detection networks have shown superior performance in pedestrian detection tasks and are widely used in modern computer vision applications. Therefore, we also implemented and tested the YOLOv4 model, which is one of the best deep learning-based models. Furthermore, we examined the YOLOv4 model as it is a dependable choice for achieving high accuracy. It is frequently employed in applications where detection precision is crucial, such as autonomous vehicles and surveillance systems. The comparative performance of these methods is detailed in Section 3.3.

2.3 Pedestrian safety module

The concept of pedestrian risk regions is illustrated in Figure 2.1. Our definition is based on the research conducted by Gerónimo et al [48]. They identified three regions that are important for pedestrian safety. The high-risk region, indicated by red color, is the area where there is a high probability of collisions with pedestrians. The risky region, shown in yellow, is the area where pedestrians are likely to cross the road, but there is no imminent danger. The safe region, depicted in green, is where pedestrians are not at risk of being hit, but they must be detected in advance as they are in the path of the vehicle. The distance of these safety regions is determined by the vehicle manufacturer's tests, which show that the stopping distance of a vehicle is about 5 meters at 30 km/h and increases up to 25 meters at 100 km/h [48]. In other words, the pedestrian's relative position to the vehicle determines the risk level.

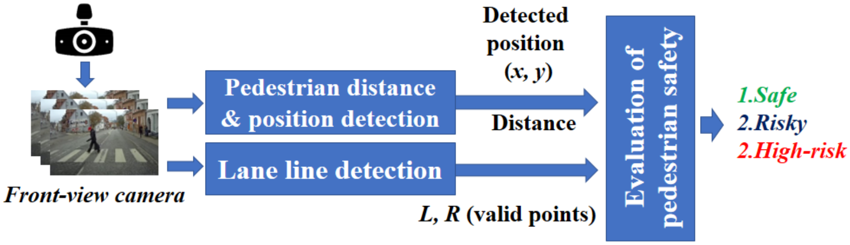


Figure 2.9: Structure of pedestrian safety module

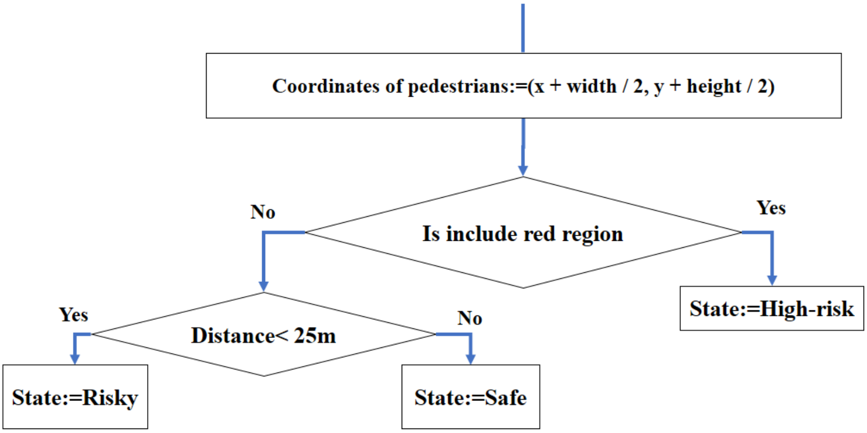


Figure 2.10: Flowchart of evaluation of the risk level of the pedestrian

According to the above concept, the lane line is a crucial key to determining the pedestrian's risk level, because all risk regions are determined by lane lines. Our pedestrian safety module relies on two procedures: lane line detection and pedestrian distance and position detection, as shown in Figure 2.9. In this module, the lane detection procedure is crucial. After thorough research, we selected Cao et al.'s[49] study since it has demonstrated high performance and reliability.

Their recognition accuracy of 98.42% surpasses deep learning methods, and their performance speed of 22.2 ms/frame is faster than traditional methods, indicating their advanced capabilities, shown in Table 2.1. Additionally, they utilized a vanishing point algorithm to detect road boundaries, which produced favorable results in detecting unstructured roads.

The lane line procedure analyzes the video feed from the front-view camera to identify the lane lines and highlight them in red up to a distance of 25 meters. To achieve this, we convert the distorted image and apply the superposition threshold algorithm to detect edges, then obtain an aerial view of the lane by extracting a region of interest and applying the inverse perspective transformation. Then, we fit the curves of the lane lines using the random sample consensus algorithm and a third-order B-spline curve model. Finally, we evaluate the fit and calculate the curvature radius of the curve. As a result, we determine the lane line of the road and extract the set of points of the section defined in red.

Simultaneously, as the front-view camera captures the video stream, the pedestrian distance and position detection procedure analyzes it

Table 2.1: The comparison of statistics in algorithms performance

Methods	**Algorithm**	**Average Detection Accuracy (%)**	**Average Processing Time (ms)/ Frame**
Traditional	Spatial Ray Features	94.40	45.0
	Improved Hough Transform	95.70	65.4
Deep Learning	FastDraw Resnet	95.00	65.3
	ConvLSTM	97.25	42.0
Cao et.al's Algorithm		**98.42**	**22.2**

to detect pedestrians and their distance from the vehicle. However, estimating object distance using a single camera can be limited as it lacks depth information, resulting in lower accuracy compared to stereo or depth-sensing cameras. Nevertheless, there are alternative methods to estimate distances using a single camera. In the challenge of estimating pedestrian distance using a front-view camera, we utilized the scale estimation method. This method involves using a known reference object with a known size within the scene to estimate distances to other objects in the same scene. We also used YOLOv4 to create bounding boxes for pedestrians, extract their coordinates, and determine their position. The pedestrian distance was also determined using the scale estimation method. By analyzing the pedestrian's distance and position in relation to lane lines, we can determine their risk level. This evaluation is based on the two procedures outlined above and is illustrated in Figure 2.10. Our algorithm identifies three distinct levels of pedestrian risk.

Chapter 3

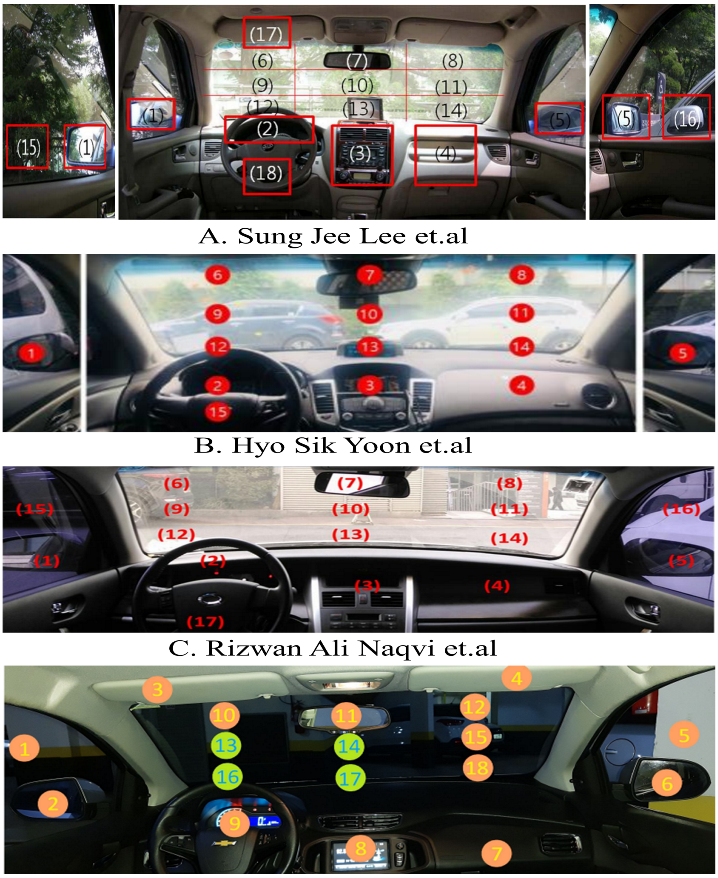
Datasets and Experiments

In this section, we present the evaluation of the procedures, gaze mapping and pedestrian detection and safety, separately and their combination. In other words, we show the evaluation of the state of the driver's distraction based on two inputs. As we explained in Section 2, we implemented different strategies for each method of the gaze mapping module in the SDSS system. Moreover, the performance of pedestrian detection methods will be evaluated on real-world driving recordings. The following sections provide a more detailed comparison of the results for each method of the gaze mapping module.

To conduct our experiments, we utilized two different datasets, The Driver Gaze Mapping (DGM) and Cave-DB [54]. In this section, we present our experiments and results on our proposed strategies using these datasets. Additionally, we conduct experiments on the DGM dataset, which includes different camera positions. It explores the possibility of adapting to different camera positions in the same domain for self-calibration tasks. Furthermore, we provided an analysis of the results obtained from the proposed methods and strategies. This includes a discussion of the implications of these results, a comparison with existing methods, and the limitations of our study. The following sections detail each dataset's characteristics and dataset-related information, as well as our experiments and results.

3.1 Evaluation metrics

Our research involved measuring the accuracy of gaze mapping through two methods: Strictly correct estimation rate (SCER) and Loosely correct estimation rate (LCER). SCER and LCER are the two metrics used to evaluate gaze mapping accuracy. SCER measures the ratio of



A. Sung Jee Lee et.al

B. Hyo Sik Yoon et.al

C. Rizwan Ali Naqvi et.al

D. Yafei Wang et.al

Figure 3.1: The gaze regions of similar studies A. [50] B. [51] C. [52] D. [53]

strictly correct frames to the total number of frames, where the estimated gaze region perfectly matches the ground truth gaze region. On the other hand, LCER measures the ratio of loosely correct frames to the total number of frames, where the estimated gaze region is within the ground truth gaze region and its neighboring regions. SCER and LCER are mostly used in our field research. In our study, the accuracy of gaze mapping also measured based on the Strictly correct estimation rate (SCER) and the Loosely correct estimation rate (LCER).

$$SCER = \frac{NumberOfStrictlyCorrectedFrames}{TotalNumberOfFrames} \tag{3.1}$$

SCER measures the ratio of the number of frames where the estimated gaze region is strictly correct (equivalent to the ground truth gaze region) to the total number of frames.

Strictly Correct Frame: A frame is considered strictly correct if the estimated gaze region is precisely equal to the ground truth gaze region.

$$LCER = \frac{NumberOfFramesWithEstimatedRegionIn(GT \cup N)}{TotalNumberOfFrames} \tag{3.2}$$

LCER measures the ratio of the number of frames where the estimated gaze region is loosely correct (within the ground truth gaze region and neighboring regions) to the total number of frames.

Loosely Correct Frame: A frame is considered loosely correct if the estimated gaze region is placed within the ground truth gaze region or in one of the neighboring regions. The numerator now represents the count of frames where the estimated gaze region is in the union of the ground truth gaze region $GT \cup N$ and the set of neighboring regions (N).

3.2 Datasets

In this section, we will provide the datasets used in our study and details about those datasets. As part of the study, we created a dataset with 15 gaze regions for gaze mapping. For absolute clarity, let's refer to this dataset as DGM. The feature of the dataset is that it is prepared by capturing images of the driver from different camera positions. This allows us to determine which camera positions are more effective in mapping the driver's gaze within the scope of the study. We also created a new dataset adapted to our research environment using an open dataset Cave-DB. We used this dataset to compare our results with those of other researchers. We also used this dataset to experiment with one of our proposed methods, the domain adaptation method. The following subsections describe the datasets and their details.

3.2.1 DGM dataset

Through our research, we aimed to determine the minimum number of gaze regions necessary for safe driving, as well as which specific regions should be targeted based on previous studies. After reviewing multiple studies on gaze mapping, we discovered that most studies, including ours, had similar divisions for gaze regions, despite different approaches. Our analysis of several studies revealed that gaze regions ranged from 9

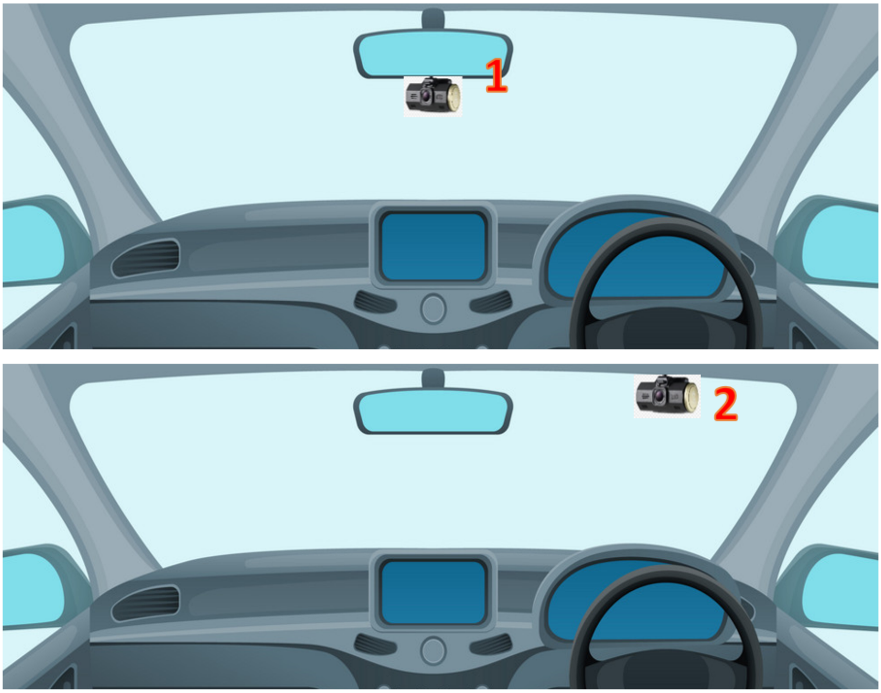


Figure 3.2: Camera positions: (1) bottom of the rear mirror and, (2) top-front of windshield

to 18, with 15 being the most commonly used as shown in Figure 3.1. These gaze regions are considered relevant for safe driving in literature [11]-[17], [50][52][55][56][57][58][59][60][61]. We selected the gaze regions based on the gaze regions of related studies. However, we collected 16 gaze regions, one of which was not considered important in most of the previous studies, so we withdrew it and created this dataset with 15 gaze regions. Therefore, region 10 is numbered without including it. We also considered the corresponding neighboring regions of each gaze region as shown in Table 3.1.

The DGM dataset was used for the gaze mapping task. This dataset features 15 distinct gaze regions and data collected from two different camera positions, as described in Figure 3.2. The dataset comprises the driver's gaze and information about the driving environment.

The 15 predefined gaze regions, illustrated in Figures 3.3 and 3.4, include the gaze region on the windshield, left and right-side mirrors, and left and right-side windows (regions 1-9). The dataset was built using images of drivers who gazed at predefined 15 regions in the vehicle. We captured the data as the vehicle went to different locations, such as

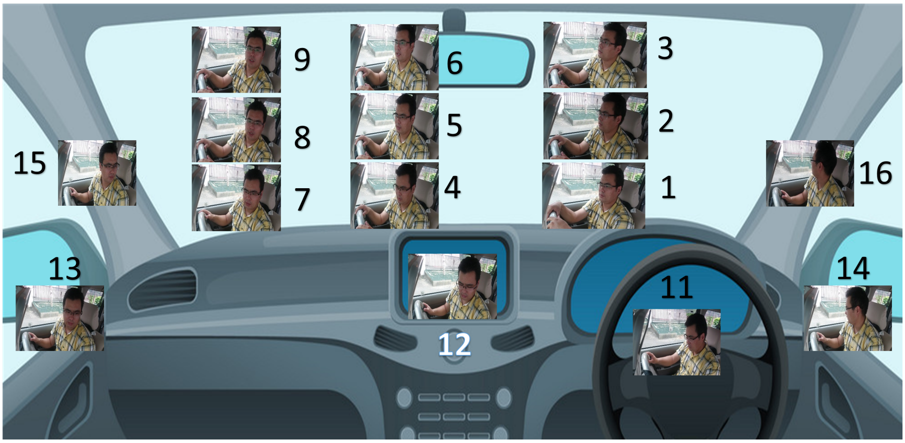


Figure 3.3: Our predefined 15 gaze regions using camera position 1

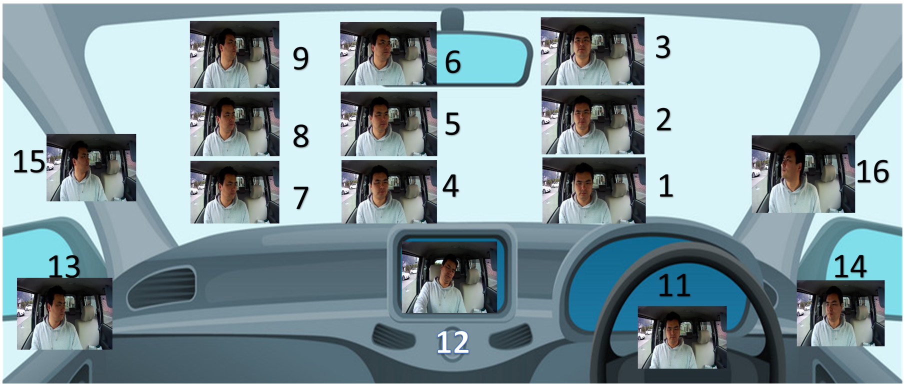


Figure 3.4: Our predefined 15 gaze regions using camera position 2

university campus roads and parking lots, in the morning, afternoon, and night to get images at different times of the day using a simple COOAU-D30-1080P dual dash camera. As the drivers gazed at the 15 predefined regions, they acted naturally, with no restrictions on changes in the head pose or other movements.

The dataset includes 12,425 images with 15 labels using camera position 1. Also, we collected 14,200 images with the same labels using camera position 2, as described in Figure 3.4.

3.2.2 Open dataset Cave-DB

To ensure fairness in our comparison, we created a new dataset using the open dataset Columbia gaze dataset CAVE-DB. This was done as

Table 3.1: Gaze regions and neighbor regions of each region

№	Gaze regions	Neighbors
1	1 windshield	2, 4, 5, 11, 14, 16
2	2 windshield	1, 3, 4, 5, 6, 14, 16
3	3 windshield	2, 5, 6, 16
4	4 windshield	1, 2, 5, 7, 8, 11, 12
5	5 windshield	1, 2, 3, 4, 7, 8, 9, 6
6	6 Rearview mirror	3, 5, 9
7	7 windshield	4, 5, 8, 13, 15
8	8 windshield	4, 5, 7, 9, 6, 13, 15
9	9 windshield	5, 8, 6, 15
10	11 Dashboard	1, 4, 12, 14
11	12 Music/radio	4, 7, 11, 13
12	13 Left side mirror	7, 8, 15
13	14 Right side mirror	1, 2, 11, 16
14	15 Left side window	7, 8, 9, 13
15	16 Right side window	1, 2, 3, 14

previous studies [52][55][50] which also evaluated their methods using SCER and LCER through CAVE-DB. Also, it enabled us to apply unsupervised classification to different domain shifts. The CAVE-DB contains a large gaze database of 56 individuals with 5880 images that vary in head poses and gaze directions. There are 105 gaze directions as 5 head poses with 21 gaze directions per head pose.

From the database, we chose 13 gaze direction images considering the environment of the DGM dataset. The examples of images with gaze regions are shown in Figure 3.5.

3.3 Gaze mapping evaluation

The most important part of the driver's distraction detection module is the gaze mapping procedure. This is because the performance of the entire driver's distraction detection module depends on the accuracy of the gaze mapping procedure. In this section, we have provided a detailed analysis of experimental results obtained from the proposed methods and strategies. This includes a discussion of the implications of these results, a comparison with existing methods, and the limitations of our study. As explained in Section 2, we implemented different strategies for gaze mapping in the driver's distraction module. The following sections

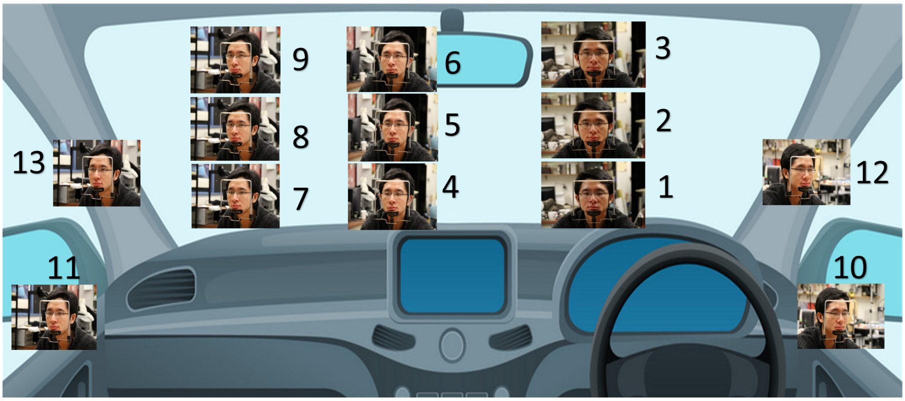


Figure 3.5: Sample images selected from CAVE-DB

provide a more detailed experiment and comparison of the results for each strategy of the gaze mapping module and the implementation of the methods used in other modules and their comparison.

3.3.1 Gaze mapping using MobileNet model

First, we fine-tuned the pre-trained MobileNet trained on the ImageNet dataset [62] with our DGM dataset. Table 3.2 shows the results of the MobileNet models using different strategies. We evaluated the performance of each model using 180 drivers' face images for each gaze region. The MobileNet model was trained using four different fine-tuning and transfer learning strategies using the DGM dataset. The first strategy is to train only the last classifier layer of the MobileNet model, while the second strategy is to train the last 30 layers of the model including the classifier layer. We trained the last 50 and 70 layers of the model including the classifier layer for the third and fourth strategies, respectively. We noted that by increasing the number of trainable layers from 30 to 50, the accuracy of the training was improved. However, setting trainable layers to 70, the accuracy was lower as shown in Table 3.2. In addition, we also tested the model with 80 trainable layers, but the result was lower than others. Therefore, we do not show the result of the model with 80 trainable layers in our results. The model with 50 trainable layers achieved the best result at 97.45% accuracy.

3.3.2 Gaze mapping using OpenFace with SVM

Secondly, we used the OpenFace toolkit to extract gaze and head features, then classified the driving gaze direction into 15 pre-defined gaze regions

Table 3.2: The MobileNet models using different strategies results

Strategies	**Accuracy**	**Precisions**	**Recall**	**F1 score**
Only classification layer	92.64%	92.97%	92.64%	92.60%
Last 30 layers	93.56%	94.16%	93.56%	93.56%
Last 50 layers	*97.45%*	*97.46%*	*97.45%*	*97.45%*
Last 70 layers	96.86%	96.91%	96.89%	96.90%

using the SVM classifier. We evaluated the performance of each strategy using the same data used in the evaluation of the MobileNet model. We tested combining the features one by one and with the other features in binary, triple, and quadruple to determine how they affect the recognition of gaze mapping using different camera positions. The average accuracy of 15 gaze regions for each camera position is shown in Table 3.3-3.6. The experiment using camera position1 showed that the gaze angle, head rotation_R, eye position, and head position-R features were effective. Other features, such as Eye_Position with Z and Head_Rotation_T, were somewhat effective but less effective than other features. The head rotation_R feature achieved the highest accuracy of 73.20%.

Table 3.3: The SVM Classifier Using Single Features results

№	**Strategies**	**Accuracy /%/**	
		Camera position 1	**Camera position 2**
1	Gaze angle	54.12%	75.75%
2	Head position_R	70.31%	77.12%
3	Head position_T	31.18%	32.06%
4	Eye position	56.85%	69.35%
5	Eye position WO-Z	60.17%	70.85%
6	Head_rotation_R	73.20%	86.80%
7	Head_rotation_T	49.06%	52.20%

Note:
Eye position WO-Z = *Eye position without Z-axis values*

The test results also showed that the head rotation_R feature was the most effective, and the test results using camera position 2 were better than camera position 1. Furthermore, Table 3.3 shows that the effectiveness of all features has increased using camera position 2.

After that, we conducted an experiment to determine the effectiveness of these features when combined with other features for gaze mapping. In this experiment, less effective features from the previous experiment were excluded.

Table 3.4: The SVM Classifier Using Dual Features results

№	Strategies	Accuracy /%/	
		Camera position 1	**Camera position 2**
1	Gaze angle + Head position_R	73.87%	78.31%
2	Gaze angle + Head rotation_R	74.12%	87.20%
3	Gaze angle + Eye position	56.31%	73.46%
4	Gaze angle + Eye position WO-Z	55.18%	74.96%
5	Head position_R + Head rotation_R	77.25%	86.04%
6	Head position_R + Head rotation_T	50.19%	54.14%
7	Head position_R + Eye position	75.72%	83.28%
8	Head position_R + Eye position WO-Z	75.79%	83.69%
9	Head rotation_T + Eye position	45.16%	51.74%

Note:
Eye position WO-Z = *Eye position without Z-axis values*

Table 3.4 shows the results of the 9 combinations with the highest results. Of these, all combinations of the Head_Rotation_R feature were effective, and the combination of gaze angle and head rotation_R achieved an accuracy of 87.20% using camera position 2 for the best results.

Through our experiments, we found that utilizing head rotation, head position, gaze angle, and eye position WO-Z features to gaze mapping is highly effective. While relying solely on head position to estimate the driver's gaze direction provides advantages such as detecting gaze mapping even when detailed facial features cannot be determined, our observations indicate that combining head (rotation and position) and gaze (gaze angle and eye position) features is a more effective approach.

Furthermore, we experimented with combining the features triple and quadruple to determine how they affect the recognition of gaze mapping. The head rotation features alone and in combination with other features were all better than the current best results. Of these, the gaze angle and the Head_position_R, and a combination of the Head_Rotation_R feature, achieved an accuracy of 92.80% for one of the best results, in

Table 3.5: The SVM Classifier Using Triple Features results

№	Strategies	Accuracy /%/	
		Camera position 1	**Camera position 2**
1	Gaze angle + Head position_R + Eye position	83.74%	90.27%
2	Gaze angle + Head position_R + Eye position WO-Za	87.57%	91.21%
3	Gaze angle + Head position_R + Head rotation_R	86.80%	92.80%
4	Gaze angle + Head position_R + Head rotation_T	53.37%	55.02%

Note:
Eye position WO-Z = *Eye position without Z-axis values*

Table 3.6: The SVM Classifier Using Quadruple Features results

№	Strategies	Accuracy /%/	
		Camera position 1	**Camera position 2**
1	Gaze angle + Head position_R + Head rotation_R + Eye position	85.12%	91.63%
2	Gaze angle + Head position_R + Head rotation_R + Eye position WO-Z	87.42%	92.85%

Note:
Eye position WO-Z = *Eye position without Z-axis values*

Table 3.5. Also, a quadruple combination of features gaze angle, head position_R, head rotation_R, and Eye position WO-Z is the best accuracy which is 92.85%, shown in Table 3.6.

Experimental results show that Camera Position 2 (top-up windshield) is more effective than Position 1 (bottom of rear mirror) for gaze estimation tasks. This is because the driver's eye gaze and head position are more clearly visible from camera position 2. For the eye position feature, it was experimentally determined that the value of the Z-axis was not considered to be more effective. Therefore, we conducted separate experiments without the inclusion of Z-axis values and denoted eye position WO-Z.

Moreover, we chose the best strategy of each method and evaluated

Table 3.7: Evaluation results of the best strategies for each method

G/R	MobileNet 1	OpenFace+SVM 1	OpenFace+SVM 2
1	3/3	2/3	2/3
2	3/3	2/3	2/3
3	3/3	3/3	3/3
4	2/3	3/3	3/3
5	2/3	3/3	3/3
6	3/3	3/3	3/3
7	2/3	3/3	3/3
8	2/3	3/3	3/3
9	3/3	1/3	2/3
11	***0/3***	3/3	3/3
12	***0/3***	3/3	3/3
13	3/3	3/3	3/3
14	3/3	3/3	3/3
15	3/3	1/3	2/3
16	3/3	2/3	2/3
Overall:	77.7%	84.4%	88.8%

Note:
G/R = *Gaze regions*
MobileNet 1 = *strategy of the gaze mapping using the MobileNet model with last 50 trainable layers*
Openface + SVM 1: = *strategy of the Openface with SVM classifier using gaze angle, head position_R, and head roation_R (camera position 2)*
Openface + SVM 2: = *strategy of the Openface with SVM classifier using gaze angle, head position_R, head roation_R and eye position WO-Z features (camera position 2)*

the performance of each strategy using a real driving video. During the video, the driver primarily focuses on gaze region 2 and subsequently looks at each gaze region three times. This includes looking at gaze region 1, returning to gaze region 2, and then returning to gaze region 1 again before repeating this pattern for all gaze regions. Table 3.7 shows the results of these evaluations. The strategy of the gaze mapping using the MobileNet model (with the last 50 trainable layers) in Table 3.7 predicted all of the gaze regions except 11 and 12 with a high percentage. Two strategies of gaze mapping using the OpenFace with SVM classifier predicted all gaze regions. The MobileNet model performed better than the OpenFace with SVM classifier when using test data (180 drivers' face images for each gaze region). However, when using the real driving video, the average accuracy of both strategies using the OpenFace with SVM classifier was higher than the strategy using MobileNet. Among these compared strategies, the Openface with SVM classifier using gaze angle, head position_R, head rotation_R and eye position WO-Z features had the highest performance accuracy of 88.8%. Also, for gaze mapping method using the MobilNet model, we noticed in this evaluation that slight camera movements greatly affect the results. Therefore, we chose the OpenFace with SVM classifier to compare with similar existing studies.

3.3.3 Gaze mapping using domain adaptation method

In this section, we have provided a detailed analysis of experimental results obtained from the gaze mapping using the domain adaptation method. This includes a discussion of the implications of these results, a comparison with existing methods, and the limitations of our study. In this, we prepared the source and target datasets in the following ways: on different drivers in the same environment, on the same driver in different environments, and on different drivers in different environments, as shown in Figure 3.6. As a result, domain adversarial training was performed on the above differently trained datasets. As a result, we determined how different drivers, different environments, and different environments and different drivers affect the results of gaze estimation methods using domain adaptation. Also, during domain adaptation, we determined which of the driver's full appearance with environment images and face images were effective for adaptive training.

To conduct our experiments, we utilized two different datasets, DGM and Cave-DB. In the training process, we utilized the DGM dataset as the source domain. We trained on this datasets and subsequently adapted and tested it to the Cave-DB dataset as the target domain. As a result of the training described in Section 2, our proposed method shown in Figure 2.6 is prepared for testing on the target domain. In the pre-processing step, there are two modes available - full appearance image (Strategy A) and face image (Strategy B), mentioned in Section 2. So, in this section, we will present and analyze the experimental results of strategies of gaze mapping using the domain adaptation method. Additionally, we conduct experiments on the DGM dataset, which includes different camera positions. It explores the possibility of adapting to different camera positions in the same domain for self-calibration tasks. Furthermore, we provided an analysis of the results obtained from the proposed model. This includes a discussion of the implications of these results, a comparison with existing methods, and the limitations

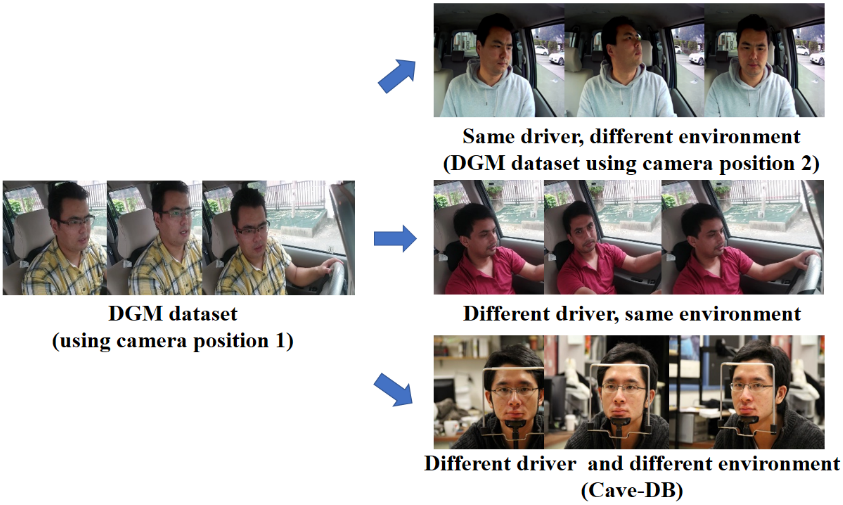

Same driver, different environment
(DGM dataset using camera position 2)
DGM dataset
(using camera position 1)
Different driver, same environment
Different driver and different environment
(Cave-DB)

Figure 3.6: Prepared datasets and domain adaptation versions

of our study.

Implementation details: We experimented with gaze mapping using the domain adaptation method and trained the model with specific parameters in both the source and target domains. For the feature extractor, the learning rates were set to 0.001 in the source domain and 0.0005 in the target domain. The classifier's learning rate was set to 0.001 in both domains. We set the adversarial loss weight and domain classifier weight to 0.1. The training was done with a batch size of 64 and 30 epochs. We initialized the feature extractor with a pre-trained model and used the Adam optimizer. We provided a step-by-step algorithm adversarial training in the following section. For more details on the training process, please refer to Algorithm 1.

Strategy A experiment: As shown in Figure 3.6, we organized domain adaptation training in 3 different versions using the driver's full appearance with environment images (Strategy A). First, we trained the DGM dataset using camera position 1 as the source domain, and camera position 2 as the target domain, using sets of datasets as shown in Table 3.8.

In this experiment, we explored the possibility of learning from each other between datasets with the same driver or facial features but different camera positions. Based on the results, the average accuracy was 85%. In the experiment, it is evident from Figure 3.7 that there is significant confusion between gaze regions 6 and 9, as well as between gaze regions 7 and 8. Furthermore, it can be observed that there is

Table 3.8: Amount of datasets used in domain adaptation versions

Domain adaptations	**Source**	**Target**	**Test**
DGM-1 to DGM-2	12285 images with 13 labels	3900 images with 13 labels	1300 images with 13 labels
DGM-1 to DGM-1	12285 images with 13 labels	3900 images with 13 labels	1300 images with 13 labels)
DGM-1 to Cave-DB	12285 images with 13 labels	3900 images with 13 labels	1300 images with 13 labels

Note:
DGM-1 = *DGM dataset using camera position 1*
DGM-2 = *DGM dataset using camera position 2*
DGM-1 to DGM-1 = *across same environment, different drivers*

some confusion in regions with low head movement. Also, a small of confusion was formed between gaze regions 1 and 2, and gaze regions 8 and 11, which are regions that can be moved by the movement of the eyeball. This suggests a risk of confusion between gaze regions that require minimal head movement and require small changes in gaze direction. Although the confusion was between the aforementioned gaze regions, the feasibility of self-calibration was demonstrated using the domain adaptation method across different camera positions within the same domain.

Secondly, we trained the DGM dataset using camera position 1 as the source domain, and a different driver with the same environment as the target domain, using sets of datasets as shown in Table 3.8. In this experiment, we aimed to determine the adaptive performance of different domains in the same environment. According to the results of the experiment, the performance of each gaze region demonstrated that the minimum accuracy was 76% or more, and the average accuracy was 88.76%. This indicates that serious confusion has not occurred in each region. Also, it can be seen from Figure 3.8 that the resulting confusion is usually observed with the neighboring gaze region.

Finally, we conducted an experiment where we used the DGM dataset as the source domain and the Cave-DB dataset as the target domain, using sets of datasets as shown in Table 3.8. The purpose of this experiment was to demonstrate how our proposed model can adapt to different domains and environments. The results showed that the target domain was classified with reasonable accuracy, except for gaze region 5 which was mis-classified as neighboring gaze region 4. Apart from this, the results were reasonable, with an average accuracy of 81.38%,

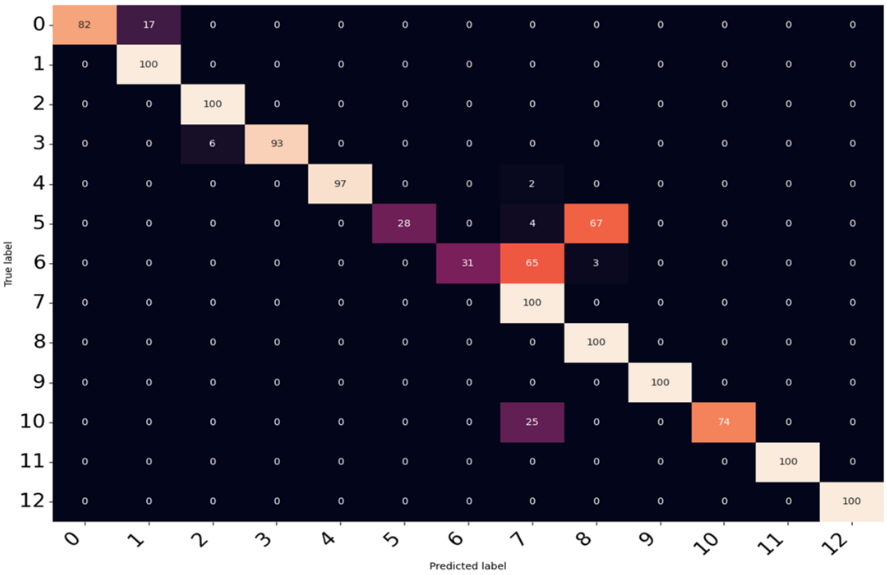


Figure 3.7: Confusion matrix of strategy A on the same driver, different environment

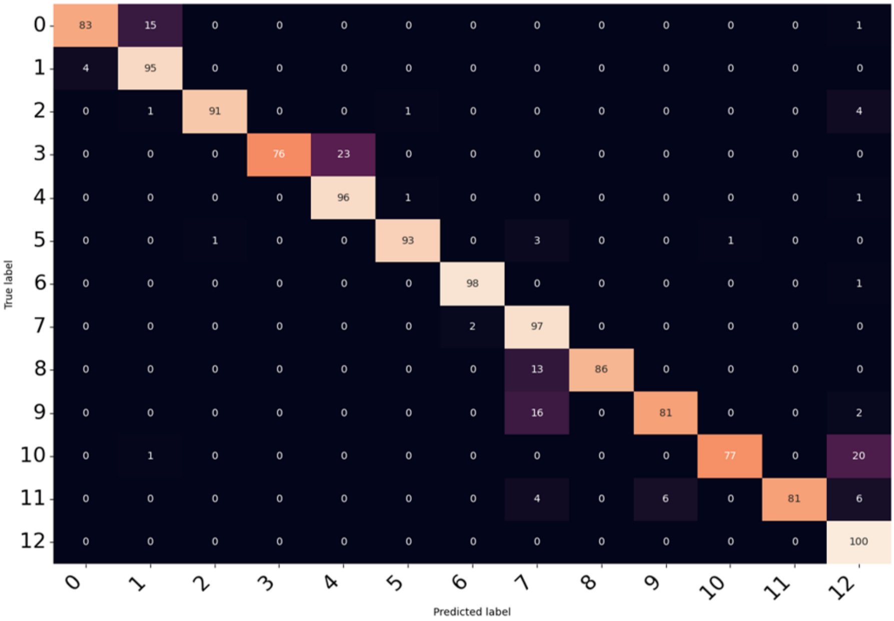


Figure 3.8: Confusion matrix of strategy A on different drivers, same environment

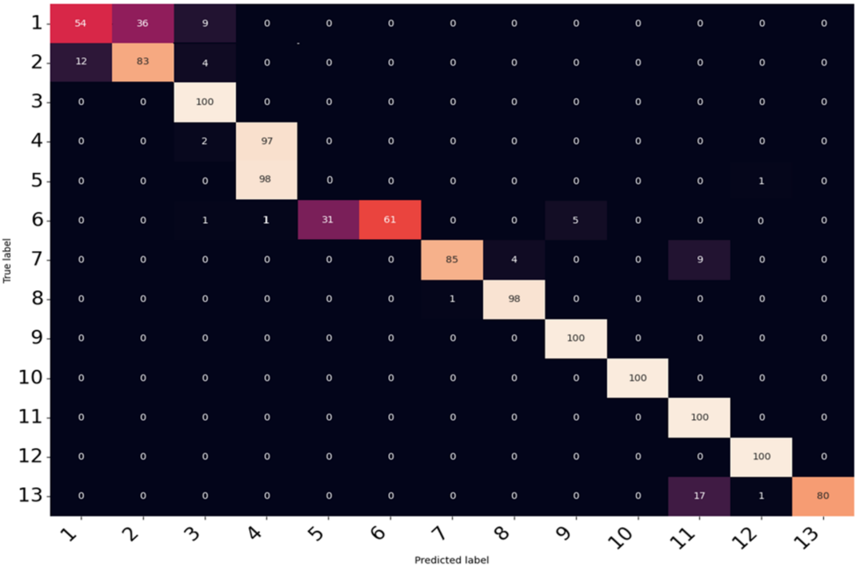


Figure 3.9: Confusion matrix of strategy A on different drivers, different environments

as shown in Figure 3.9.

Strategy B experiment : In this experiment, we trained the DGM dataset as the source domain and the Cave-DB dataset as the target domain by strategy B of pre-processing which uses a face image. The average SCER accuracy was 93.53% and the LCER rate was 98.9%, as shown in Figure 3.10.

Based on the results, Strategy B proves to be more effective than Strategy A. The average SCER accuracy rate of Strategy B is 12.15% higher compared to Strategy A which uses the driver's full appearance image. Moreover, the experiment's findings indicated that there is more confusion when transitioning between gaze regions that require only slight head and eye movements, such as gaze regions 1 and 2. However, there seems to be less confusion when transitioning between gaze regions that require more significant head and eye movements. For example, the gaze regions of side mirrors can be mentioned.

Then, we tested on the DGM dataset, where camera position 1 was the source domain and camera position 2 was the target domain. The accuracy of strategy B of pre-processing which uses a face image was reasonable. On average, the accuracy of the SCER was 94.85%, as illustrated in Figure 3.11. This indicates that Strategy B is also more efficient than Strategy A, with an average SCER accuracy rate that is

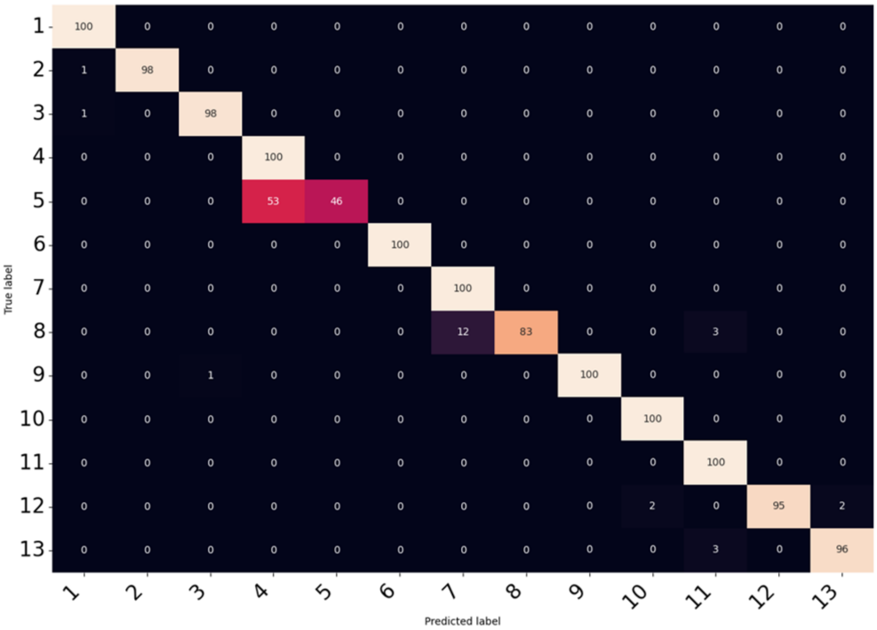


Figure 3.10: Confusion matrix of strategy B on different drivers, and different environments

Table 3.9: Performance results on domain adaptation versions

Training versions	**Full-appearance image**		**Face image**	
	SCER	LCER	SCER	LCER
DGM-1 to DGM-2	85.00%	98.80%	94.85%	99.23%
DGM-1 to DGM-1	88.76%	96.23%	-	-
DGM-1 to Cave-DB	81.38%	96.69%	93.53%	98.90%

Note:
DGM-1 = *DGM dataset using camera position 1*
DGM-2 = *DGM dataset using camera position 2*
DGM-1 to DGM-1 = *across same environment, different drivers*

9.8% higher. As a result, strategy B proved to be more effective on the above two tasks.

During the above three domain adaptation experiments, shown in Table 3.9, various findings were observed. Firstly, it was discovered that accurate gaze mapping on different drivers can be performed using domain adaptation. Secondly, the position of different cameras in the same domain can self-calibrate. Additionally, the experimental results show that the proposed method can reduce gaze mapping errors. The

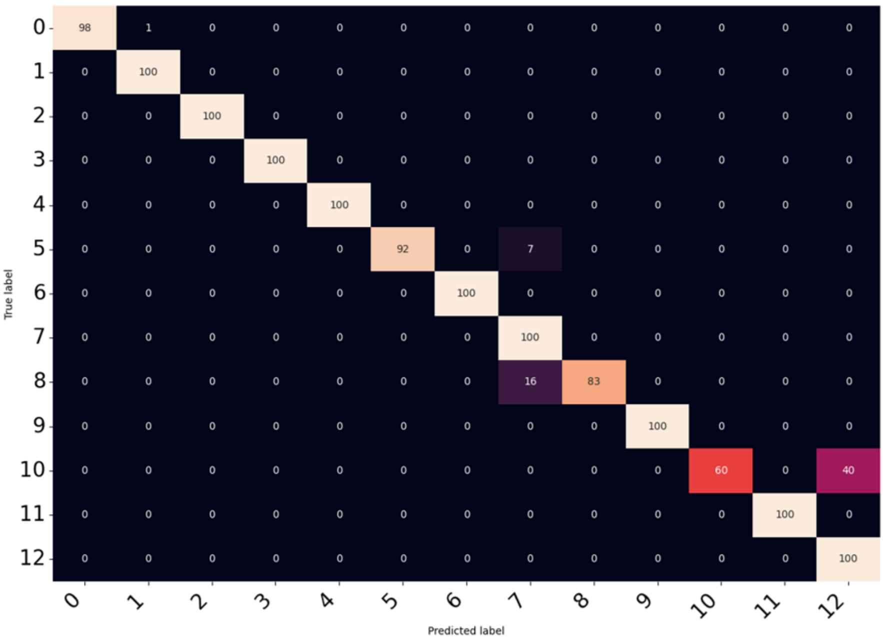


Figure 3.11: Confusion matrix of strategy B on the same driver, different environment

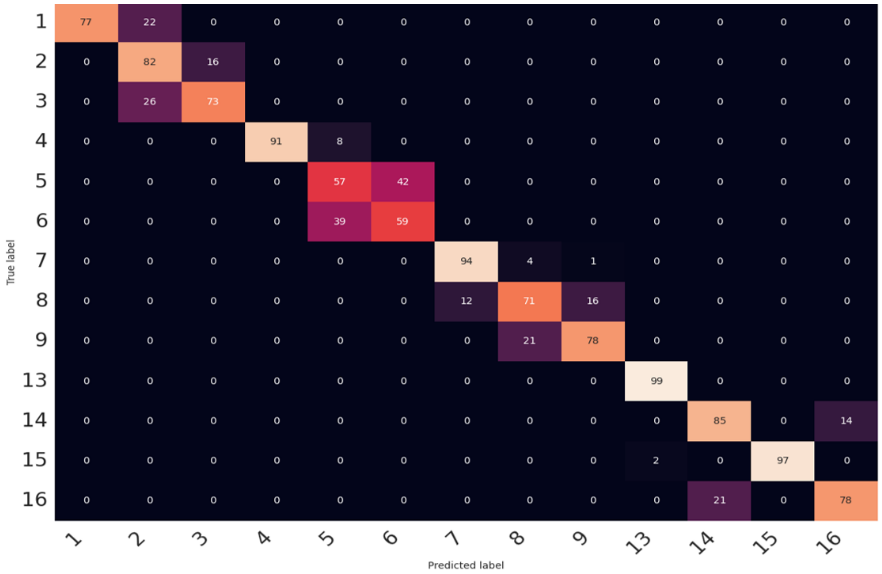
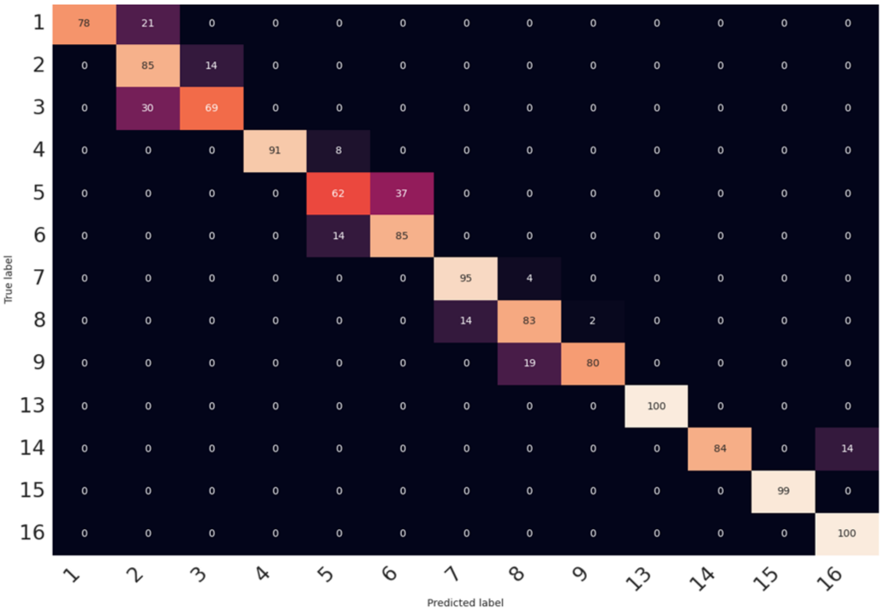


Figure 3.12: Confusion matrix of the strategy using gaze angle, head position R and head rotation_R features



	1	2	3	4	5	6	7	8	9	13	14	15	16
1	78	21	0	0	0	0	0	0	0	0	0	0	0
2	0	85	14	0	0	0	0	0	0	0	0	0	0
3	0	30	69	0	0	0	0	0	0	0	0	0	0
4	0	0	0	91	8	0	0	0	0	0	0	0	0
5	0	0	0	0	62	37	0	0	0	0	0	0	0
6	0	0	0	0	14	85	0	0	0	0	0	0	0
7	0	0	0	0	0	0	95	4	0	0	0	0	0
8	0	0	0	0	0	0	14	83	2	0	0	0	0
9	0	0	0	0	0	0	0	19	80	0	0	0	0
13	0	0	0	0	0	0	0	0	0	100	0	0	0
14	0	0	0	0	0	0	0	0	0	0	84	0	14
15	0	0	0	0	0	0	0	0	0	0	0	99	0
16	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 3.13: Confusion matrix of the strategy using gaze angle, head position_R, head rotation_R and Eye position WO-Z features

findings also demonstrate that the proposed method can reduce the gaze mapping error of the pre-trained adapted model, and even perform better on different drivers (cross-subject) and environments (different camera positions). In addition, when analyzing the results of the three confusion matrices, it was seen that the model is very stable only in domain transition without environmental change. These results underscore the effectiveness of our method in adapting to different domains. On the other hand, it was observed that it is comparably weak for the same domain and different environments (using different camera positions). In other words, we noticed that our domain adaptation model for gaze mapping, while robust for different domains, is affected by significant camera changes. This highlights the adaptability of our approach to diverse environments and even different camera positions for the same driver, indicating potential self-calibration capabilities. We also discovered that strategy B was more effective than strategy A in both of the given tasks. This indicates that strategy B is more successful in domain adaptive learning. In other words, we observed that the feature extraction step produces cleaner output as the environment's influence decreases.

3.3.4 Cave-DB and comparison with the existing studies

In this section, we compare all of our above-mentioned proposed strategies using the Cave-DB dataset compare them with other similar studies and present the results. On the other hand, we assessed the best-performing strategies for our proposed methods with other existing approaches. In this comparison, we evaluated 13 gaze regions that are commonly used in other existing studies that evaluated them on Cave-DB. Also, since we presented the experimental result of the domain adaptation method on CAVE-DB in the previous section, we will use the above results in this section.

As we mentioned in Section 3, we considered the SCER and LCER metrics. The SCER metric is the percentage of frames that are strictly correct, meaning the estimated gaze region matches the ground truth gaze region. The LCER metric is the percentage of frames that are loosely correct, meaning the estimated gaze region is within the ground truth gaze region and its neighboring regions. For gaze mapping using Openface with SVM classifier, during the experiment on the Cave-DB dataset, it was discovered that the combination of gaze angle, head position_R, and head rotation_R features (best of triple features) led to an accuracy rate of 80.4%, shown in Figure 3.12. However, when Eye position WO-Z features were added (best of quadruple features), the accuracy rate increased to the average Strictly Correct Estimation Rate (SCER) accuracy of 85.6%, in Figure 3.13.

Moreover, based on the analysis of the CAVE-DB dataset, it has been observed that the use of Openface with SVM classifier, along with the combination of gaze angle, head position_R, head rotation_R and Eye position WO-Z features (the best of quadruple features) results in better performance than the best of triple features for gaze mapping.

Finally, our strategies were compared with other existing studies on the Cave-DB dataset. The summary results are shown in Tables 3.10 and 3.11. As can be observed from this evaluation, our results are slightly better than those of the previous study. Our first strategy which is a combination of gaze angle, head position_R, and head rotation_R features showed a SCER rate of 80.4% and an LCER rate of 98.3%. Also, our second strategy which is a combination of Gaze angle, head position_R, head rotation_R, and eye position WO-Z features showed the best result of SCER rate of 85.6% and LCER rate of 98.7%. Also, our other strategies using the domain adaptation method, achieved an average Strictly Correct Estimation Rate (SCER) accuracy of 81.38% and 93.53%, and a Loosely Correct Estimation Rate (LCER) accuracy of 96.69% and 98.9% for the two strategies, respectively. Strategy A

Table 3.10: Comparison of our methods with existing studies on Cave-DB by SCER

G/R	Choi et.al	Naqvi et.al	Lee et.al	Our 1	Our 2	Our 3	Our 4
1	55%	87%	52%	77%	78%	54%	100%
2	52%	73%	64%	83%	85%	83%	98%
3	51%	80%	64%	74%	70%	100%	98%
4	53%	79%	21%	91%	91%	97%	100%
5	47%	67%	48%	58%	63%	0	46%
6	51%	72%	33%	60%	86%	61%	100%
7	52%	69%	3%	94%	95%	85%	100%
8	45%	67%	29%	72%	83%	98%	83%
9	53%	79%	35%	78%	81%	100%	100%
13	53%	86%	85%	99%	99%	100%	100%
14	53%	81%	4%	85%	85%	100%	100%
15	70%	88%	88%	97%	99%	100%	95%
16	50%	77%	46%	78%	99%	80%	96%
Avg:	***53.1%***	***77.7%***	***44.0%***	***80.4%***	***85.6%***	***81.3%***	***93.53%***

Note:
G/R = *Gaze regions*
Our 1 = *The strategy using gaze angle, head position R and head rotation_R features*
Our 2 = *The strategy using gaze angle, head position R, head rotation_R and Eye position WO-Z features*
Our 3 = *The strategy A using domain adaptation method*
Our 4 = *The strategy B using domain adaptation method*

using domain adaptation method demonstrated 0% recognition in gaze region 5 according to the SCER metrics, resulting in a decrease in the overall average accuracy. The recognition accuracy of other regions is reasonable. The confusion matrix indicates that gaze region 5 is often confused with its neighboring region 4, which is the closest gaze region, Figure 3.14 illustrates the point clearly. It is worth mentioning that Strategy B not only predicts high performance in all gaze regions but also improves the performance of Strategy A in region 5, which experiences high confusion. Additionally, Strategy B is more effective than Strategy A in terms of accuracy. On average, the SCER accuracy rate of Strategy B is 12.15% higher than that of Strategy A, which uses the driver's full appearance image. In other words, all of our strategies outperformed other existing methods, but among them, Strategy B of the domain adaptation method showed the best performance.

In addition to these studies, another state-of-the-art study is Vora et al. [63]. But they used 6 gaze regions: Forward, Right, Left, Center Stack, Rear view mirror, and speedometer. It is difficult to compare the

Table 3.11: Comparison of our methods with existing studies on Cave-DB by LCER

G/R	Choi et.al	Naqvi et.al	Lee et.al	Our 1	Our 2	Our 3	Our 4
1	83%	97%	98%	97%	99%	89%	100%
2	98%	97%	100%	99%	99%	99%	99%
3	76%	94%	92%	99%	99%	100%	98%
4	85%	97%	93%	99%	99%	97%	100%
5	100%	97%	93%	99%	99%	98%	99%
6	83%	95%	86%	99%	99%	92%	100%
7	83%	96%	72%	97%	98%	89%	100%
8	98%	98%	81%	98%	97%	97%	95%
9	83%	92%	76%	99%	99%	100%	100%
13	96%	97%	100%	99%	99%	100%	100%
14	72%	93%	54%	98%	98%	100%	100%
15	93%	99%	100%	98%	99%	100%	97%
16	96%	95%	61%	98%	99%	97%	99%
Avg:	***88.7%***	***96.3%***	***85.1%***	***98.3%***	***98.7%***	***96.7%***	***98.9%***

Note:
G/R = *Gaze regions*
Our 1 = *The strategy using gaze angle, head position R and head rotation_R features*
Our 2 = *The strategy using gaze angle, head position R, head rotation_R and Eye position WO-Z features*
Our 3 = *The strategy A using domain adaptation method*
Our 4 = *The strategy B using domain adaptation method*

results of studies with 13 gaze regions. Because there are fewer gaze regions and less chance of confusion. In their study, SqueezeNet, the best method on Face Embedded FoV, has 89.3% accuracy. However, our results of strategies of our proposed methods obtained matching results for more gaze regions, which can be considered as decent results comparable to state-of-the-art studies. Based on the experimental results, our proposed methods demonstrate comparable performance to the current state-of-the-art studies, and in some cases, the result of strategy using gaze angle, head position R, head rotation_R and Eye position WO-Z features even outperforms the result of existing studies. Our study also indicates that the problem of different domains or different driver performance degradation can be effectively addressed by utilizing domain adaptation methods, which have shown reasonable results. This highlights the adaptability of our approach to diverse environments.

3.4 Pedestrian detection evaluation

We implemented and tested both of the Lucas-Kanade method and YOLOv4 model for the same task of our research, shown in Figure 3.14. The evaluation results of these two methods on real video of road environment are shown in Table 3.12. It can be seen from the results that both methods performed reasonably well in the evaluation. However, a few things were noticed during the experiment.

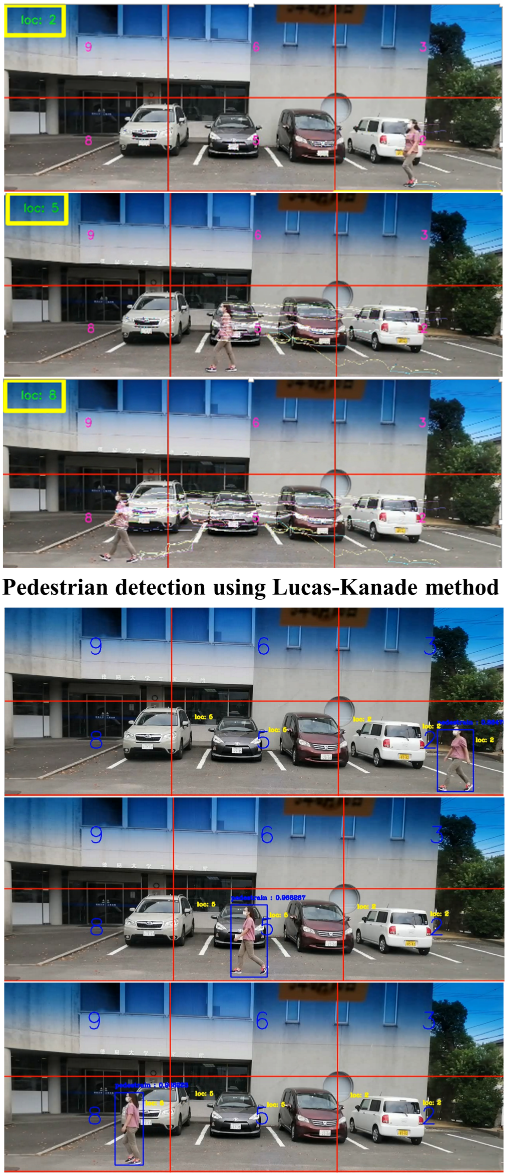
The Lucas-Kanade method may be suitable for real-time applications where speed is critical and you make certain assumptions about the scene's simplicity and the motion of objects. In evaluation, it was not the best choice for high-precision pedestrian detection in challenging environments. However, in the case of Gaze Region 8, when entering the region, it was wrongly recognized as Gaze Region 5 by the neighboring region, but after entering the center of the region, it was correctly recognized as Gaze Region 8. However, it has been observed that there is a problem with occlusions, varying lighting conditions, complex backgrounds, multi-object detection, and multi-recognition.

In other words, the Lucas-Kanade method detects moving objects such as pedestrians; there were cases where it was lost due to a noisy background during tracking, and it was not detected due to the effect of light. This means that although the pedestrian is recognized, it cannot be detected in some frames, and the tracking process is lost. On the other hand, YOLOv4, in addition to high-accuracy detection, required lower resources in terms of performance compared to similar deep learning-based methods, which was the reason for our choice.

Table 3.12: Evaluation results of the Lukas-Kanade dense and YOLOv4 for pedestrian detection task

G/R	***Procedures***			
	Method 1		***Method 2***	
	Detect	***Pred.State***	***Detect***	***Pred.State***
2	yes	2	yes	2
5	yes	5	yes	5
8	yes	5/8	yes	8

G/R = *Gaze regions*
Method 1 = *Pedestrian detection using Lucas-Kanade method*
Method 2 = *Pedestrian detection using YOLOv4 model*



Pedestrian detection using Lucas-Kanade method

Pedestrian detection using YOLOv4

Figure 3.14: Scene of comparative evaluation of the Lucas-Kanade and YOLOv4

3.5 Evaluation of driver's distraction detection module

In this section, we evaluated the combination of the gaze mapping method and the pedestrian detection method. We evaluated the state of the distraction of the driver based on the combination of two procedures, such as gaze mapping and pedestrian detection, on the evaluation video. In the video, the pedestrian starts his movement from gaze region 8, then passes through gaze region 5 and 2 to reach gaze region 1, shown in Figure 3.15. At this moment, the driver is gazing at the pedestrian.

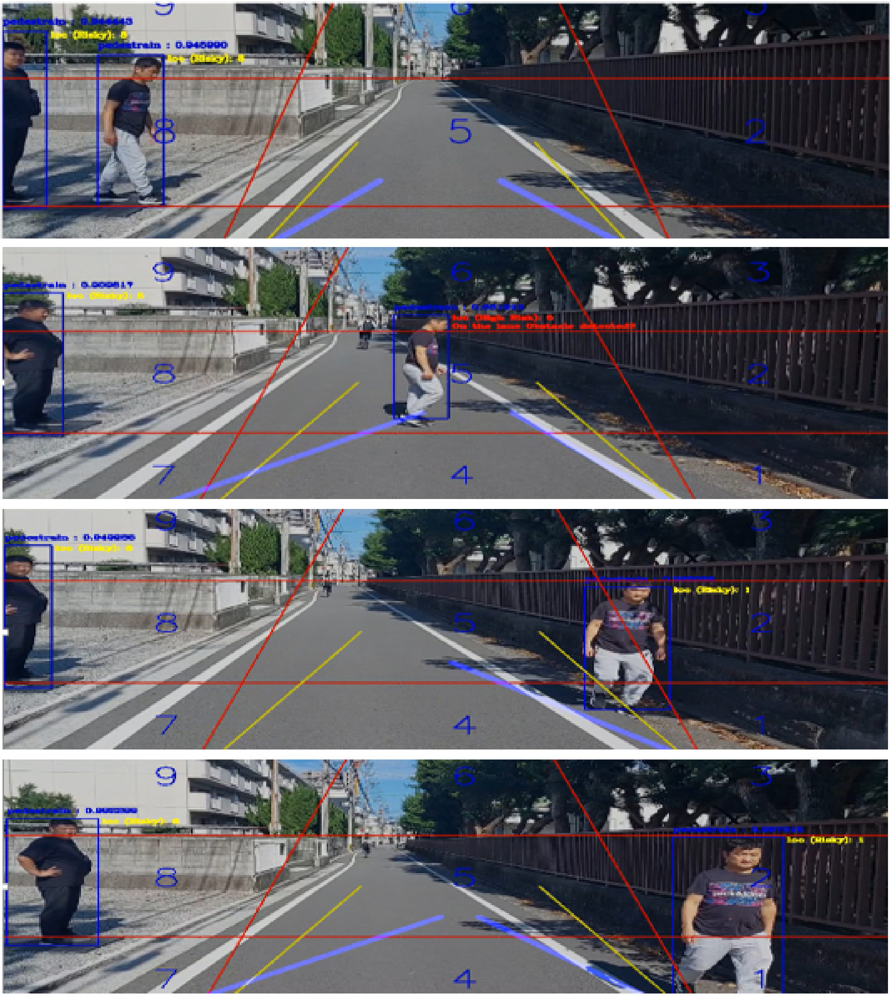


Figure 3.15: Evaluation scene of the combination of gaze mapping using Openface with SVM and YOLOv4

In Table 3.13, the gaze mapping method using domain adaptation determined the direction of a driver's gaze, gaze regions 8 and 5 were correct, but Gaze Region 2 and 1 were incorrectly predicted as Gaze Region 5 and 2. However, in the case of Gaze Region 2, when entering

Table 3.13: Comparative results of the combination of gaze mapping and pedestrian detection

G/R	**Procedures**					
	Ped.Det		***GM 1***		GM 2	
	Detect	***Pred.d***	***Detect***	***Pred.d***	***Detect***	***Pred.d***
8	yes	8	Yes	8	yes	8
5	yes	5	Yes	5	yes	5
2	yes	2	Yes	2	Yes/No	5/2
1	yes	2	No	2	No	2

G/R = *Gaze regions*
Ped.Det = *Pedestrian detection using YOLOv4 model*
GM 1 = *Gaze mapping using Openface with SVM using the combination of Gaze angle, head position_R, head rotation_R, and eye position WO-Z features*
GM 2 = *Gaze mapping using Domain adaptation (Strategy B)*

the region, it was wrongly recognized as Gaze Region 5 by neighboring region, but after entering the center, it was correctly recognized as Gaze Region 2. On the other hand, when the gaze mapping method using the OpenFace with SVM classifier determined the direction of a driver's gaze, gaze regions 8, 5 and 2 were correctly predicted, but gaze region 1 was incorrectly predicted as gaze region 2 by neighboring region. We observed that the incorrectly predicted gaze regions were a neighboring of the target gaze region in both methods.

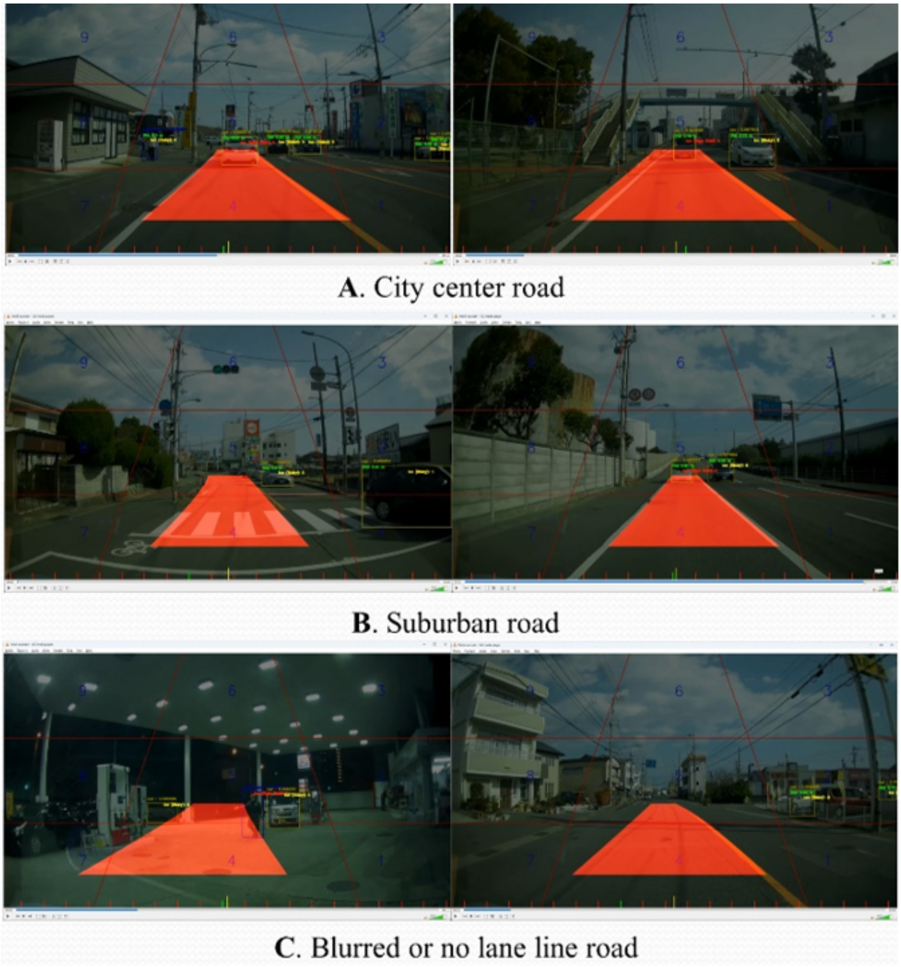
3.6 Evaluation of pedestrian safety module

In this section, we will present the evaluation of the pedestrian safety module based on its two main procedures: pedestrian distance and position detection, and lane line detection. We tested the pedestrian safety module using a combination of two procedures. For the evaluation, we used three different road videos in different environments. The first video was shot in a city center with clear lane lines but many other road users, including pedestrians and vehicles. The second video had clear lane lines and fewer pedestrians who were further away from the vehicle. The final video had either blurred or no lane lines and fewer pedestrians, as described in Figure 3.16. The evaluation results are displayed in Table 3.14.

Based on the test results, both procedures appeared to perform

Table 3.14: Comparative results of the combination of gaze mapping and pedestrian detection

Videos	*Accurate Recognition Rate (%) of the lane line*	*Accurate Recognition Rate (%) of pedestrian distance & position*
City center	96.07%	94.24%
Suburban	98.45%	80.97%
Rural	92.85%	84.15%
Average:	**95.79%**	**86.45%**



A. City center road

B. Suburban road

C. Blurred or no lane line road

Figure 3.16: Evaluation scene of the pedestrian safety module

reasonably. However, our video analysis revealed that the pedestrian distance and position detection procedure faced issues with recognizing small objects in noisy backgrounds. If it is not recognized as a pedestrian, the whole procedure will not work properly. Because it creates a bounding box by identifying it as a pedestrian and then estimates the distance and location of the pedestrian. Specifically, when analyzing video from the suburban area, pedestrians at a distance were not detected until they were closer. We also observed that the model accurately recognized pedestrians up to approximately 30 meters away from the camera, but its performance declined for pedestrians further away. Fortunately, the road range we are interested is within 25 meters. Based on our pedestrian distance and position detection procedure, we can confidently say that it meets our requirements as it performs exceptionally well up to a distance of 30 meters.

Chapter 4

Conclusion and Future work

4.1 Conclusion

In this thesis, we introduced an innovative and cost-efficient Safe Driving Support System designed to address the critical issues of driver distraction and pedestrian safety. Our system leverages the capabilities of dual dashboard cameras to create a non-intrusive and lightweight solution that can enhance road safety.

Within this proposal, we investigated the effects of combining the features one by one and with the other features in binary, triple, and quadruple to determine how they affect the estimation of gaze mapping using different camera positions. According to experiment results, Camera Position 2 is superior to Camera Position 1 for gaze estimation. The driver distraction detection module in our system employs advanced gaze mapping techniques and facial feature extraction. We compared three strategies for gaze mapping, and the results showed that the strategy using OpenFace with SVM classifier (using gaze angle, head position_R features, head rotation_R features, and eye position WO-Z features) outperformed all other methods, achieving an impressive 85.6% accuracy for Strictly Correct Estimation Rate (SCER) and a 98.7% accuracy for Loosely Correct Estimation Rate (LCER). These results indicate the effectiveness of our approach in accurately identifying driver distraction.

Our other approach gaze mapping used the domain adaptation method. The research also addressed common challenges encountered in existing gaze mapping systems, such as performance deterioration across different drivers and environments. To overcome these challenges, domain adaptation techniques were employed to mitigate the negative effects of domain shifting. The thesis discussed several related studies that utilized adversarial learning, elimination of gaze-irrelevant features, and unsupervised domain adaptation techniques. The proposed method consisted of three steps: pre-processing, facial feature extraction, and gaze region classification. Two pre-processing strategies were explored: using the driver's full appearance image and focusing on the driver's face image. Experimental results demonstrated that gaze mapping using the domain adaptation method achieved an average Strictly Correct Estimation Rate (SCER) accuracy of 81.38% and 93.53%, and a Loosely Correct Estimation Rate (LCER) accuracy of 96.69% and 98.9% for the two strategies, respectively. These results underscore the effectiveness of our method in adapting to different domains. Furthermore, we attain an average SCER accuracy of 85.00% and 94.84%, and LCER accuracy of 98.80% and 99.23% for the two strategies, respectively. This highlights the adaptability of our approach to diverse environments and even different camera positions for the same driver, indicating potential self-calibration capabilities. The method achieved remarkable accuracy rates in gaze region classification, reducing the gaze mapping error and showcasing better performance across different drivers.

The thesis also introduced the Driver Gaze Mapping (DGM) dataset, which was prepared specifically for the gaze mapping task. The dataset included images from different camera positions and diverse driving environments. Additionally, the open dataset Columbia Cave-DB was utilized to evaluate the proposed method's accuracy through unsupervised classification and comparison with existing studies. The experimental results demonstrated that the proposed method surpassed previous approaches in terms of accuracy and performance.

Furthermore, our system integrates a pedestrian safety module that analyzes road conditions, lane lines, and pedestrian positions. This module demonstrated promising results, with an average lane line recognition accuracy of 95.79% and a pedestrian distance and position detection accuracy of 86.45%. While there were some challenges in recognizing pedestrians at greater distances, our system performed exceptionally well within the critical 25-meter range. By combining these two modules, our SDSS aims to proactively detect and mitigate the root causes of many accidents - driver distraction and pedestrian risk. The integration of real-time video analysis and state-of-the-art technologies enhances road safety by providing timely warnings and assistance to drivers.

In a nutshell, our research offers a comprehensive solution to address the pressing issues of driver distraction and pedestrian safety. The results from our experiments highlight the system's effectiveness in detecting and mitigating potential hazards on the road. We believe that the deployment of such systems can play a crucial role in reducing accidents and saving lives on our roads, ultimately contributing to a safer and more secure transportation environment.

4.2 Future work

Gaze mapping using face generalization method:

Researchers are currently conducting innovative studies in the field of gaze mapping. One such study uses Face Generalization for gaze estimation, which is very inspiring. The study removes personal facial features from the training data and only uses gaze-related features. This approach ensures that factors such as the personal appearance of participants used in the training dataset, the number of participants, and their gender do not affect the information.

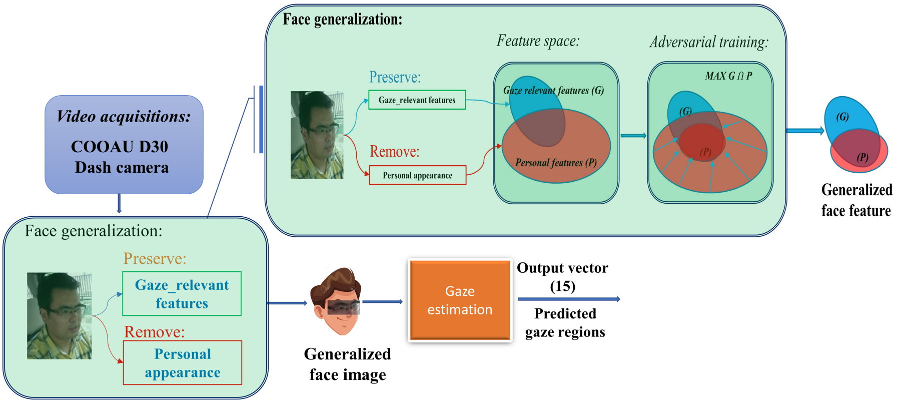


Figure 4.1: Overview of gaze mapping using face generalization method

The primary objective of this study is to determine the direction of eye gaze using generalized facial features. The use of face generalization in gaze estimation tasks ensures that only gaze-relevant features are preserved, while gaze-irrelevant features are eliminated. Figure 4.1 provides a detailed description of the face generalization task. The Adversarial Neural Network method is used to preserve the gaze-relevant feature and eliminate gaze-irrelevant features. This method aims to reduce gaze-irrelevant features and maximize gaze-relevant features.

In other words, the extracted feature needs to contain more gaze information and less general image information. Adversarial learning is used during the process to eliminate gaze-irrelevant features and preserve the gaze-relevant feature.

In this way, with zero-shot gaze mapping using gaze generalization which is the task of driver face feature extraction where no visual training data is available for some of the target drivers, it will be possible to determine the gaze direction without any configuration and additional training. This will be a significant contribution to this field of research, which lacks real-world datasets during driving.

Post-processing:

In our future work, we also plan to enhance the results of the gaze mapping step by focusing on the Features of Neighboring Gaze Regions (FNGR). Our observations revealed that certain gaze regions with high probabilities were not necessarily neighbors, leading us to the need for optimization. By considering the features of neighboring gaze regions, we aim to improve the accuracy of the classifier's predictions. This optimization process involves identifying neighboring gaze regions in feature space clusters, closely resembling the actual driver's gaze regions.

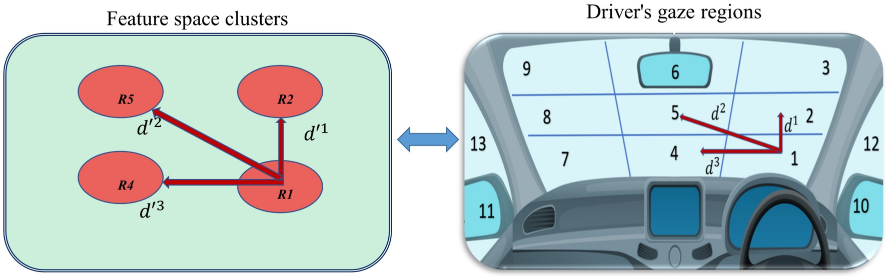


Figure 4.2: Overview of Features of Neighboring Gaze Regions

FNGR has been observed that the feature extractor categorizes images from distinct gaze regions into separate different feature space clusters. The gaze regions depicted in these feature space clusters and the distance between them are akin to the actual driver's gaze regions, we call these the features of neighboring gaze regions. To put it simply, the gaze region where the driver looks in real life and the feature space clusters have identical neighbors, and the relative distances between these neighbors are maintained in the same way as demonstrated in Figure 4.2. Therefore, we need to implement the enhancer module to improve the results based on the features of neighboring gaze regions. The Features of Neighboring Gaze Regions (FNGR) is a simple yet crucial feature in gaze mapping, as it considers the neighboring regions of real gaze regions, even within feature space clusters. In other words, it is possible to determine the region of any gaze region by the position of its neighboring regions. This means that even if it does not have a label, it can be assigned a label by its immediate neighbors. Assuming that *G(i)* is the *i*-th gaze region, *N(G(i))* represents its neighboring regions.

To describe *G(i)* based on its neighboring regions, we denote as a *F(G(i)) = f(N(G(i)))*. For our task, we will use this feature as an optimizer. The enhancer module of FNGR retains the previously predicted gaze region, denoted as *G(i-1)*. As mentioned above, in this optimazation process, the gaze regions predicted by the gaze mapping step are taken and select the first five regions with the highest probability. Consequently, it is determined whether each selected gaze region is a neighboring to the gaze region *G(i-1)*. If the gaze region is not contiguous, it is removed from the recommendation list. To determine the final gaze region, the enhancer module analyzes the remaining gaze regions on the recommendation list and selects the one with the highest probability.

In addition to this specific improvement, we have outlined several other activities for our future work:

- Enhancing Pedestrian Recognition: We aim to improve the SDSS's ability to recognize pedestrians at greater distances, specifically within a range of up to 30 meters. This is crucial for enhancing the safety and situational awareness of the system.

- Predictive Algorithms: We plan to develop predictive algorithms that go beyond driver distraction detection and also anticipate potentially distracting situations by analyzing driver behavior and context. This proactive approach will enable the SDSS to issue warnings and provide assistance before distractions become critical.

- To ensure optimal performance, it is crucial to conduct extensive real-world testing and validation of the SDSS under diverse driving conditions and environments. This comprehensive evaluation would enable a thorough assessment of its performance in various scenarios, and facilitate refinement of its algorithms accordingly.

- It is also essential to investigate ways to reduce the cost of implementing the SDSS, making it more accessible for a broader range of vehicles.

Bibliography

[1] L. Sminkey. "Road traffic injuries." (accessed Jun. 9, 2023). (2010), [Online]. Available: `https://www.who.int/news/item/11-12-2010-pedestrians-cyclists-among-main-road-traffic-crash-victims` (cit. on p. 1).

[2] M. B. E. Tomás, Q. P. Mario, L. B. Sergio and M. P. Gregorio, "Safecar: A brain–computer interface and intelligent framework to detect drivers' distractions," *Expert Systems with Applications*, vol. 203, p. 117 402, 2022 (cit. on p. 2).

[3] P. Yong, X. Qian, L. Shuxiang *et al.*, "The application of electroencephalogram in driving safety: Current status and future prospects," *Decision Neuroscience, a section of the journal Frontiers in Psychology*, vol. 13, pp. 1–16, 2022 (cit. on p. 2).

[4] B. Cheng, C. Fan, H. Fu, J. Huang, H. Chen and X. Luo, "Measuring and computing cognitive statuses of construction workers based on electroencephalogram: A critical review," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 6, pp. 1644–1659, 2022 (cit. on p. 2).

[5] C. Fan, J. Hu, S. Huang, Y. Peng and S. Kwong, "Eeg-tnet: An end-to-end brain computer interface framework for mental workload estimation," *Decision Neuroscience, a section of the journal Frontiers in Psychology*, vol. 16, pp. 1–11, 2022 (cit. on p. 2).

[6] U. Fugiglando, E. Massaro, P. Santi *et al.*, "Driving behavior analysis through can bus data in an uncontrolled environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, pp. 737–748, 2017 (cit. on p. 2).

[7] M. N. Azadani and A. F. M. Boukerche, "Performance evaluation of driving behavior identification models through can-bus data," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6 (cit. on p. 2).

[8] M. H. Alkinani, W. Z. Khan and Q. Arshad, “Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements, and open challenges,” *IEEE Access*, vol. 8, pp. 105 008–105 030, 2020 (cit. on p. 2).

[9] I. Dua, A. U. Nambi, C. V. Jawahar and V. N. Padmanabhan, “Evaluation and visualization of driver inattention rating from facial features,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 2, pp. 98–108, 2019 (cit. on p. 2).

[10] F. Vicente, Z. Huang, X. Xiong, F. D. la Torre, W. Zhang and D. Levi, “Driver gaze tracking and eyes off the road detection system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, 2015 (cit. on p. 2).

[11] N. Mizuno, A. Yoshizawa, A. Hayashi and T. Ishikawa, “Detecting driver’s visual attention area by using the vehicle-mounted device,” in *Proceedings of the IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing*, 2017, pp. 346–352 (cit. on pp. 3, 28).

[12] Y. Wang, X. Ding, G. Yuan and X. Fu, “Dual-cameras-based driver’s eye gaze tracking system with non&ndash;linear gaze point refinement,” *Sensors*, vol. 22, no. 6, p. 2326, 2022 (cit. on p. 3).

[13] D. Xiao and C. Feng, “Detection of drivers’ visual attention using smartphones,” in *Proceedings of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, 2016, pp. 630–635 (cit. on p. 3).

[14] P. Smith, M. Shah and N. da Vitoria Lobo, “Determining driver visual attention with one camera,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 4, pp. 205–218, 2003 (cit. on p. 3).

[15] K. Ohue, Y. Yamada, S. Uozumi, S. Tokoro, A. Hattori and T. Hayashi, “Development of a new pre-crash safety system,” in *Proceedings of the Soc. Automotive Eng. World Congr*, 2006, p. 1461 (cit. on p. 3).

[16] J. P. Batista, “A real-time driver visual attention monitoring system,” in *Proceedings of the Iberian Conf. Pattern Recog. Image Anal.*, vol. 3522, 2005, pp. 200–208 (cit. on p. 3).

[17] J. Nuevo, L. M. Bergasa, M. A. Sotelo and M. Ocana, “Real-time robust face tracking for driver monitoring,” in *Proceedings of the IEEE Conf. Intell. Transp. Syst.*, 2006, pp. 1346–1351 (cit. on pp. 3, 28).

[18] J. Y. Kaminski, D. Knaan and A. Shavit, “Single image face orientation and gaze detection,” *Machine Vision and Applications*, vol. 21, no. 1, pp. 85–98, 2009 (cit. on p. 3).

[19] R. Naqvi, M. Arsalan, G. Batchuluun, H. Yoon and K. Park, “Deep learning-based gaze detection system for automobile drivers using a nir camera sensor,” *Sensors*, vol. 18, no. 2, p. 456, 2018 (cit. on p. 3).

[20] K. Wang, R. Zhao, H. Su and Q. Ji, “Generalizing eye tracking with bayesian adversarial learning,” in *Proceedings of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, 2019, pp. 11 907–11 916 (cit. on p. 4).

[21] Y. Cheng, Y. Bao and F. Lu, “Puregaze: Purifying gaze feature for generalizable gaze estimation,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 436–443 (cit. on p. 4).

[22] Y. Bao, Y. Liu, H. Wang and F. Lu, “Generalizing gaze estimation with rotation consistency,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4197–4206 (cit. on p. 4).

[23] C. Papageorgiou and T. Poggio, “A trainable pedestrian detection system,” *International Journal of Computer Vision (IJCV)*, pp. 15–33, 2000 (cit. on p. 5).

[24] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893 (cit. on p. 5).

[25] A. Solichin and A. Harjoko, “A survey of pedestrian detection in video,” *International Journal of Advanced Science and Application (IJACSAI)*, pp. 41–47, 2014 (cit. on p. 5).

[26] B. Wu and R. Nevatia, “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005, pp. 90–97 (cit. on p. 5).

[27] K. Mikolajczyk, C. Schmid and A. Zisserman, “Human detection based on a probabilistic assembly of robust part detectors,” in *The European Conference on Computer Vision (ECCV)*, vol. 3021/2004, 2005, pp. 69–82 (cit. on p. 5).

[28] S. Piérard, A. Lejeune and M. V. Droogenbroeck, “A probabilistic pixel-based approach to detect humans in video streams,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 921–924 (cit. on p. 5).

[29] R. Girshick, J. Donahue and T. D. et al., “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587 (cit. on p. 6).

[30] K. He, G. Gkioxari and P. D. et al., “Mask r-cnn,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988 (cit. on p. 6).

[31] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448 (cit. on p. 6).

[32] S. Ren, K. He and R. G. et al., “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 39, no. 6, pp. 91–99, 2015 (cit. on p. 6).

[33] J. Redmon, S. Divvala and R. G. et al., “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788 (cit. on p. 6).

[34] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017 (cit. on p. 6).

[35] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, `http://arxiv.org/abs/1804.02767`, 2018 (cit. on p. 6).

[36] W. Liu, D. Anguelov and D. E. et al., “Ssd: Single shot multibox detector,” in *Proceedings of the European Conference on Computer Vision*, Springer, 2016, pp. 21–37 (cit. on p. 6).

[37] W. Liu, S. Liao and W. H. et al., “Learning efficient single-stage pedestrian detectors by asymptotic localization fitting,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 643–659 (cit. on p. 6).

[38] Z. Zheng, P. Wang and W. Liu, *Distance-iou loss: Faster and better learning for bounding box regression*, `https://arxiv.org/abs/1911.08287`, 2019 (cit. on p. 6).

[39] B. A. Wang and C. W. Liao, *Optimal speed and accuracy of object detection*, `https://arxiv.org/abs/2004.10934`, 2020 (cit. on p. 6).

[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520 (cit. on p. 13).

[41] T. Baltrusaitis, A. Zadeh, Y. C. Lim and L. P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2018, pp. 59–66 (cit. on pp. 13, 14).

[42] I. Nitze, U. Schulthess and H. Asche, "Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to the maximum likelihood for supervised crop type classification," in *Proceedings of the 4th GEOBIA*, 2018, pp. 035–045 (cit. on pp. 14, 15).

[43] M. Monaro, S. Maldera, C. Scarpazza and G. Sartori, "Detecting deception through facial expressions in a dataset of videotaped interviews: A comparison between human judges and machine learning models," *Computers in Human Behavior*, vol. 127, no. 107063, pp. 1–10, 2021 (cit. on p. 15).

[44] R. Rill-Garcia, H. Escalante, L. Villasenor-Pineda and V. Reyes-Meza, "High-level features for multimodal deception detection in video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 0–0 (cit. on p. 15).

[45] F. Pedregosa, G. Varoquaux, A. Gramfort and V. Michel, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011 (cit. on p. 15).

[46] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014 (cit. on p. 19).

[47] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2016, 770–778 (cit. on p. 19).

[48] D. Gerónimo, A. Sappa, D. Ponsa and A. López, "2d–3d-based onboard pedestrian detection system," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 583–595, 2010 (cit. on p. 21).

[49] J. Cao, C. Song, S. Song, F. Xiao and S. Peng, "Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments," *Sensors*, vol. 19, no. 14, p. 3166, 2019 (cit. on p. 22).

[50] S. J. Lee, J. Jo, H. G. Jung, K. R. Park and J. Kim, "Real-time gaze estimator based on driver's head orientation for forward collision warning system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 254–267, 2011 (cit. on pp. 26, 28, 30).

[51] H. Yoon, N. Baek, N. Truong and K. Park, "Driver gaze detection based on deep residual networks using the combined single image of dual near-infrared cameras," *IEEE Access*, vol. PP, pp. 1–1, Jul. 2019. DOI: `10.1109/ACCESS.2019.2928339` (cit. on p. 26).

[52] R. Naqvi, M. Arsalan, G. Batchuluun, H. Yoon and K. Park, "Deep learning-based gaze detection system for automobile drivers using a nir camera sensor," *Sensors*, vol. 18, no. 2, p. 456, 2018 (cit. on pp. 26, 28, 30).

[53] Y. Wang, G. Yuan, Z. Mi *et al.*, "Continuous driver's gaze zone estimation using rgb-d camera," *Sensors*, vol. 19, no. 6, 2019, ISSN: 1424-8220. DOI: `10.3390/s19061287`. [Online]. Available: `https://www.mdpi.com/1424-8220/19/6/1287` (cit. on p. 26).

[54] B. A. Smith, Q. Yin, S. K. Feiner and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proc. 26th ACM Symp. User Interface Softw. Technol*, 2013, 271–280 (cit. on p. 25).

[55] I. H. Choi, S. K. Hong and Y. G. Kim, "Real-time categorization of driver's gaze zone using the deep learning techniques," in *Proceedings of the International Conference on Big Data and Smart Computing*, 2016, 143–148 (cit. on pp. 28, 30).

[56] K. Tsubowa, T. Akiduki, Z. Zhang, H. Takahashi and Y. Omae, "A study of effects of driver's sleepiness on driver's subsidiary behaviors," *International Journal of Innovative Computing, Information and Control*, vol. 17, no. 5, pp. 1791–1799, 2021 (cit. on p. 28).

[57] J. Araluce, L. M. Bergasa, M. Ocana, E. Lopez-Guillen, P. A. Revenga and O. Perez, “Gaze focalization system for driving applications using openface 2.0 toolkit with narmax algorithm in accidental scenarios,” *Sensors*, vol. 21, no. 18, pp. 1–19, 2021. DOI: `10.3390/s21186262` (cit. on p. 28).

[58] L. Fridman, P. Langhans, J. Lee and B. Reimer, “Driver gaze region estimation without use of eye movement,” *IEEE Intell. Syst*, vol. 31, no. 3, pp. 49–56, 2016 (cit. on p. 28).

[59] L. Fridman, J. Lee, B. Reimer and T. Victor, “Owl and lizard: Pattern of head pose and eye pose in driver gaze classification,” *IET Computer Vision*, vol. 10, no. 4, pp. 308–314, 2016 (cit. on p. 28).

[60] S. Ulziibayar, S. Karungaru, K. Terada and A. Altangerel, “Detection of driver’s visual distraction using dual cameras,” *Int. J. Innov. Comput. Inform. Control*, vol. 18, no. 05, pp. 1445–1461, 2022 (cit. on p. 28).

[61] S. Ulziibayar, S. Karungaru, K. Terada and A. Altangerel, “Appearance-based driver’s gaze mapping using a dash camera,” vol. 12, no. 23, pp. 1–5, 2022 (cit. on p. 28).

[62] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” pp. 248–255, 2009 (cit. on p. 31).

[63] S. Vora, A. Rangesh and M. Trivedi, “Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis,” *IEEE Trans. Intell. Veh*, vol. 3, no. 03, pp. 254–265, 2018. DOI: `10.1109/TIV.2018.2843120` (cit. on p. 45).