GABRIEL NASCARELLA HISHIDA DO NASCIMENTO

LOOK WHERE YOU'RE GOING: CLASSIFYING DRIVERS' ATTENTION THROUGH 3D GAZE ESTIMATION

(*pre-defense version, compiled at March 2, 2023*)

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Todt.

CURITIBA PR

2023

ABSTRACT

Recognizing where humans have their attention is a relevant and challenging task in areas such as intelligent tutoring systems, human-robot interaction, or driver monitoring systems. However, thanks to modern research, publicly available AI can predict gaze direction with less than 10° of average angular error in unconstrained environments. This work proposes using a 3D gaze estimation model to categorize drivers' attention in a visual attention classification pipeline. Six simple machine learning algorithms are trained to classify whether the driver's gaze direction means they are looking at relevant road elements or not, achieving ∼92% accuracy. A simple driver monitoring system is developed by implementing the pipeline. The visual attention classification models and the driver monitoring system are available at `https://github.com/VRI-UFPR/LWYG-drivers-attention`.

Keywords: Gaze estimation. Driver monitoring. Computer vision. Machine learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DMD	Driver Monitoring Dataset
DT	Decision Tree
GNB	Gaussian Naive Bayes
KNN	K-Nearest Neighbours
LR	Looking Road
ML	Machine Learning
NLR	Not Looking Road
RBF	Radial Basis Function
RF	Random Forest
RGB	Red, Green and Blue
SVC	Support Vector Classifier
SVM	Support Vector Machine
UFPR	Universidade Federal do Paraná
VRI	Vision, Robotics and Imaging

CONTENTS

1 INTRODUCTION

As a result of the newest advancements, gaze estimation (estimating where humans are looking) is becoming both more accessible and precise, for computer vision research has come to a point where AI models – such as Gaze360 [1] and L2CS-Net [2] – can process images and acquire data such as head pose and/or eyeball orientation to estimate gaze directions in any environment, as shown in Figure 1.1.

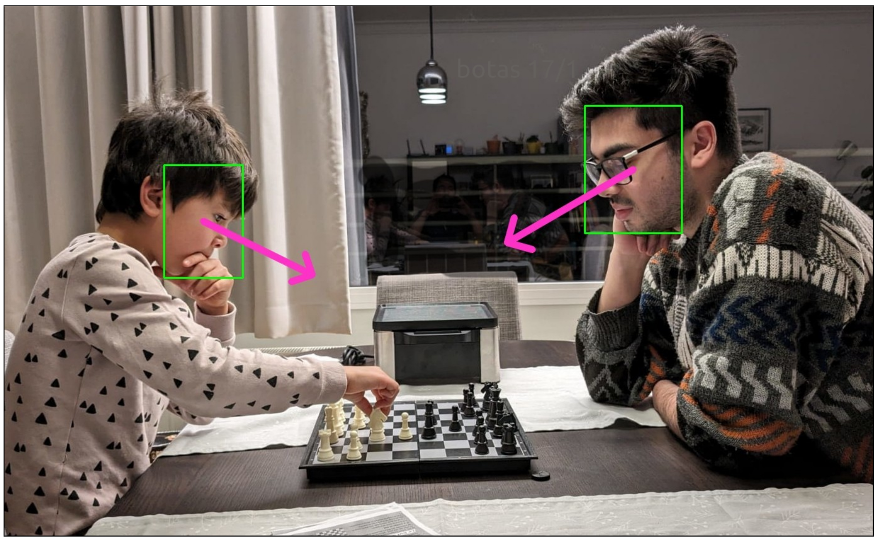

Figure 1.1: A 3D gaze estimation model, L2CS-Net [2], predicts where chess players look.

Eye-tracking data is proven to be useful in many ways. It can be used to train AI models to learn how to do tasks such as object referencing, decision-making, or recognizing the context of an image [3].

Gaze data can also be used by machines in order to understand human attention [3]. For instance, Massato et al. [4] used existing eye-tracking technology in a behavioural study to see where art specialists set their eyes when looking at paintings (see Figure 1.2).

Hutt et al. [5] also proposed that intelligent tutoring systems should be able to recognize when students are distracted. They trained a mind-wandering detector by using contextual features such as gaze direction, the teaching subject, the student's score, and time spent on each lesson.

The idea of understanding human attention through gaze can also be applied when monitoring drivers' attention. In recent studies, Yang et al. [6] proposed that a multi-task CNN can be trained to detect head pose and gaze direction to calculate attention parameters such as distraction and drowsiness.

These two studies on detecting attention through gaze [5, 6] proposed to train models that, before being able to classify attention, had to learn the task of gaze

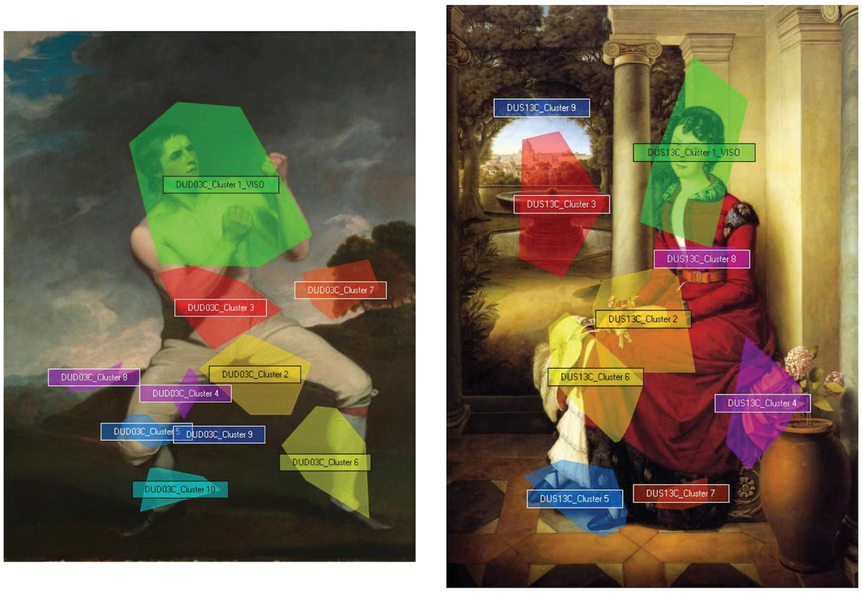

Figure 1.2: Eye-tracking software clusters gaze data points on artworks. Source: Massaro et al. [4]

estimation on their own. However, thanks to modern research, there are now 3D gaze estimation models publicly available that predict gaze direction with less than 10° of angular error [2], that could be used as tools to understand human attention. In the same study in which it was introduced, Gaze360 [1] was tested to predict which product a potential customer was looking at on a supermarket-style shelf, as shown in Figure 1.3.

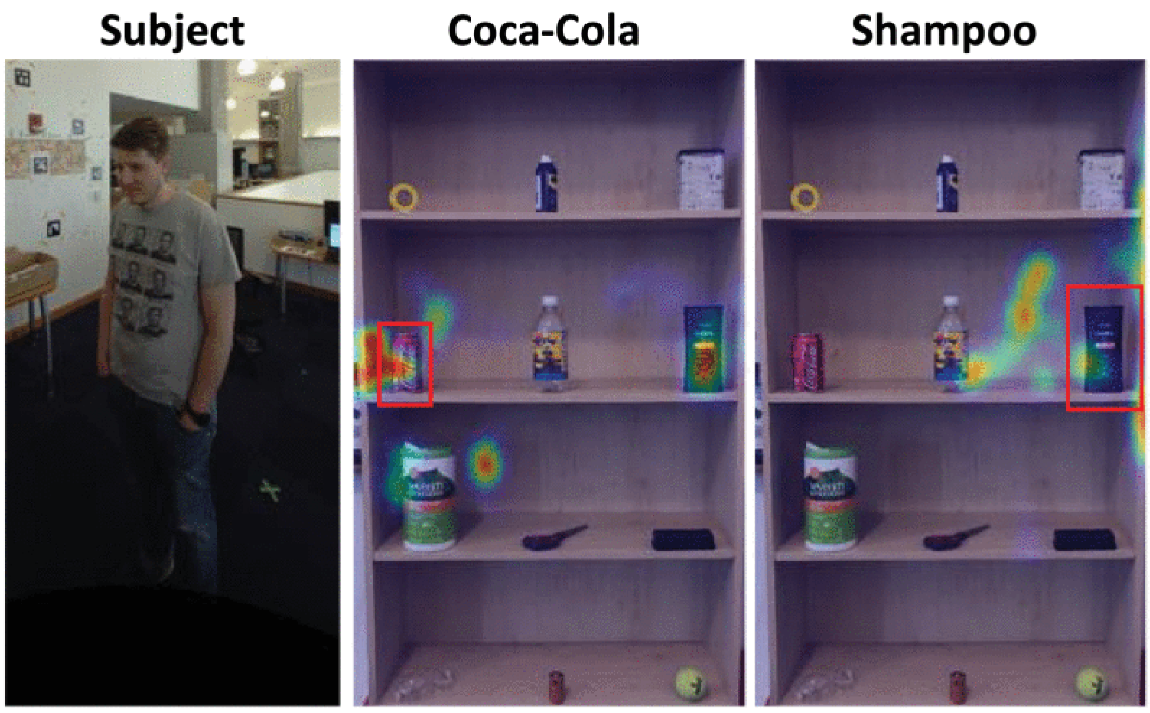

Figure 1.3: Gaze360 predicts the product a participant is looking at. Source: Kellnhofer et al. [1]

In a more recent study, Shi et al. [7] proposed the use of a 3D gaze estimation model in driver monitoring systems, but their study was more focused towards identifying the driver's gaze region rather than classifying the driver's attention. Their study was also limited by the use of a gaze estimation model that has been outperformed by novel methods.

1.1 OBJECTIVE

This work aims to test how a novel 3D gaze estimation model publicly available such as L2CS-Net [2] performs as a tool to categorize human attention. As a case study, this work takes on the task of classifying drivers' attention, for it is a simple context yet an important one – visual distractions such as telephone usage and texting increase the likelihood of being involved in a car crash by 4-23 times [8].

A visual attention classification pipeline is proposed to obtain the driver's gaze direction from L2CS-Net and use it as input to train a visual attention classifier. In total, six machine learning classification models were trained and evaluated. In order to train them, the Driver Monitoring Dataset (DMD) was used, an annotated dataset made by videos of 35 different drivers in 3 different camera angles [9].

1.2 STUDY OUTLINE

This study is structured into six chapters.

Chapter 2 provides an overview of the current literature on gaze estimation. It covers gaze estimation techniques, their categories and the state-of-the-art; the machine learning classification algorithms used in this study are also introduced. Chapter 3 presents an overview of driver monitoring systems. It defines driver monitoring and reviews the different approaches used in the literature to detect driver distraction, fatigue, and drowsiness.

Chapter 4 introduces the proposed visual attention classification pipeline. It describes the methodology and the dataset used to train and evaluate the machine learning classification models. Chapter 5 presents and discusses the results achieved by using the gaze estimation model on the DMD dataset, compares the performance of the six gaze direction classification models and discusses the implications of the results. Finally, Chapter 6 summarises the study's achievements, explores potential applications for the visual attention classifiers, and suggests opportunities for future research.

2 BACKGROUND

This chapter reviews the current literature on gaze estimation; techniques for acquiring gaze direction are presented and categorized, and the state-of-the-art gaze estimation model, L2CS-Net, is introduced. This chapter also introduces the machine learning classification algorithms used in this study.

2.1 GAZE ESTIMATION

Gaze estimation is the task of determining where a person is looking at a given moment. According to Hansen et al. [10], a person's gaze is determined by the head pose (position and orientation) and eyeball orientation. Head pose determines general gaze direction, while eyeball orientation determines local and detailed gaze direction.

Gaze estimation models, therefore, need to (either directly or implicitly) model both head pose and pupil/iris position. In this sense, gaze estimators can generally be distinguished as feature-based or appearance-based [10].

2.1.1 Feature-based Gaze Estimation

Feature-based gaze estimation methods use extracted local features of the face, such as contours, eye corners, and reflections from the eye. Most of these methods are based on a 3D model of the eye – as illustrated in Figure 2.1 – and directly compute the gaze direction based on a geometric eye model.

According to Zhang et al. [11], model-based gaze estimation techniques have lower accuracy because they depend on accurately detecting eye features. To do this, they require high-resolution images and homogeneous illumination, preventing these methods from being used in real-world scenarios. Furthermore, model-based techniques may not generalize well across different subjects due to variations in eye morphology.

2.1.2 Appearance-based Gaze Estimation

Appearance-based gaze estimation methods calculate 3D gaze direction directly from images. These methods use machine learning algorithms to learn a mapping between the input image and the corresponding gaze direction.

Doing this in unconstrained environments is a novelty. Previously, images used to train appearance-based gaze estimation models were collected under controlled laboratory conditions [11]. The challenge was not to make any assumptions regarding the user, environment, or camera.

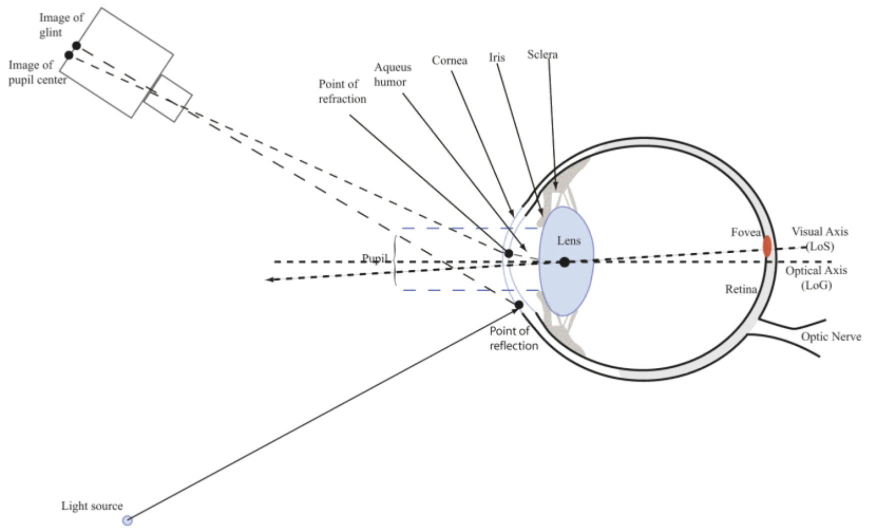

Figure 2.1: A general model of the structures of the human eye, light, light sources, and projections. Source: Hansen et al. [10]

It was not until 2015 that Zhang et al.[11] published the MPIIGaze dataset, with more than 210,000 images taken in a span of three months with variable lightning and background conditions by 15 participants on their laptops. Figure 2.2 shows a sample of these images.

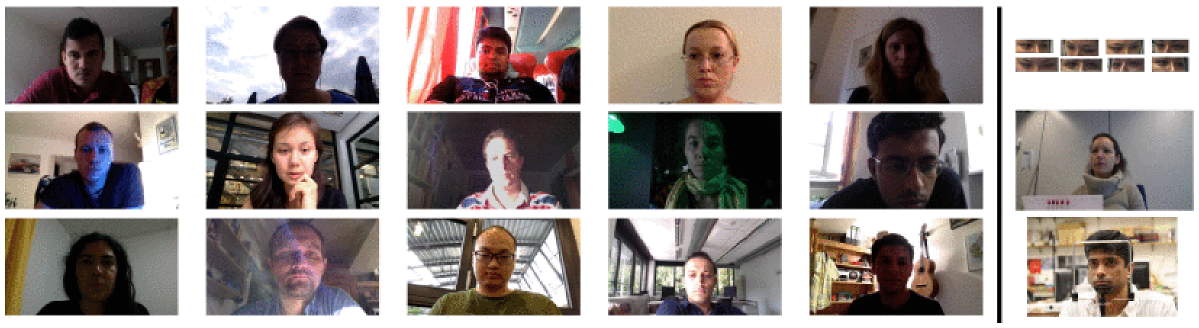

Figure 2.2: Sample images from the MPIIGaze dataset. Source: Zhang et al. [11]

But even with MPIIGaze, all gaze-related datasets were captured using either a static recording setup or limited to a specific device – such as smartphones and tablets [12] or laptops [11].

So another milestone in gaze estimation research is the release of the Gaze360 dataset by Kellnhofer et al. in 2019 [1]. As illustrated in Figure 2.3, the images were captured using a 360° panoramic camera, where more than 230 participants were instructed to look at a sign also captured by the camera.

This way, more than 130,000 images with 3D gaze annotations were captured indoors and outdoors. The subjects were not constrained to look at a screen device, so the dataset covers the entire horizontal range of 360°. Kellnhofer et al., using this dataset, trained and made available their own gaze estimation model, using the same name, ”Gaze360” [1].

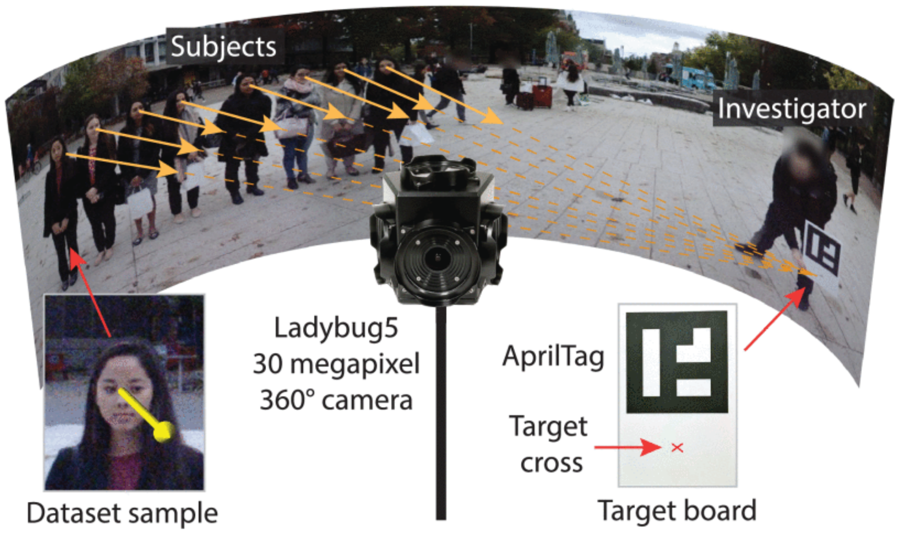

Figure 2.3: Gaze360 image acquisition setup. Source: Kellnhofer et al. [1]

2.1.3 L2CS-Net

As of 2022/2023, the best gaze estimation model publicly available is L2CS-Net [2]. It takes as input an image and returns the predicted gaze direction angle as a pair of (yaw, pitch) rotation angles.

The network was trained and validated on both MPIIGaze and Gaze360. It outperforms the Gaze360 model by using a CNN architecture of 50 layers and testing novel loss functions [2].

L2CS-Net is based on the backbone CNN ResNet [13], but has two fully connected layers in order to regress each gaze angle (yaw, pitch) independently. Figure 2.4 illustrates its architecture. Instead of directly calculating continuous angle values (Yaw angles range from -180° to +180° while pitch angles range from -90° to +90°), L2CS-Net first splits these ranges into 90 different bins; for the size of the fully connected layers is of 90.

When the Softmax function is applied to these layers, it normalizes them into probability distributions [14], where each $\texttt{fc\_layer}[i]$ is equal to the probability of the gaze angle belonging to the bin of index i. As every p in probability distributions is such that $p \in [0,1]$ and the sum of every p equals 1, the value obtained from Equation 2.1 is a value $\in [0,89]$, becoming a measure for angle.

$$\text{unnormalized angle} = \sum_{i=0}^{89} \texttt{fc\_layer}[i] * i \tag{2.1}$$

Equation 2.2 should be used in order to obtain a normalized value for each output angle, for it converts the angle from [0, 89] bin units to [-180°, 180°] degrees.

$$\text{gaze angle} = \text{unnormalized angle} * 4 - 180. \tag{2.2}$$

When the network was trained, two separate loss functions were used for each gaze angle: Cross-entropy loss for angle classification into bins and L2 loss for the continuous value obtained through Equation 2.2. Both losses were backpropagated through the network's layers, fine-tuning them into estimating both gaze direction angles.

L2CS-Net got its name from how it was trained and implemented: L2 loss + Cross-entropy loss + Softmax layer = L2CS [2].

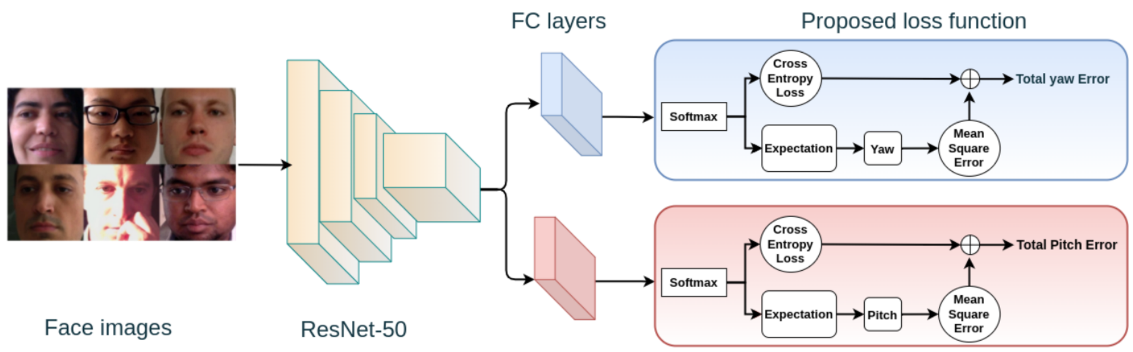

Figure 2.4: L2CS-Net architecture. Source: Abdelrahman et al. [2]

2.2 CLASSIFICATION ALGORITHMS

This section introduces the machine learning algorithms used to train the classification models used in this study.

2.2.1 Decision Tree

Decision Tree classifiers are models that learn to take simple decisions inferred from features of the training data [15]. A decision tree model resembles a directed graph where each node represents a decision over features of the input data, and the leaf nodes represent the result of the classification.

These models are trained by recursively splitting the training data into subsets and grouping the values that have similar features. Once they are trained, Decision trees are simple to understand, for each output of the model can be explained using boolean logic by tracing the path in the graph, but they are prone to overfitting when the training data is large and/or noisy, for the trees become too complex and do not generalize the data well [15].

2.2.2 Random Forest

Random Forest classifiers combine multiple decision trees to improve the accuracy and stability of predictions [16]. Multiple trees are trained using random subsets of the training data; in order to predict the class for the input, each decision tree makes a prediction, and the final prediction is the majority vote of all the trees in the forest. Although Random Forest classifiers are more robust than single Decision Trees, there is still chance that the decision trees in the forest won't generalize the data well, leading to overfitting as well.

2.2.3 K-Nearest Neighbours

KNN classifiers are easy to understand and implement. To predict the class a given input belongs to, it evaluates the k nearest points in the training data, and predicts that the input belongs to the same class of the simple majority of these k points [17].

2.2.4 Support Vector Classifiers

Support Vector Classifiers find the function that separates the data by their classes in a way that the distance between the points of different labels is maximized. When the training dataset is not entirely separable by a function, some misclassifications are penalized but allowed in order to find a function that best splits the data according to their classes.

SVCs were initially designed to work only for linearly separable points, but in 1992, the use of nonlinear kernel functions was proposed in order to have SVCs separate nonlinearly separable points [18]. Examples of nonlinear kernel functions include polynomial, radial basis function (RBF), and sigmoid.

2.2.5 Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic machine learning algorithm based on the Bayes theorem [19]. Using the training data as a sample, for every class, the algorithm calculates the mean and variance for every input attribute of a point x that belongs to the class; modelling a normal/Gaussian distribution of the features. It is called "naive" for it assumes that all features are independent [20].

$$P(\texttt{x[i]} \mid y = \mathrm{c}) = \frac{1}{\sqrt{2\pi * \mathrm{variance}^2[\mathrm{c}, \mathrm{i}]}} \exp\left(-\frac{(\texttt{x[i]} - \mathrm{mean}[\mathrm{c}, \mathrm{i}])^2}{2 * \mathrm{variance}^2[\mathrm{c}, \mathrm{i}]}\right) \tag{2.3}$$

Equation 2.3 returns the probability of x belonging to class c according to the value of its attribute `x[i]`. Multiplying the values of $P(\texttt{x[i]} \mid y = c)$ for every i results in the probability of x belonging to class c.

Calculating this value for every class c results in the probability of x belonging to every class. The Gaussian Naive Bayes predicts x to be part of the class where $P(\mathrm{x} \mid y = c)$ is the largest.

3 RELATED WORK

This chapter presents an overview of existing studies on driver monitoring systems. It reviews the definition of driver monitoring, presents features a system can use to evaluate drivers' attention, and discusses relevant driver monitoring systems from the literature.

3.1 DRIVER MONITORING SYSTEMS

Ortega et al. [9] define "driver monitoring" as the analysis of parameters such as the direction of gaze, head pose, fatigue/drowsiness, disposition of the hands, and more. According to Manstetten et al. [21] , basic research activities on driver inattention started around the year 2000, and the first driver monitoring system based on driving performance was introduced in 2008. Sigari et al. [22] consider that driver monitoring systems have two main challenges: measuring fatigue and measuring concentration.

When developing a driver monitoring system using a CNN, Yang et al. [6] captured many parameters from the driver's face – head pose, gaze direction, yawning, and eye state analysis such as blink rate, blink duration, and eye open/close see Figure 3.1 – and used them all in a single multi-task CNN framework, DANet, bringing together all the data to determine how distracted or drowsy the driver is.

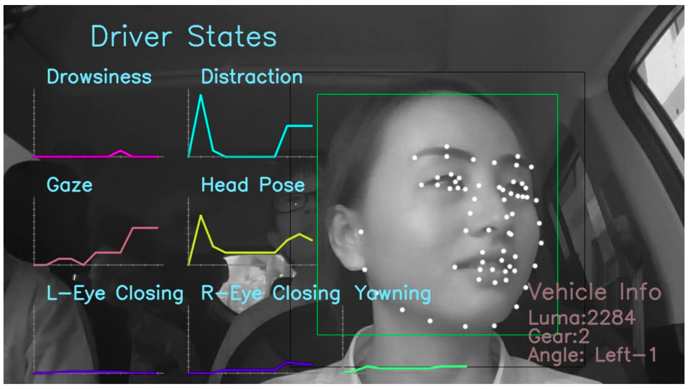

Figure 3.1: A demonstration of the DANet. Source: Yang et al. [6]

In 1999, a study by Häkkänen et al. [23] confirmed that sleepiness increases the average blink closure duration. Since then, monitoring systems can use an eyelid aperture estimator developed by Ortega et al. [24] to detect blink patterns that indicate microsleeping and other fatigue levels.

Some attention-monitoring systems use full-body images to detect distractions such as interacting with passengers or grabbing items like water bottles or telephones

[9]; a camera pointed at the driver's hands could also tell whether the driver is paying attention to what they are doing.

In a study similar to the present one, Shi et al. [7] used a CNN face detector to detect the driver's face and also proposed using a 3D gaze estimation model published in 2017, Full-Face [25] , in order to obtain the driver's gaze direction.

They gathered 4,477 different gaze directions from 30 drivers in simulated and real environments. They classify gaze direction as ”normal driving”, ”looking at the dashboard”, ”looking left”, or ”looking right” – as shown in Figure 3.2.

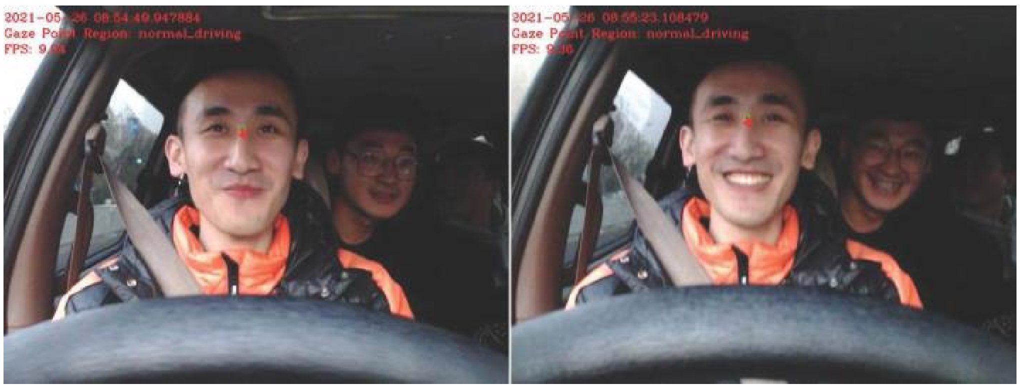
Gaze Point Region: normal_driving
Gaze Point Region: normal_driving

Figure 3.2: Classification of driver's gaze region. Source: Shi et al. [7]

Shi et al.'s proposal is about using gaze estimation to classify the driver's general gaze region (forward, left, right or downwards), but this does not tell the whole picture about the driver's attention. Sure, when the driver's gaze region is forward, the driver is probably paying attention to the road, but the driver can look at relevant road elements that are not directly forward. For instance, their KNN model classifies gaze directions at the car's centre console, passenger, and right rear-view mirror as ”looking right”. The console and the passenger can be visual distractions, but the right mirror is not.

Their definition of ”visual distraction” is simply whether the driver was looking forward or not. The present work differs from the study by Shi et al., for in this study, ”visual distraction” is defined as the state when the driver is not looking at any of the road elements, such as the front road, rear mirrors, or left/right windows; broadening the scope of visual distraction beyond Shi et al.'s definition.

4 PROPOSAL

This chapter introduces the proposed method to obtain a visual attention classifier to monitor whether the driver is visually distracted at a given frame. It presents the dataset used for training and introduces the proposed 3-step pipeline used to train the visual attention classifier and implement a simple driver monitoring system.

4.1 DRIVER MONITORING DATASET

The visual attention classifier is trained with gaze directions of images from driving footage, with labels for when the driver is looking at road elements (windscreen, mirrors, etc.) or not. For that, the Driver Monitoring Dataset [9] published by Ortega et al. in 2020 shall be used.

The DMD contains annotated footage not only of participants driving in a car, but also of them sitting in a parked car, or driving in a simulator indoors. That's because the dataset covers many different actions, some of which would be dangerous for the participants to do while driving in real-life conditions – texting, talking on the phone, combing their hair, reaching for objects in the back seat, etc. Annotations include ”gaze on road”, ”driver is talking”, ”hands using wheel”, ”hand on gear”, ”objects in the scene”, ”occlusion in cameras”, and driver actions such as ”texting”, ”phone call”, ”reach backseat”, and etc.

For each recording session, the dataset offers videos from three different angles (body, hands, and face) in three different formats (RGB, depth, and infrared), as illustrated in Figure 4.1. There is footage of 35 different participants, and the videos range from 1 to 8 minutes.

In this work, only footage from the RGB face camera is needed, as the visual attention classifier will only use gaze data obtained from L2CS-Net output. And of all annotations for distraction, only ”Gaze on Road” labels – `looking_road` (`LR`) and `not_looking_road` (`NLR`), see Figure 4.2 – shall be used as the expected value for training the visual attention classifier for each frame.

4.2 A VISUAL ATTENTION CLASSIFICATION PIPELINE

This study proposes classifying the driver's attention in a 3-step pipeline, as represented in Figure 4.3. First, it acquires the region of interest of the image, which is the driver's face. Then a gaze estimation model predicts the driver's gaze direction. Last but not least, a visual attention classifier is used to analyze whether the gaze direction means the driver is distracted or not.

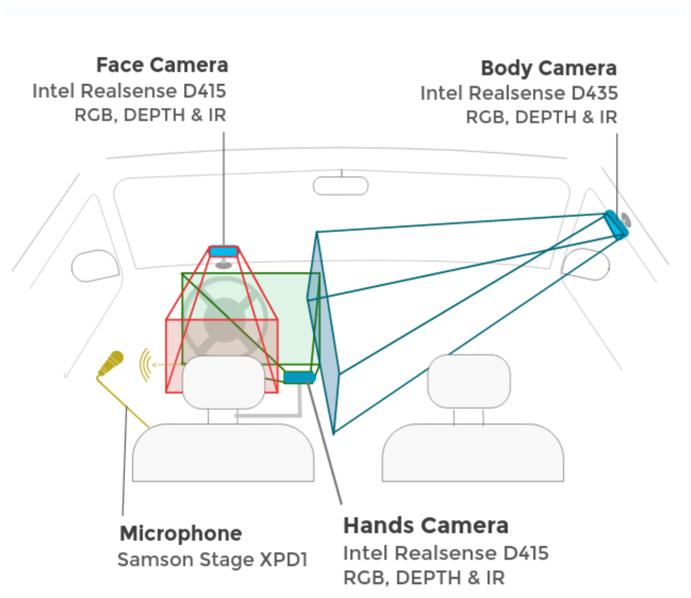

Figure 4.1: DMD camera setup for both real car and simulator scenarios. Source: Ortega et al. [9]

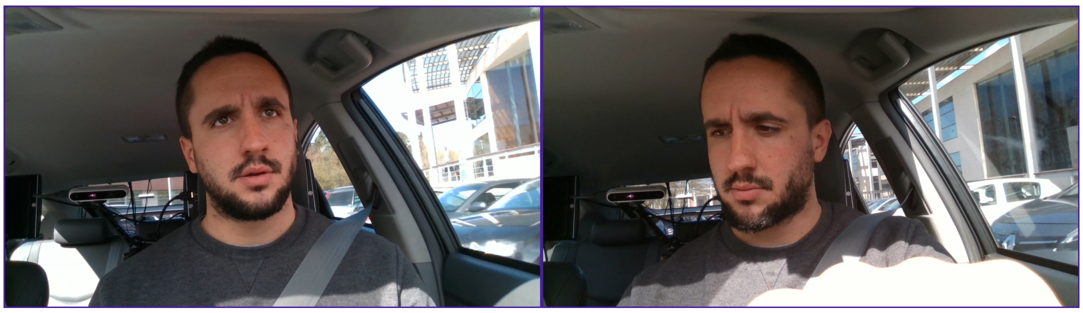

Figure 4.2: DMD footage examples for ”Gaze on Road” annotations. The image on the left is labelled as `LR` while the one on the right is labelled as `NLR`. Source: Ortega et al. [9]

4.2.1 Face Detection

A face detection module based on RetinaFace [26] is used to crop the input image and feed it to the 3D gaze estimator. However, only frames where the driver’s face can be detected with a score of more than 95% are selected in order to get accurate gaze directions from the estimation model and avoid misclassification of gaze direction. When no such face is detected by the model – for reasons such as occlusion caused by the driver’s hands or other objects in the car – then no image should not be given as input to the gaze estimator. The driver monitoring system should ignore and tag as ”unable to decide” frames in which no person is identified.

4.2.2 3D Gaze Estimation

L2CS-Net is the gaze estimation model chosen for this work. It expects an image of resolution (224, 224) pixels as input and returns the predicted gaze direction angle as a pair of (yaw, pitch) rotation angles. Refer to Section 2.1.3 for more details.

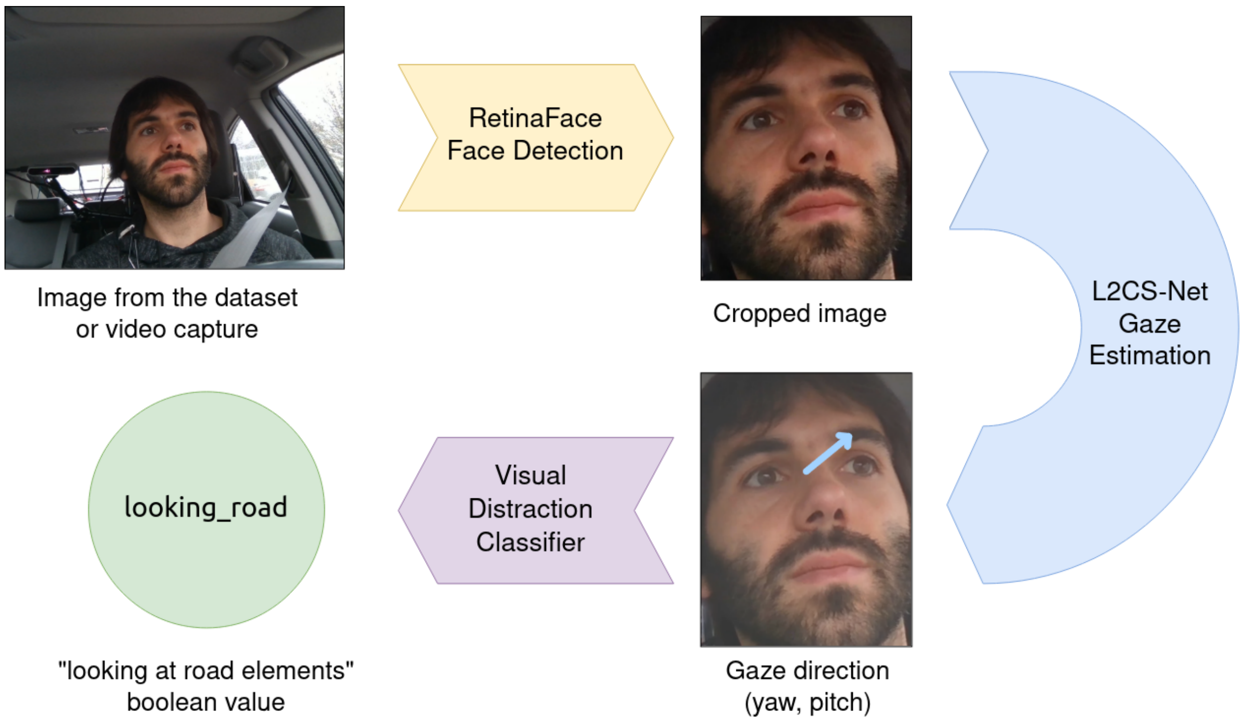
Image from the dataset or video capture
RetinaFace Face Detection
Cropped image
L2CS-Net Gaze Estimation
Gaze direction (yaw, pitch)
Visual Distraction Classifier
looking_road
"looking at road elements" boolean value

Figure 4.3: The proposed visual attention classification pipeline.

4.2.3 Gaze direction classification

Finally, the machine learning library ”scikit-learn” [27] is used to train algorithms such as DT (Decision Tree), SVC (Support Vector Classifier) with Polynomial and RBF (Radial Basis Function) kernels, KNN (K-Nearest Neighbours), RF (Random Forest), and GNB (Gaussian Naive Bayes) – refer to Section 2.2 for more details.

The best model will be chosen as the visual attention classifier, taking (yaw, pitch) gaze direction angles as input and returning whether the driver is in visual distraction or not as a boolean value. The hyperparameters for each classifier are shown in Table 4.1. If a hyperparameter is not defined, its default value in ”scikit-learn” is used instead.

Table 4.1: Hyperparameters for each trained classifier.

Classifier	Hyperparameters
RBF SVC	`kernel='rbf', gamma='scale'`
KNN	`n_neighbors=30, metric='euclidean'`
Gaussian NB	`priors=None`
Random Forest	`n_estimators=100, criterion='gini'`
Polynomial SVC	`kernel='poly', gamma='scale', degree=3`
Decision Tree	`criterion='gini', splitter='best'`

5 RESULTS AND DISCUSSION

This chapter discusses the results of applying L2CS-Net on the DMD dataset, as graphs linking gaze direction angles and distraction are displayed and analyzed. Training results for the visual attention classification models are also presented in this chapter, and the resulting simple driver monitoring system is introduced.

5.1 GAZE DATA ACQUISITION RESULTS

The results of applying the L2CS-Net on every frame from the DMD dataset can be seen in Figure 5.1, as each gaze direction is labelled according to the value of "Gaze on Road" from the dataset annotations.

Notice that there are many times where directions labelled as `NLR` overlap regions marked as `LR` even when the driver is looking forward (values close to 0+, 0). This happens because when using all 730,637 frames collected from the dataset, the footage includes situations where the participants are not really driving, but parked and performing various actions such as combing their hair and other distraction-related actions mentioned before, which may not be the best for classifying attention through gaze. Using this dataset, the best classification model only achieved 89% accuracy.

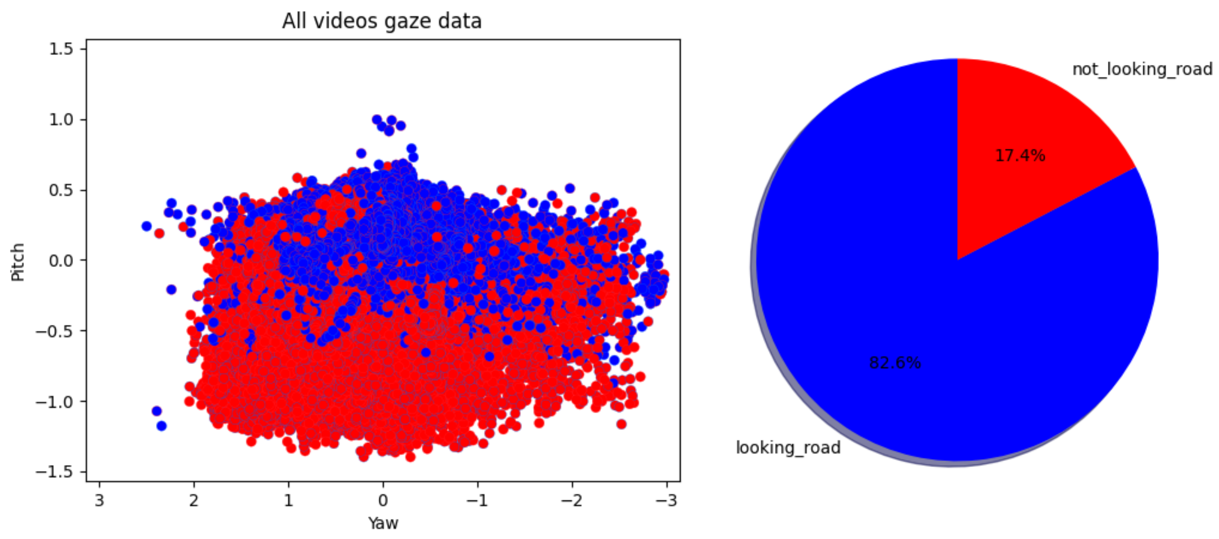

Figure 5.1: Gaze data extracted from 730,637 frames of DMD videos. Blue dots represent `looking_road(LR)`, while red dots represent `not_looking_road(NLR)`.

That's why another smaller set of gaze data was acquired to train the visual attention classifiers by using only footage from real driving scenarios. Thirty-five selected videos were used to obtain 209,108 frames where the participants were driving

– more than 114 minutes of footage. From those, 206,170 could be processed by the face detector (refer to Section 4.2.1) – generating a dataset more than 46 times larger than Shi et al.'s mentioned in Chapter 3. This labelled subset of gaze direction data is also available for future research.

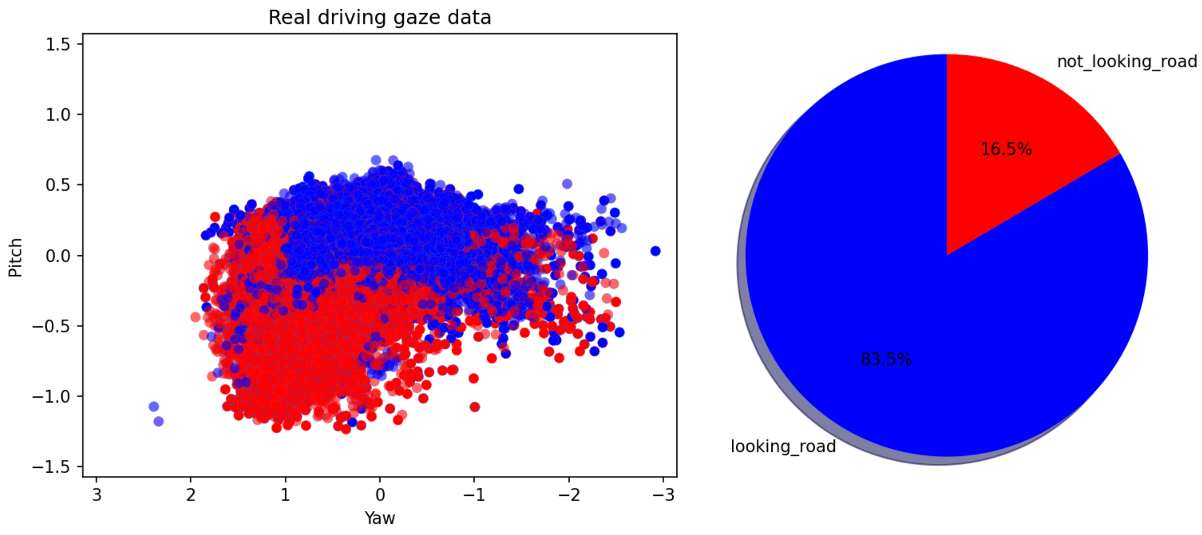

Figure 5.2: Gaze data extracted from 206,170 frames of real driving DMD videos. Blue dots represent `looking_road(LR)`, while red dots represent `not_looking_road(NLR)`.

Figure 5.2 shows that labels are scattered much cleaner than before, and seem to cluster naturally. On both opposite sides along the Yaw axis, however, it is hard to judge whether the driver has their attention on the road. In this dataset, when looking too far to the left or to the right, the driver can be talking to a passenger, checking whether it is safe to change lanes, looking at a side-view mirror, or getting a glimpse of something completely unrelated to road elements.

Although the data is plotted in a way that seems to be scattered and well distributed, in more than 83% of the footage, frames are labelled as ”looking at road elements”. The region around the centre (0, 0) is dense with LR-labelled dots.

It is also noticeable that when driving, distracted participants were primarily looking at the radio system/infotainment or cell phones positioned at the same region (negative Yaw values); while in videos where the participants are in a parked car, in the first dataset, they can look at the steering wheel, dashboard, or at their laps.

5.2 CLASSIFICATION RESULTS

In total, six classifiers (Decision Tree, Polynomial SVC, Random Forest, Gaussian Naive Bayes, KNN and RBF SVC) were trained using the smaller filtered dataset, and their decision boundaries are shown in Figure 5.3. Out of the 206,170 gaze directions acquired from the DMD, 136,581 were used for training (25 videos) and 69,589 for testing (10 videos).

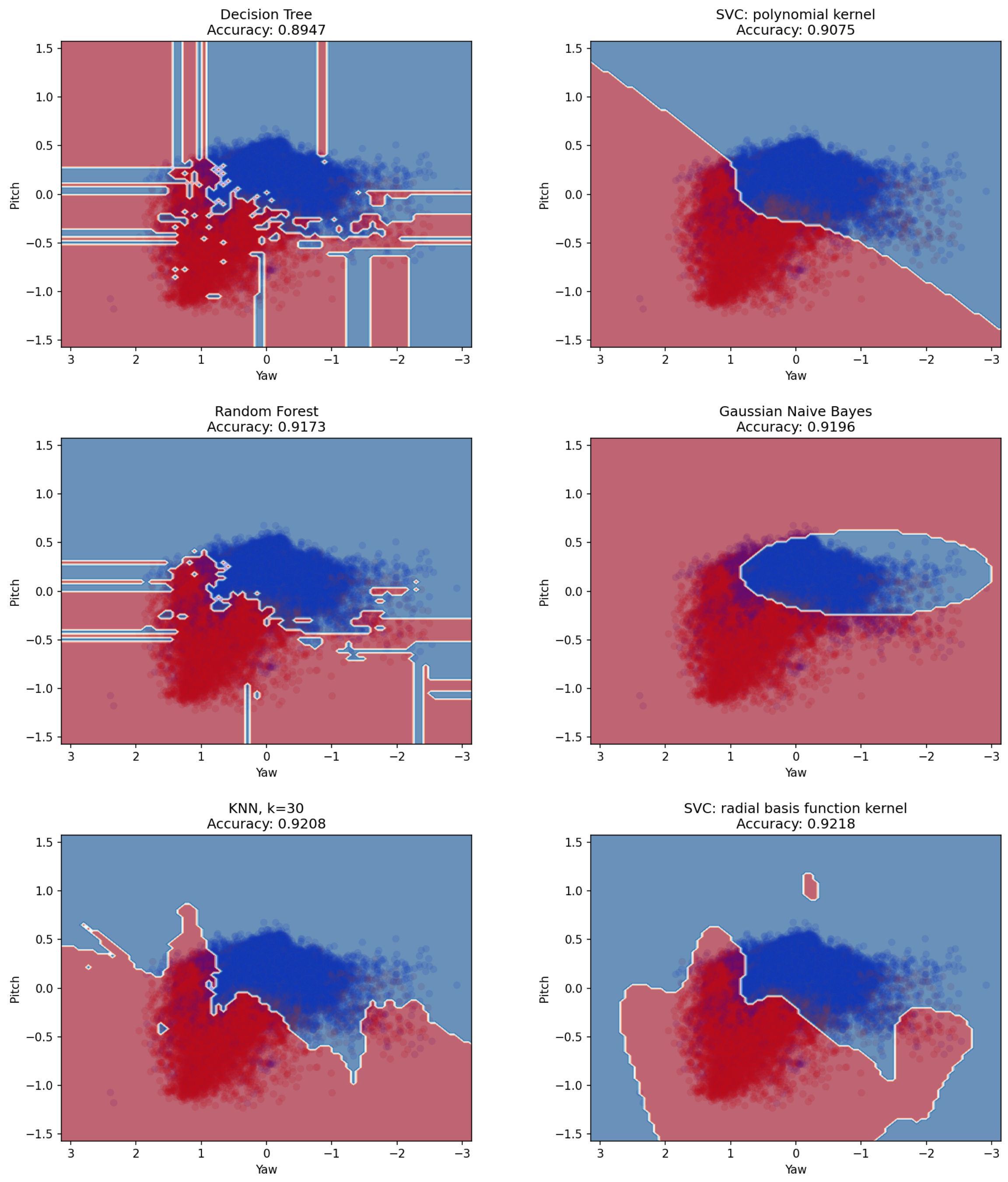

Figure 5.3: Decision boundaries for all trained classifiers, ordered by their accuracy. Light blue areas are classified as `LR`, while red zones are classified as `NLR`.

Even though the boundaries are different for each model, their general accuracy score is somewhat close, ranging from 89% to 92%. It can be seen in Table 5.1 that the RBF SVC classifier has the best general accuracy (92%), but it does not have the best accuracy for predicting `not_looking_road`.

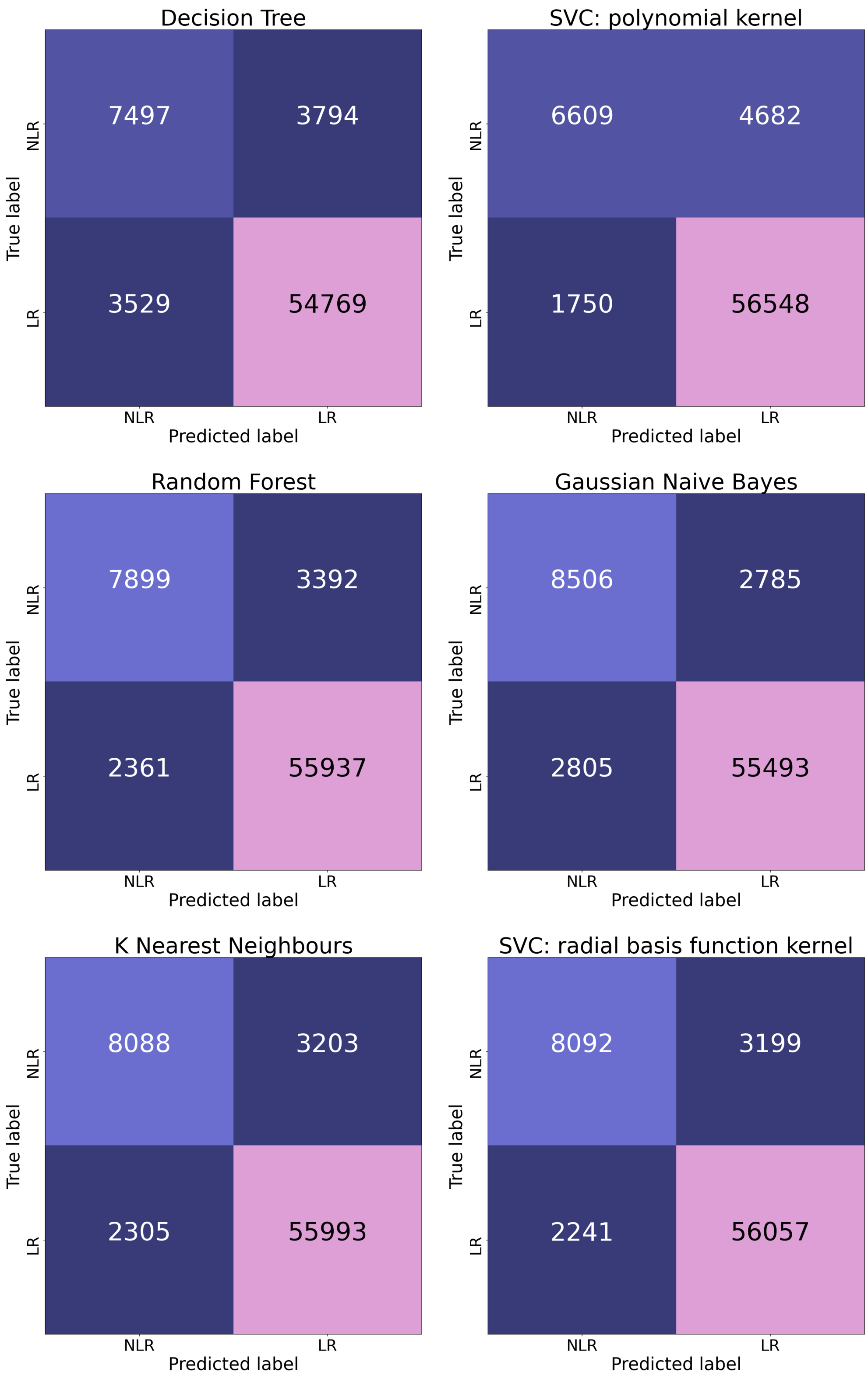

Figure 5.4: Confusion matrixes for all trained classifiers, ordered by their accuracy. The testing subset has 69,589 directions, where 11,291 are labelled as NLR and 58,298 are labelled as LR.

The DT, Polynomial SVC, and RF models have the worst general accuracy and the worst NLR accuracy as well. Even though the Polynomial SVC has the best NLR precision according to Table 5.1, it has the worst NLR accuracy of them all, for the data

Table 5.1: Classifier comparison based on accuracy and precision for each label.

Classifier	Accuracy ↑	NLR precision	NLR accuracy	LR precision	LR accuracy
RBF SVC	0.9218	0.78	0.71	0.95	0.96
KNN	0.9208	0.78	0.71	0.95	0.96
Gaussian NB	0.9196	0.75	0.75	0.95	0.95
Random Forest	0.9173	0.77	0.69	0.94	0.96
Polynomial SVC	0.9075	0.79	0.58	0.92	0.97
Decision Tree	0.8947	0.68	0.66	0.94	0.94

is not polynomially separable with a degree of three. The decision boundaries for the DT and RF classifiers in Figure 5.3 suggest overfitting as well.

All classifiers achieved a great LR precision score for a simple reason: the vast majority of the LR-labelled gaze directions are located around the centre of the graph, so any model that classifies values close to the centre as LR gets more than 90% looking_road precision.

The Gaussian Naive Bayes classifier is the most interesting of all models. It has one of the best general accuracy scores, and Table 5.1 confirms that it has the best accuracy for classifying NLR. According to the confusion matrix in Figure 5.4, it correctly classified 8,506 NLR directions out of 11,291. It also shows that it is the most likely to declare a gaze direction as NLR, classifying as visual distraction $8,506 + 2,805 = 11,311$ out of the subset of 69,589 gaze directions used for testing.

By looking at its decision boundaries, the GNB also seems to be the best fit for the visual attention problem, even outside the dataset's bias. It chooses to classify as LR only gaze directions near the centre of the graph, and everything that differs from it is considered NLR. Therefore, it would classify gaze directions not seen before in the training dataset – such as looking at the ceiling (high Pitch values) – as ”visual distraction”, when the other two best classifiers (KNN and RBF SVC) would not, for lack of drivers' gaze direction looking up labelled as ”visual distraction”. The GNB's decision boundary can be explained by its design: it assumes that the distribution of Yaw and Pitch for LR-labelled gaze directions are Gaussian, and because most of the LR gaze directions are close around the centre, the variance of the distribution is low. As values far from the centre have less probability of being in this Gaussian distribution, they are classified as NLR.

Based on this discussion, it is reasonable to conclude that the Decision Tree, Polynomial SVC, and Random Forest classifiers do not offer any advantages over the other three classifiers. Therefore, the KNN, RBF SVC, and especially the GNB are the ones recommended for classifying drivers' attention, as they have been shown to be more suitable for the task. These three classifiers are made available for the use of other researchers.

As mentioned in Section 5.1, however, the dataset's bias of labelling gaze directions to the right as visual distraction – as "talking to the passenger" is a common action – is reflected on every resulting classifier. Every classifier categorizes gaze directions too far to the right (Yaw > 1) as `NLR`, while gaze directions to the left are classified as `LR` – for there is no passenger to be looked at on the left.

5.3 A SIMPLE DRIVER MONITORING SYSTEM

By implementing the pipeline mentioned in Chapter 4 (refer to Section 4.2), to put the visual attention classifier in use, a simple monitoring system was developed and made publicly available in order to assist future researchers. For each frame, the driver's face is cropped, the gaze estimator processes the image, and the gaze direction is classified. Using the GNB classification model, the video frame is captioned as "Looking at road elements" or "Visually distracted" – see Figure 5.5.

Notice that when the face detector fails to recognize the driver, for there is no driver in the image or they are occluded by other objects in the scene, the program can't classify the driver's visual attention.

5.4 COMPARISON TO PREVIOUS STUDIES

Shi et al.'s [7] study mentioned in Chapter 3 used a gaze estimation model, FullFace [25], in order to classify the driver's general gaze zone. This gaze estimation model was trained in 2017 using only the MPIIGaze dataset [11], before the release of the Gaze360 dataset [1]. For that and other reasons, they could not properly differentiate gaze directions far from the centre of the camera.

By using a gaze estimation model that outperforms the one used in Shi et al.'s study, this work is able to differentiate some of the gaze directions that the previous study could not. For instance, even though this study cannot differentiate looking at the passenger from looking at the right mirror, it can tell apart the infotainment/radio system from the passenger and right side mirror, when Shi et al.'s could not.

And by using the Driver Monitoring Dataset's definition of "looking at road elements", this study does not classify the driver's gaze directions as just looking forward or not, but takes on the task of classifying drivers' attention to any relevant road elements. Using the DMD also allowed this study to use more than 206 thousand labelled gaze directions, a dataset more than 46 times larger than the one obtained by the previous study.

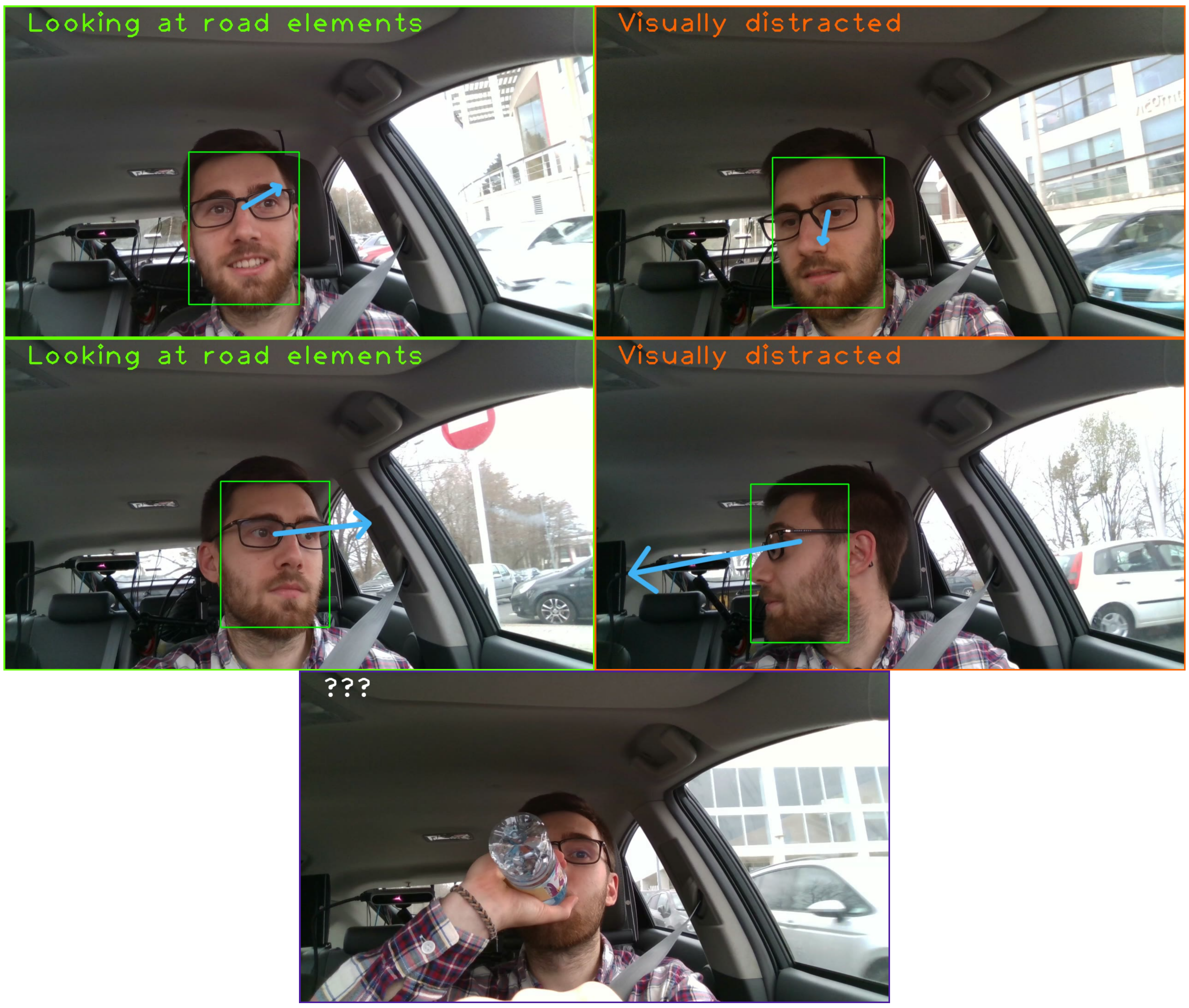

Figure 5.5: The driver monitoring system captions frames from a video by using the Gaussian Naive Bayes visual attention classifier.

6 CONCLUSION

This chapter concludes the study by summarising what it achieved. It also provides possible applications for the classification models trained in this study. Finally, it presents opportunities for future research on the use of gaze estimation in research on drivers' attention.

6.1 SUMMARY

3D gaze estimation models with less than 10° of angular error are already publicly available, and this study used one to develop a pipeline to classify drivers' visual attention, training multiple models to classify drivers' gaze direction angles as "looking at road elements" or not. The classifiers and a simple monitoring system are made available to the general public.

The initial goal of testing L2CS-Net as a tool to understand human attention in a case study for classifying driver's attention was accomplished. By using this gaze estimation model in the proposed visual attention classification pipeline, the complex task of deciding whether a driver is visually distracted was reduced to classifying points.

6.2 POSSIBLE APPLICATIONS

The resulting visual attention classifier can be used in real-life applications such as research on drivers' attention; or in a monitoring system embedded in a real car, to warn the driver through sound cues that they must look where they are going. Modern autonomous cars could also, under extreme conditions, take control of the vehicle when the user that is manually driving happens to be visually distracted for too long.

6.3 FUTURE WORK

There is more to be explored regarding the idea of using gaze estimation models in research on drivers' attention. For instance, future work may be to analyze more footage and recognize which regions drivers look at more often. During the development of this work, DMD researchers started to label the dataset on where the driver is looking: steering wheel, left mirror, rearview mirror, infotainment, etc.

One of the reasons why classifying attention was difficult when the driver was looking at their surroundings was because the DMD included staged distractions such as "talking to passengers", which biased the algorithms into classifying every glance to the right as ”distraction”. In this study, the location of the driver’s face in the image was not used as a parameter for classifying their attention, for the participants in the dataset had different heights; but in hindsight, it could be useful to separate glances to the side-view mirrors from glances to the passenger. Future work could use the driver’s location in the image as a parameter for classifying visual attention.

Still on the topic of differentiating passengers and glances to the right, future work could evaluate whether a visual distraction classifier trained using other videos would perform better. Future researchers could also map when a passenger is present, for the driver can’t be distracted by a passenger if they are not in the car.

Other ML classification algorithms can be explored in the future, such as gradient boosting and neural networks. In a quick test, the Python module ”auto-sklearn” [28, 29] hints that it is possible to train a classifier with up to 94% accuracy using the same dataset presented in chapter 5.

Finally, this work does not consider the scenario outside of the car to know where the driver should be looking: when switching lanes, for instance, the driver is supposed to be looking at their side, not just forward; and in a straight line, looking at the sides for too long could be a sign of visual distraction. Future work could integrate a dashboard camera pointing forward to understand its surroundings and classify the driver’s attention even more accurately. Also, if researchers had access to both a camera pointed to the front road and a skilled driver’s gaze directions, they could use them to teach autonomous car systems to look where they are going.

REFERENCES

[1] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[2] Ahmed A. Abdelrahman, Thorsten Hempel, Aly Khalifa, and Ayoub Al-Hamadi. L2cs-net: Fine-grained gaze estimation in unconstrained environments, 2022.

[3] Ruohan Zhang, Akanksha Saran, Bo Liu, Yifeng Zhu, Sihang Guo, Scott Niekum, Dana Ballard, and Mary Hayhoe. Human gaze assisted artificial intelligence: A review. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4951–4958. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Survey track.

[4] Davide Massaro, Federica Savazzi, Cinzia Di Dio, David Freedberg, Vittorio Gallese, Gabriella Gilli, and Antonella Marchetti. When art moves the eyes: a behavioral and eye-tracking study. *PLoS One*, 7(5):e37285, May 2012.

[5] Stephen Hutt, Caitlin Mills, Shelby White, Patrick J. Donnelly, and Sidney K. D'Mello. The eyes have it: Gaze-based detection of mind wandering during learning with an intelligent tutoring system. In *EDM*, 2016.

[6] Dawei Yang, Xinlei Li, Xiaotian Dai, Rui Zhang, Lizhe Qi, Wenqiang Zhang, and Zhe Jiang. All in one network for driver attention monitoring. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2258–2262, 2020.

[7] Huili Shi, Longfei Chen, Xiaoyuan Wang, Gang Wang, and Quanzheng Wang. A nonintrusive and real-time classification method for driver's gaze region using an rgb camera. *Sustainability*, 14(1), 2022.

[8] World Health Organization. *Global status report on road safety 2018*. World Health Organization, 2018.

[9] Juan Diego Ortega, Neslihan Kose, Paola Cañas, Min-An Chao, Alexander Unnervik, Marcos Nieto, Oihana Otaegui, and Luis Salgado. Dmd: A large-scale multi-modal driver monitoring dataset for attention and alertness analysis. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 387–405. Springer International Publishing, 2020.

[10] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010.

[11] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):162–175, 2019.

[12] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2176–2184, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[14] Wikipedia. Softmax function — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Softmax%20function&oldid=1136143942`, 2023. [Online; accessed 15-February-2023].

[15] Brian Holt. Decision trees — Scikit-Learn. `https://scikit-learn.org/stable/modules/tree.html`. [Online; accessed 1-March-2023].

[16] Gilles Louppe. Ensemble methods — Scikit-Learn. `https://scikit-learn.org/stable/modules/ensemble.html#random-forests`. [Online; accessed 1-March-2023].

[17] Jake Vanderplas. Nearest neighbors — Scikit-Learn. `https://scikit-learn.org/stable/modules/neighbors.html#classification`. [Online; accessed 1-March-2023].

[18] Wikipedia. Support vector machine — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Support%20vector%20machine&oldid=1129233078`, 2023. [Online; accessed 01-March-2023].

[19] Rohan Vats. Gaussian naive bayes: What you need to know? `https://www.upgrad.com/blog/gaussian-naive-bayes/`. [Online; accessed 1-March-2023].

[20] Stefan Hrouda-Rasmussen. (Gaussian) Naive Bayes. `https://towardsdatascience.com/gaussian-naive-bayes-4d2895d139a`. [Online; accessed 1-March-2023].

[21] Dietrich Manstetten, Frank Beruscha, Hans-Joachim Bieg, Fanny Kobiela, Andreas Korthauer, Wolfgang Krautter, and Claus Marberger. The evolution of driver monitoring systems: A shortened story on past, current and future approaches how cars acquire knowledge about the driver's state. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '20, New York, NY, USA, 2021. Association for Computing Machinery.

[22] Mohamad-Hoseyn Sigari, Muhammad Pourshahabi, Mohsen Soryani, and Mahmood Fathy. A review on driver face monitoring systems for fatigue and distraction detection. *International Journal of Advanced Science and Technology*, 64:73–100, 03 2014.

[23] Helinä Häkkänen, LicPsych, Heikki Summala, Markku Partinen, Mikko Tiihonen, and Jouni Silvo. Blink Duration as an Indicator of Driver Sleepiness in Professional Bus Drivers. *Sleep*, 22(6):798–802, 09 1999.

[24] Juan Diego Ortega, Marcos Nieto, Luis Salgado, and Oihana Otaegui. User-adaptive eyelid aperture estimation for blink detection in driver monitoring systems. In *International Conference on Vehicle Technology and Intelligent Transport Systems*, 2020.

[25] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It's written all over your face: Full-face appearance-based gaze estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jul 2017.

[26] Jiankang Deng, Jia Guo, Zhou Yuxiang, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. In *arxiv*, 2019.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems 28 (2015)*, pages 2962–2970, 2015.

[29] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *arXiv:2007.04074 [cs.LG]*, 2020.