



**TALLINN UNIVERSITY OF TECHNOLOGY**

SCHOOL OF ENGINEERING

Department of Electrical Power Engineering and Mechatronics

**LOW-COST IMAGE PROCESSING SOLUTION FOR DRIVER DROWSINESS  
ASSESSMENT**

**ODAV PILDITÖÖTLUSLAHENDUS AUTOJUHI UNISUSE HINDAMISEKS**

MASTER THESIS

Student: Alejandro Noé Díaz Vargas

Student code: 194225MAHM

Supervisor: PhD. Alar Kuusik, Senior researcher  
of T.J. Seebeck Dept of Electronics

Tallinn 2021

(On the reverse side of title page)

**AUTHOR'S DECLARATION**

Hereby I declare that I have written this thesis independently.  
No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

"....." ..... 20....

Author: .....  
/signature /

Thesis is in accordance with terms and requirements.

"....." ..... 20....

Supervisor: .....  
/signature/

Accepted for defence.

"....." .....20... .

Chairman of theses defence commission: .....  
/name and signature/

**Non-exclusive License for Publication and Reproduction of Graduation Thesis<sup>1</sup>**

I, Alejandro Noé Díaz Vargas (date of birth: 02/12/1994) hereby

1. grant Tallinn University of Technology (TalTech) a non-exclusive license for my thesis  
Low-cost image processing solution for driver drowsiness  
assessment

\_\_\_\_\_  
(title of the graduation thesis)

supervised by

Alar Kuusik\_\_\_\_\_

(Supervisor's name)

1.1 reproduced for the purposes of preservation and electronic publication, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright;

1.2 published via the web of TalTech, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright.

1.3 I am aware that the author also retains the rights specified in clause 1 of this license.

2. I confirm that granting the non-exclusive license does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

---

<sup>1</sup> *Non-exclusive License for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university's right to reproduce the thesis only for preservation purposes.*

\_\_\_\_\_ (signature)

\_\_\_\_\_ (date)

## Department of Electrical Power Engineering and Mechatronics

### THESIS TASK

**Student:** Alejandro Noé Diaz Vargas, 194225MAHM  
Study programme, MAHM02/18 - Mechatronics  
main speciality:  
**Supervisor(s):** PhD. Alar Kuusik, Senior researcher of T.J.  
Seebeck Dept of Electronics, +372 5020210  
Engineer Even Sekhri, +372 6203306

#### Thesis topic:

(in English) Low-Cost Image Processing Solution for Driver Drowsiness Assessment

(in Estonian) Odav Pilditöötluslahendus Autojuhi Unisuse Hindamiseks

#### Thesis main objectives:

1. Selection of perspective image-based face-tracking algorithm(s) for real-time drowsiness detection.
2. Selection of an appropriate low-cost embedded CV platform for real-time video processing.
3. Porting and optimization of selected algorithm(s) to the target platform.

#### Thesis tasks and time schedule:

No	Task description	Deadline
1.	Literature Review	December, 2020
2.	Computer-Vision platform selection	December, 2020
3.	Study of computer vision algorithms and techniques the selected board supports	February 2021
4.	Development of each software layer of the system, testing and optimization	May 2021

**Language:** English **Deadline for submission of thesis:** ".18" May 2021

**Student:** Alejandro Noé Díaz Vargas ".....".....2021  
/signature/

**Supervisor:** PhD. Alar Kuusik ".....".....2021  
/signature/

**Head of study programme:** Prof. Mart Tamre ".....".....2021  
/signature/

*Terms of thesis closed defense and/or restricted access conditions to be formulated on the reverse side.*

# CONTENTS

PREFACE .....	7
List of abbreviations and symbols .....	8
1. INTRODUCTION .....	10
1.1 Overview .....	10
1.2 Task definition.....	10
1.3 Description of tasks involved.....	11
1.4 Thesis structure.....	11
2. LITERATURE OVERVIEW/ANALYSIS .....	12
2.1 Physiological signals.....	12
2.1.1 Academic related works .....	12
2.1.2 Commercial devices.....	13
2.2 Operating patterns.....	14
2.2.1 Academic related works .....	14
2.2.2 Commercial devices.....	15
2.3 Driver's behavior .....	16
2.3.1 Academic related works .....	16
2.3.2 Commercial products .....	17
2.4 Hybrid measures .....	19
2.5 Chapter conclusion.....	20
3. IMPLEMENTATION PLATFORM .....	22
3.1 Computer vision platforms .....	22
3.2 Board setup .....	24
4. SYSTEM DEVELOPMENT .....	25
4.1 Dataset Imaging .....	25
4.2 System operation .....	27
4.3 Face Detection .....	29
4.3.1 Haar cascade classifier.....	29
4.3.2 Neural network model.....	30
4.4 Eye detection .....	32
4.4.1 Models' details .....	34
4.5 Face orientation.....	39
4.5.1 Key points comparison .....	39
4.5.2 Neural Network approach .....	41
4.6 Eye status.....	45
4.6.1 Color detection .....	45
4.6.2 Pupil detection .....	47

4.6.3 Neural Network classifier .....	48
4.7 Analysis and results .....	51
5. SUMMARY .....	54
5.1 Future work .....	54
6. KOKKUVÕTE.....	55
6.1 Järgmised sammud .....	55
LIST OF REFERENCES .....	56

## **PREFACE**

The topic of this thesis was proposed by senior researcher Alar Kuusik in a joint discussion with the author. This thesis was carried out in Tallinn under the supervision of Alar Kuusik, who provided the necessary hardware for its development.

Computer vision has gained a lot of ground in recent years for the development of solutions that can detect drowsy state in drivers. Currently there is a boom in the development of powerful alternatives for the development of computer vision task prototypes at accessible cost. This thesis merge both paths, with the aim of developing a low-cost portable solution that could assess the state of drivers in real-time.

I would like to thank my supervisor Alar Kuusik, for the support provided along this path that allowed me to complete the work presented in this thesis. I want to dedicate this thesis to my parents, grandparents, and brother, who despite the thousands of kilometers of distance that separate us always supported me and motivated me unconditionally regardless of the difficulties and who during this journey were my light in times of darkness.

Keywords: Computer-Vision, Driver drowsiness detection, Deep Learning, Master Thesis

## List of abbreviations and symbols

<b>ANN</b>	Artificial Neural Network
<b>BD</b>	Binary Division
<b>BP</b>	Back Propagation
<b>BR</b>	Breath Rate
<b>CNN</b>	Convolutional Neural Network
<b>CV</b>	Computer Vision
<b>DBN</b>	Deep Belief Network
<b>DMS</b>	Driver Monitoring System
<b>DT</b>	Decision Tree
<b>EAR</b>	Eye Aspect Ratio
<b>ECG</b>	Electrocardiogram
<b>EEG</b>	Electroencephalogram
<b>FKNN</b>	Fuzzy K-nearest neighbor
<b>GBT</b>	Gradient Boosting Tree
<b>HOG</b>	Histogram of Oriented Gradients
<b>HR</b>	Heart Rate
<b>HRV</b>	Heart Rate Variability
<b>JDS</b>	Johns Drowsiness Scale
<b>KFC</b>	Kernelized Correlation Filters
<b>KPU</b>	Neural network processor
<b>KNN</b>	K-nearest neighbor
<b>LBP</b>	Local Binary Patterns
<b>LSTM</b>	Long Short-Term Memory
<b>MAR</b>	Mouth Aspect Ratio
<b>MCD</b>	Multi Class Division
<b>MLC</b>	Multiple conditions
<b>MOL</b>	Multilevel Ordered Logit
<b>MTCNN</b>	Multi-task Convolutional Neural Network
<b>MVC</b>	Majority Voting Classifier
<b>NR</b>	Not Reported
<b>NSL</b>	Non-Sleepy
<b>NTHU</b>	National Tsuing Hua University
<b>PERCLOS</b>	Percentage of Eyelid Closure
<b>PPG</b>	Photoplethysmography
<b>ORB</b>	Oriented Fast and Rotated Brief
<b>RBF</b>	Radial Basis Function
<b>RF</b>	Random Forest
<b>RIP</b>	Respiratory Inductive Plethysmography
<b>ROI</b>	Region of Interest
<b>SOC</b>	System on a Chip
<b>SL</b>	Sleepy
<b>SVM</b>	Support Vector Machine



**SWM**  
**TEDD**  
**THW**  
**UTA**  
**YD**

Steering Wheel Movement  
Thoracic Effort Drowsiness Detection  
Time Headway  
University of Texas at Arlington  
Yaw Detection

# **1. INTRODUCTION**

## **1.1 Overview**

Falling asleep behind the wheel is one of the most dangerous situations for any driver, that not only endangers driver's life but also lives of other people who use the traffic system. According to the World Health Organization 1.35 million of people die each year around the world in road crashes [1]. It is estimated that around 20% of car crashes in Europe are related to drive in a sleepy way [2]. Statistics show that in 2019 in EU around 22,800 people died in road accidents, they also shown that per every life lost five more people suffer injuries with life-changing consequences, in 2019 it was around 120,000 people. Despite these numbers, Europe remains to be the safest region in the world when it comes to road safety [3].

However, for the current decade (2,020-2,030) EU has set as a goal a reduction target of 50% for deaths and serious injuries towards "Vision Zero" (zero road deaths by 2,050) [3]. For achieving this they have introduced a new regulation that imposes enhanced safety standards. Under the new rules all motor vehicles will have to be equipped with a driver drowsiness and attention warning systems as well as an advanced driver distraction warning. Starting from 2,022 those systems will be mandatory for new models and from 2,024 for existing models [4]–[7]. Driver drowsiness and attention system must be able to assess the driver's alertness through vehicle systems analysis and warning the driver if needed [8]. Currently in the market, efforts to develop such systems are focused on equipping them to the most recent vehicles when they are being assembled in the factory, which creates a gap of solutions available in the market that can be implemented in any vehicle regardless of the vehicle year of production.

## **1.2 Task definition**

The aim of this thesis arises from the previous diploma work carried out by a TalTech alumni in conjunction with Stoneridge Electronics, and it seeks the development, testing and efficiency assessment of a portable image recognition solution for driver drowsiness detection that can run in low-cost embedded systems using non-contact technologies. The system must be capable of detect the driver face, extract the facial landmark features, and use them for predict the driver status as alert, drowsy or distracted. The

system should run in frame rate of at least 20FPS and should provide a classification accuracy equal or greater than 90% with a F1 score equal or greater than 0,85.

### **1.3 Description of tasks involved.**

1. Review of low-cost computer vision (CV) candidate platforms, and selection of target platform.
2. Setup and configuration of CV platform selected.
3. Development of each software component capable of doing the following tasks:
  - a. Face detection
  - b. Head orientation prediction
  - c. Eye localization
  - d. Eye status prediction
4. Check the performance of each developed layer and make modifications based on the obtained results.
5. Assemble each software component into the system and test the entire system.

### **1.4 Thesis structure**

The structure of this thesis is compound by three main chapters, followed by the summary and the references at the end.

At the first chapter the literature review is presented, which consists of present and describe the different academic and commercial approaches for drowsiness detection in its three aspects, driver's behavior, physiological signals, and operating patterns. The chapter ends with a clearer problem statement and the scope of this thesis as conclusion.

The second chapter provides an analysis and comparison between different low-cost CV platforms. It also presents the setup of the selected CV platform and the configuration of the used dataset for run in the selected platform. The third chapter presents the development of the different layer of the system along with a description of the problems faced per layer based on the different approaches tested. It also provides tables that compare the performance of each layer with its respective approach. The chapter ends with the evaluation of the whole system.

## **2. LITERATURE OVERVIEW/ANALYSIS**

### **2.1 Physiological signals**

#### **2.1.1 Academic related works**

The following sections provides a description of recent studies related to the detection of drowsy state. Commonly there are three different approaches for drowsy state detection, by tracking the driving behavior, physiological signals and driving behavior. [9]-[11]

Physiological signals have a visible strong relationship with the drowsy state since they start to change in early stages of drowsy state. The most common physiological signals used for drowsiness detection are respiration, cardiac, brain, ocular and muscle activity. [12], [13]

For monitoring the respiratory activity, the most used parameters are the breathing rate (BR) and the inspiration expiration ratio (I/E). Some authors have used respiratory inductive plethysmography (RIP) belts to monitor the BR by attaching the belt around the abdomen, diaphragm or chest or the driver. Something to consider when using this belts is to differentiate the variations caused by natural movements (speak, cough, sing) against the ones caused by the drowsy state [14], [15]. Other researchers chose non-contact methods to monitor BR like Solaz *et al.* whom showed that the variations obtained by RIP belts and the ones obtained by a camera are similar setting the thoracic region as Region of Interest (ROIs) [10].

Kiashari *et al.* also chose non-contact technology but they replace the RGB camera with a thermal camera to monitor the BR and I/E, their results show that as the level of fatigue in the driver is increasing the BR decreases while the I/E increases [16]. The main advantage of using thermal cameras it is that they are not affected by the incidence of light.

For monitoring the brain activity electroencephalography (EEG) is the most common method used. It consists into monitor the relative and absolute power of the delta, theta, alpha and beta waves using dry electrodes. The main advantage of monitoring brain activity is that the variations in each of the mentioned waves is directly associated with an alertness level. However, the incidence of noise caused by the driver movements is something to take into account when using it. [12], [13], [17]

Regarding the cardiac activity, researches have used electrocardiogram (ECG) to measure the heart rate (HR) and perform an absolute power analysis of the variability of it [15]. Other authors like Kundinger *et al.* chose wrist-worn wearable device available on the market to monitor the heart rate variability [18]. However, their results showed lower accuracy compared to traditional ECG. Table 2.1 summarize drowsiness detection research based on physiological signals.

Table 2.1. Summary of physiological signals research.

Feature	Parameter	Method	Limitations	Accuracy	Author
Respiratory activity	BR	RIP, TEDD algorithm.	Required 3 belts attached to the driver.	90% BD	[14]
		RGB Imaging.	Not tested in driving environments.	NR	[10]
	BR and I/E	Thermal Imaging and SVM.	Air blowing from A/C or window affect the system readings.	90% MCD	[16]
Respiratory and heart activity	BR and HR	ECG and RIP	Not tested in driving environments.	NR	[15]
Brain activity	Relative power of $\{\theta, \beta, \alpha\}$ waves.	EEG with dry electrodes.	Driver movements affect the system readings, low battery duration.	88,9% MCD	[17]
Brain and heart activity	Relative & absolute power of $\delta$ wave and HRV.	EEG & ECG with dry electrodes.	High computational complexity, not suitable for real-time.	80% BD	[12]
Heart activity	HRV	PPG (wrist band)	Lower accuracy compared to traditional ECG.	82,7% BD	[18]

### 2.1.2 Commercial devices

StopSleep is a double ring which must be placed on any two fingers of driver's right or left hand. It is equipped with 8 cutaneous sensors that continuously measures the driver's skin conductance and electrodermal activity. As soon as the concentration starts to drop the device alerts the driver. [19]

Life is a wearable device designed by SmartCap that monitors fatigue levels. It uses EEG technology to measure the brain activity and determine the alertness levels. It provides a real time fatigue scale that can be monitored through their smartphone app. [20]

CardioWheel it is a driver assistance system designed by CardioID. It uses ECG technology to monitor the driver's heart rate from the drivers hand through a set of sensors embedded in the steering wheel. [21], [22]

Although the monitoring of the signals allows detecting the state of the drivers, its implementation is not practical since it requires additional devices that the driver must put on during his driving routine, as shown in the commercial products presented in this section. On the other hand, it is notable that currently in the market there are not many alternatives that implement this methodology.

## 2.2 Operating patterns

### 2.2.1 Academic related works

In this area, the researchers use several sensors to obtain information about the movement of the vehicle and its environment. Usually researcher focus their efforts in monitor the steering wheel movement; vehicle deviation and position; and its speed an acceleration. [13]

For monitoring the steering wheel movements (SWM) usually an angle sensor mounted on the steering wheel shaft is used. A set of time and frequency domain features are collected and based on them the drowsy state can be predicted. (Chai *et al.*) [23] and Arefnezhad *et al.*) [24]. Other authors like Lee *et al.* used a smartwatch to monitor the movements of the steering wheel [25]. On the other hand, Wang *et al.* combined the information provided by the steering wheel sensor with the vehicle acceleration[26]. Something to consider when using SWM is that the geometry of the road and presence of obstacles may affect the steering wheel movements. Table 2.2 summarize drowsiness detection research based on the driver's operating patterns.

Table 2.2 Summary of operating patterns research.

Feature	Parameter	Method	Accuracy	Author
Steering Wheel	Angle and velocity	MOL, SVM & BP	62,1-72,9% MCD	[23]
	Angle and velocity	Adaptative neuro fuzzy and SVM	98,1,% BD	[24]
	Angle and velocity	SVM	98,1% MCD	[25]
Steering wheel and acceleration	Angle and velocity, vehicle lateral and longitudinal accelerations.	Random Forest Algorithm	84,8% MCD	[26]

## 2.2.2 Commercial devices

This technique is the most used by vehicle manufacturers companies (e.g., Ford, Mercedes, Lexus, Subaru, etc.) They based on analyzing the steering wheel movement and monitoring the position of the vehicle in the road.

For instance, Lane-Keeping System is a feature of Ford Co-Pilot 360™ technology. It uses a camera for scan the vehicle position between the road lanes. If it detects that the vehicle is reaching the edge of the lane it will alert the driver trough steering wheel vibrations. After a threshold of detections it will display a visual and audible alert [27]. Mercedes-Benz Attention Assistant use the steering wheel data for drowsy driver detection. The system draws a profile of the driver driving style during the first minutes of the drive, and it constantly compares with the steering wheel data. It will display an alert or indicate the nearest rest area when fatigue is detected [28], [29]. Other companies like Volvo and Volkswagen combine the steering wheels movements with tracking the road lanes for their Driving Attention Systems. [30],[31]. Table below summarize operating patterns products available in the market.

Table 2.3. Summary of operation patterns products.

<b>Product</b>	<b>Company</b>	<b>Detection Method</b>
Co-Pilot 360™	Ford [27]	Lane Departure
Safety System+ 2.0	Lexus [32]	Lane Departure
EyeSight	Subaru[33]	Lane Departure
Driver Attention Monitor	Honda[34]	SWM
Attention Assist	Mercedes-Benz [28], [29]	SWM
Driver Drowsiness Detection	Bosch [35]	SWM
Driver Attention Alert	Nissan [36]	SWM
Driver Attention Warning	Hyundai [37]	SWM and Lane Departure
Driver Alert Control	Volvo [30]	SWM and Lane Departure
Driver Alert System	Volkswagen [31]	SWM and Lane Departure
Driver Attention Alert	Mazda [38]	SWM and Lane Departure
Driver Condition Monitor	Jaguar & Land Rover [39]	Not Specified
Driver Attention Warning	Kia [40]	Not Specified

Although it is true this technology is the most used by car manufacturers, it is a sensitive technology since the geometry of the roads affects its accuracy. On the other hand, its implementation is carried out when the vehicles are in the manufacturing stage, generating a gap in its implementation since it is not possible to use it in any vehicle model.

## 2.3 Driver's behavior

### 2.3.1 Academic related works

This method consists of monitoring the following features in the driver: eye movement, facial expressions, and head position, using cameras or sensors. To determine the level of drowsiness researchers monitor the eyes analyzing the Eye Aspect Ratio (EAR) and its difference in a certain number of frames like Hossain and George [41]. Other authors like Fouzia *et al.* and He *et al.* used the blinking frequency [42],[43]. While, Nahar *et al.* and Xu *et al.* track the pupil [44],[45]. Miranda *et al.* used the percentage of Eyelid Closure or PERCLOS [46]. Some of the limitations in these features lie in the incidence of light and the presence of corrective lenses or sunglasses. Some authors like He *et al.* propose using commercial gadgets as the Google Glasses to overcome the incidence of light. However, they found a key limitations related to the battery life [43].

As far as the position of the head is concerned, the angle of inclination of the head is monitored since it is assumed that a driver in a sleepy state cannot have his head straight [47],[48]. One of the important factors to consider when using only head movements is necessary to consider the natural movements of the head to avoid erroneous detection.

For facial expressions, the most used parameter to detect fatigue is yawning, since it is an evident indicator of fatigue. Zhang and Su perform this analysis extracting Spatial Features from images in order to locate the mouth and monitor the yawning [49]. However, good operation is conditioned to certain lighting conditions. Jafari and Soryani proposed an alternative to overcome this constraint using depth data of the image[50]. Although it overcomes the light conditions, the degree of head rotation as well as the use of the hand to cover the yawn represent a new challenge. Jie *et al.* proposed a method for yawning detection based on geometric and appearance features of two regions of interest (ROI), their results show that it's possible to detect the yawning even with the hand covering the mouth and with non-ideal illumination conditions [51].

Other, authors propose methods that combine eye movement and facial expressions, such as Deng and Wu who use the EAR, frequency of blinking and number of yawning to evaluate the fatigue level. They also implemented an illumination enhancement method to pre-process the image before the face detection to overcome the constrains under illumination conditions [52]. On the other hand, Khunpisuth *et al.* proposed a



system using the eye blinking rate and head orientation [53]. Table 2.4 summarize drowsiness detection research based on the driver's behavior.

Table 2.4 Summary of driver's behavior research.

Feature	Parameter	Method	Results	Accuracy	Author
Eye	EAR	Haar Cascade classifier	Illumination dependent.	NR	[41]
	Blink frequency	Shape predictor algorithm	Illumination dependent.	NR	[42]
	Blink frequency	Proximity sensor.	Fast battery consumption.	NR	[43]
	Pupil tracking	Viola Jones and Mean Shift algorithm.	Illumination dependent.	97%	[44]
	PERCLOS	Shape predictor algorithm.	Large model size and model speed.	95% BD	[46]
	Pupil Area	FKNN Classifier	Detection performance weaker compared to EEG.	88,7% BD	[45]
Head	Inclination	Watson	Calibration dependent.	NR	[47]
	Posture estimation	Viola Jones algorithm	Poor posture estimation when the angle is too big.	88,3% BD	[48]
Mouth	Aspect ratio	Canny Edge Detector, CNN & LSTM	Illumination dependent, cannot difference between long time open mouth and yawning.	87,3% BD	[49]
	Aspect ratio	Depth map	Head orientation dependent. Does not work if yawning is covered.	95% YD	[50]
	Aspect ratio	OpenFace, HOG, LBP & SVM	Occlusions independent.	94,6% YD	[51]
Mouth and eyes	EAR, blinking rate, and No. of yawning.	MTCNN, MC-KFC algorithm	Low average processing speed.	93,6% BD	[52]
Face and eyes	Blink frequency and head orientation.	Haar Cascade classifier and Template Matching.	Illumination and skin color dependent.	99,6% Detection	[53]

### 2.3.2 Commercial products

Eagle Light and Eagle Industrial is the system commercialized by Optalert, it consists of a pair of glasses and a smartphone or tablet. It works by measuring the velocity of the driver's eyelid using a LED built into the frame of it, and display the drowsiness score based on the Johns Drowsiness Scale (JDS™) [54], [55]. Vigo is another device with similar operation principle as Eagle glasses in headset form, that tracks the eye and head motion. It also keeps a record of the alertness patterns in its app [56].

Driver Monitoring System (DMS) is a SOC developed by SmartEye that identifies and tracks the eye and head of the driver and sent the processed data to a third party integrator who decides how to manage the data and its application [57]. Other companies like Tobii Technologies [58] and NVIDIA [59] have develop similar platforms.

Guardian is a vision system developed by SeeingMachine. It consists by a camera that tracks the eyes and head driver's position, a controller where the peripherals are connected, and a vibration motor attached to the driver seat that alerts the driver when is fatigated or distracted. According to the functionalities specified on its page, the system is more oriented towards commercial fleets [60]. Interior Monitoring System is a solution developed by Bosch, that consists of a camera integrated to the steering wheel of the car and monitors the eye and face of the driver, the company expects that by 2022 the system goes to production [61].

Driver Sate Warning System is part of the Advanced Driver Assistance System developed by Hyundai Mobis. It tracks the driver's gaze, head position and direction to determine the level of distraction of the driver. It also monitors the eye lid state to detect drowsiness. It will be supplied for commercial vehicles in Korea for the first time in 2021. [62]

Jabil and Eyesight made a partnership to present the Driver Monitoring System (DMS) a system that "see's more" as they stated. The system is divided in driver vision and driver logic. Driver vision detects and recognizes drivers face, eyes, and tracks the head position. While driver logic analyses the data provided by driver vision to determine the state of the driver in terms of distraction, drowsiness, identity, and intent. The device is currently available for vehicle test and development. [63]

Other companies such as Meitrack [64], Stonkman [65] and Jimi IoT [66] had also developed a DMS according to their website demonstrations both systems are capable both systems are capable of detecting when a driver is drowsy, when he is using the cell phone, the absence of the driver and if he is smoking while at the wheel. According to their website, the device provided by Jimi IoT is aimed at monitoring commercial vehicles and fleets, on the other hand when contacting Stonckman they commented that they cannot provide information about their market since it is their secret business, on the part of Meitrack did not have any answers when consulting them.

As it can be seen most commercial products focus their efforts on using this methodology to detect the alertness of drivers, mainly because of the three methodologies presented, the investigations show that it has the highest level of accuracy. It is also important to point that most of the commercial products and the researchers performed are migrating to used technologies that apply CV techniques, it could be related to the fact that using this the number of sensors required decreased compared to the monitoring of physiological devices and drivers do not required to carry additional equipment on them, While it is true that it has limitations related to real-time processing and lighting conditions, the development of new technologies in the CV field are rapidly mitigating these limitations.

## 2.4 Hybrid measures

Recently researchers have been moving towards the feature's fusion for drowsy state detection. Some used a combination of physiological signals with driver's behavior such as Oliveira *et al.* and Abbas who combine heart rate with visual features like EAR, Mouth Aspect Ratio (MAR), no. of yawning and blinking frequency. Their results show that by combining these two methods the detection accuracy for binary and multiclass segmentation obtains a higher accuracy compared to the use of a single feature. [6],[67]

Other authors, such as Gwak *et al.* and Sunagawa *et al.* propose a full hybrid system where they combine the tree features driver behavior, physiological signals, and operating patterns. They also introduce a posture index which is calculated by adding a pressure sensor to the seat of the simulator. Results of both studies show that combining the three methods better results in detection of the driver state are achieved [30],[69]. However, it is important to point that combining all the feature the designing cost of the system may become a major drawback. Table 2.5 summarize drowsiness detection research based on hybrid methods.

Table 2.5 Summary of hybrid measures research.

<b>Feature</b>	<b>Parameter</b>	<b>Method</b>	<b>Accuracy</b>	<b>Author</b>
Physiological and behavioral	HRV, Head pose, pupil diameter, blink frequency.	SVM, RF, ANN, GBT & KNN	74-80% MCD	[6]
	EAR, MAR, HR	CNN & DBN	89-93% BD 94,5% MCD	[67]
Physiological, behavioral, and operating	PERCLOS, HR, THW, X; Y coordinates, No. Blinks, b wave	DT, MVC & RF	88,9% BD	[68]
	PERCLOS, blink frequency, vehicle speed and acceleration, SWM, BR, HR.	RBF	NR	[69]

## 2.5 Chapter conclusion

The research presented in this chapter show the following results:

- Using the method of analyzing the driver's behavior more accurate results can be obtained.
- The accuracy obtained by multiclass division is lower compared to the one obtained by binary division.
- Limitation related to the illumination quality can be overcome doing pre-processing techniques or using depth map methods, however this decreases the processing speed.
- By combining facial features to analyze them an improvement in accuracy can be achieved.
- Most car manufacturing companies deploy systems that analyze the driving operating patterns to determine the level of fatigue.
- Driver drowsiness detection systems developed under driver behavior and physiological signals need to differentiate between driver's natural movements and movements caused by a drowsy state.
- Some gadgets currently commercialized on the market can be adapted to monitor the driver's drowsy level.
- Driver drowsiness detection system developed under analysis of operating patterns need to differentiate between movements due to the geometry of the road and movements related to drowsy state.
- Some car manufacturing companies are moving towards systems based on driver's behavior analysis and expected to deploy their system in their cars in the next couple of years.

- Currently on the market there are few products available based on physiological signal analysis, despite they can provide information on the transition from alert state to drowsy state.
- Systems in development stage using non-contact technologies use a combination of eye and face analysis to determine the level of fatigue.

Despite the great progress that has been made in the design of systems for detection of driver drowsiness there are still certain aspects in which more effective solutions must be sought being these:

- Systems based on facial analysis must improve their accuracy for multi-class division.
- Systems that alert the driver when he is drowsy and indicate the nearest rest zone or that can recommend the most suitable time to rest.
- Design of low-cost portable solutions that can run on embedded systems.
- Smart systems that analyze lighting conditions to pre-process the image only when necessary.
- Smart systems that can monitor the level of fatigue using other facial features when sunglasses do not allow monitoring of the eyes.
- Smart systems based on operating pattern analysis that compare the driving patterns with respect to the road geometry.
- Systems that can communicate with the car Smart Recorder providing the required information.
- Smart systems capable of detecting when an accident occurs and that can alert the corresponding emergency services the location of it.

The aim of this thesis the development, testing and efficiency assessment of a portable image recognition solution for driver drowsiness detection that can run in low-cost embedded systems using non-contact technologies. The solution must be capable to identify the drivers face, extract the facial landmark features, and use them to predict the driver status. For achieving this the following objectives are proposed:

- 1) Select perspective image-based face-tracking algorithm(s) for real-time drowsiness detection.
- 2) Select an appropriate low-cost embedded CV platform for real-time video processing.
- 3) Porting and optimizing selected algorithm(s) to the target platform.
- 4) Solution testing and performance assessment.

### 3. IMPLEMENTATION PLATFORM

#### 3.1 Computer vision platforms

The first stage in the system development was the selection of a target platform. To select the platforms some of the characteristics and functionalities of the commercial devices presented on the previous section were considered. Such as, object detection, availability of connection to external actuators, on-site processing, machine learning inference, capacity of remote connections and that the price range should be lower than commercial physiological devices. The selected platforms are presented in Table 3.1.

Table 3.1 Alternative platforms

<b>Index</b>	<b>Board</b>
A	Maix Cube
B	Nvidia Jetson Nano Developer Board
C	Rock PI N-10 Model A
D	Maix-II Dock
E	ECM3532 AI Vision Board
F	Raspberry Pi 4
G	OpenMV Cam H7

The selected platforms were subjected to a trade study to determine which one was the most viable for the implementation of a system with a frame rate of at least 20 FPS, capable of detecting the face, facial landmarks and classify the driver status. Table 3.2 presents the proposed criterion among with its importance in a scale from 1 to 9 where, 9 indicates a parameter that is extremely strong important, while 1 indicates that the parameter is not crucial for a final selection. The table also presents the criteria weights which were calculated using a linear approach.

Table 3.2 Trade study criteria

<b>Index</b>	<b>Criteria</b>	<b>Importance</b>	<b>Weight</b>
H	Cost	9	17,31%
I	Market availability	7	13,46%
J	Development environment	4	7,69%
K	Scalability	6	11,54%
L	Acceleration Unit	9	17,31%
M	Size	7	13,46%
N	Speaker Interface	5	9,62%
O	Power Consumption	5	9,62%

Table 3.3 and Table 3.4 present the results of the decision matrix as it can be seen the platform with the highest sum was the Maix Cube. The Maix Cube is a development board powered by Kendryte K210 core. The chip is designed for machine vision and machine hearing applications, it is equipped with a neural network processor (KPU) and an audio processor unit (APU ) for processing audio inputs, with a total computing power of 1 TOPS. The SRAM of the chip is divided in two parts, 6 MiB used for general purpose operations and 2MiB allocated for the KPU. It supports models created with popular frameworks such as TensorFlow, Keras, Darknet and Caffe. Model's input and output feature maps are stored in at the KPU memory. Weights and other parameters are stored in the general-purpose memory. [70]



Figure 3.1 Selected platform (Maix Cube) [71]

Table 3.3 Scoring calculation part 1.

Criteria	A		B		C		D	
	PF	W.S.	PF	W.S.	PF	W.S.	PF	W.S.
H	0,21	0,04	0,14	0,02	0,07	0,01	0,19	0,03
I	0,15	0,02	0,15	0,02	0,12	0,02	0,13	0,02
J	0,13	0,01	0,15	0,01	0,15	0,01	0,13	0,01
K	0,11	0,01	0,17	0,02	0,17	0,02	0,15	0,02
L	0,17	0,03	0,15	0,03	0,17	0,03	0,07	0,01
M	0,17	0,02	0,12	0,02	0,12	0,02	0,15	0,02
N	0,20	0,02	0,15	0,01	0,15	0,01	0,20	0,02
O	0,15	0,01	0,12	0,01	0,12	0,01	0,15	0,01
Sum		0,17		0,14		0,13		0,14
Final Score		100		86		78		87

Table 3.4 Scoring calculation part 2.

Criteria	E		F		G	
	PF	W.S.	PF	W.S.	PF	W.S.
H	0,09	0,02	0,16	0,03	0,14	0,02
I	0,13	0,02	0,15	0,02	0,15	0,02
J	0,13	0,01	0,15	0,01	0,13	0,01
K	0,11	0,01	0,17	0,02	0,13	0,01
L	0,17	0,03	0,07	0,01	0,11	0,02
M	0,17	0,02	0,12	0,02	0,15	0,02
N	0,02	0,01	0,13	0,01	0,15	0,01
O	0,17	0,02	0,13	0,01	0,15	0,01
Sum		0,13		0,13		0,14
Final Score		77		81		84

## 3.2 Board setup

Before starting to develop the tests, it was necessary to setup the Maix Cube. It can be programmed using different programming environments [70]. However, for the purpose of this thesis it was decided to use the Micropython development environment. When this environment is used it is required to flash the firmware to the board, by default the Maix Cube comes with the Micropython UI framework [72]. However, for the purpose of this system said framework is not required. For flashing the firmware the host computer must be equipped with certain requirements [73]. After these requirements are installed in the host computer the firmware can be downloaded from the Sipeed website [74] or it can be built from source code using the Sipeed repository [75]. For the development of this thesis the experiments were carried out with firmware V0.6.2.27 in its respective variants described in the table below. Also, other experiments were carried out building the firmware from source. The main reason why different firmware versions were used is because the firmware is stored in the general-purpose memory limiting the amount of memory available for loading the models. Some of this memory issues are pointed in section 4.2.1.

Table 3.5 Firmware types

<b>Firmware</b>	<b>Description</b>	<b>Size</b>
Normal firmware	Includes basic API, kmodel V4 support, NES emulator support, AVI format video support & IDE support	2,01 MB
Minimum firmware with IDE support	Includes basic API & kmodel V3 support	700,25 kB
Minimum firmware with kmodel V4 support	Includes the same as minimum firmware and add kmodel V4 support	1,02 MB
OpenMV kmodel V4 with IDE	Includes the same as minimum firmware add IDE and kmodel V4 support	1,45 MB
Custom firmware	Includes IDE support, video and kmodel V4 support	1,1 MB



## 4. SYSTEM DEVELOPMENT

### 4.1 Dataset Imaging

Before developing the software layers, it was necessary to verify if the selected board was capable of loading and showing on the display the dataset. For this, I had access to two datasets, one prepared by National Tsing Hua University (NTHU) [76] and other one prepared by the University of Texas at Arlington (UTA) [77]. It was decided to select the NTHU dataset since this dataset contained videos under two lighting conditions (day and night) and three conditions of obstruction in the eyes (no glasses, glasses, and sunglasses). According to the MaixPy documentation [78] the board can play encoded videos with the following format:

- Screen size: 320x240
- Compress format: MJPEG
- Video format: .AVI
- Audio format: PCM

To make the videos from the dataset meet the indicated requirements, the FFmpeg software was used to perform the corresponding conversions. For achieving this the code described in the Sipeed repository was used [79]. Figure below presents one frame of a video from the dataset running in the target platform. As it can be seen from the figure the device is capable of play videos with a frame rate closer to 22FPS. Other tests were performed with different size videos, frame rate obtained can be seen in Table 4.1, as it can be seen the average frame rate of the board is 24 FPS.



Figure 4.1 Video frame running on Maix Cube.

Table 4.1 Frame rate results

Video				Size	Frame rate
Subject	Time	Glasses	Condition		
1	Day	Yes	NSL	10,10 MB	26 FPS
10	Night	Yes	MLC	33,60 MB	24 FPS
21	Day	Yes	MLC	81,24 MB	22 FPS
34	Day	No	NSL	9,03 MB	25 FPS

## 4.2 System operation

The developed system in this thesis works according to the flow chart presented in Figure 4.2. After power on the device, the camera is configured, and the four AI models are loaded into the KPU memory. Loading the models to the KPU memory only once at the beginning of the program benefits that the FPS are not seriously affected. If the image is loaded from the SD card it is required to copy the image to the KPU, on the other hand if the image is acquired directly from the camera this step is not required. If the system detects a face in the image, it creates a 36x36 grayscale copy of the image. The system uses this resized image to feed the face orientation model, if it detects that the orientation of the face is inside the allowed poses it continues with the localization of the eyes. If the orientation of the face is not allowed the system considers this orientation as distracted face and it use a timer to track the amount of time that the face is distracted. It was established that if the distraction last more than 1.5 seconds a distraction alarm will be triggered. The 1,5 seconds threshold was selected based on a study that establishes that looking away from the road for two or more doubles the risk of an accident or near crash [80].

As it is mentioned before if the system detects that the face orientation is allowed it will crop the ROI where the face is located and create a 96x96 grayscale copy of the cropped ROI. It uses this copy as input of the eye detection model, which provides as output the location of the center of the eyes. Then it adjusts this output to create two ROI's that surround the left and right eye. It creates a copy of each ROI in a 34x26 grayscale image, that it uses as input of the eye status model. If the model classifies one of the eyes as closed the system use a timer to track the time that the eye has been closed. The 1,5 seconds threshold is used again for triggering the drowsiness alert. The system proceeds will proceed to capture images in a loop until the user turn offs the device in case is using the camera for obtaining images or until all the images are extracted from the SD card.

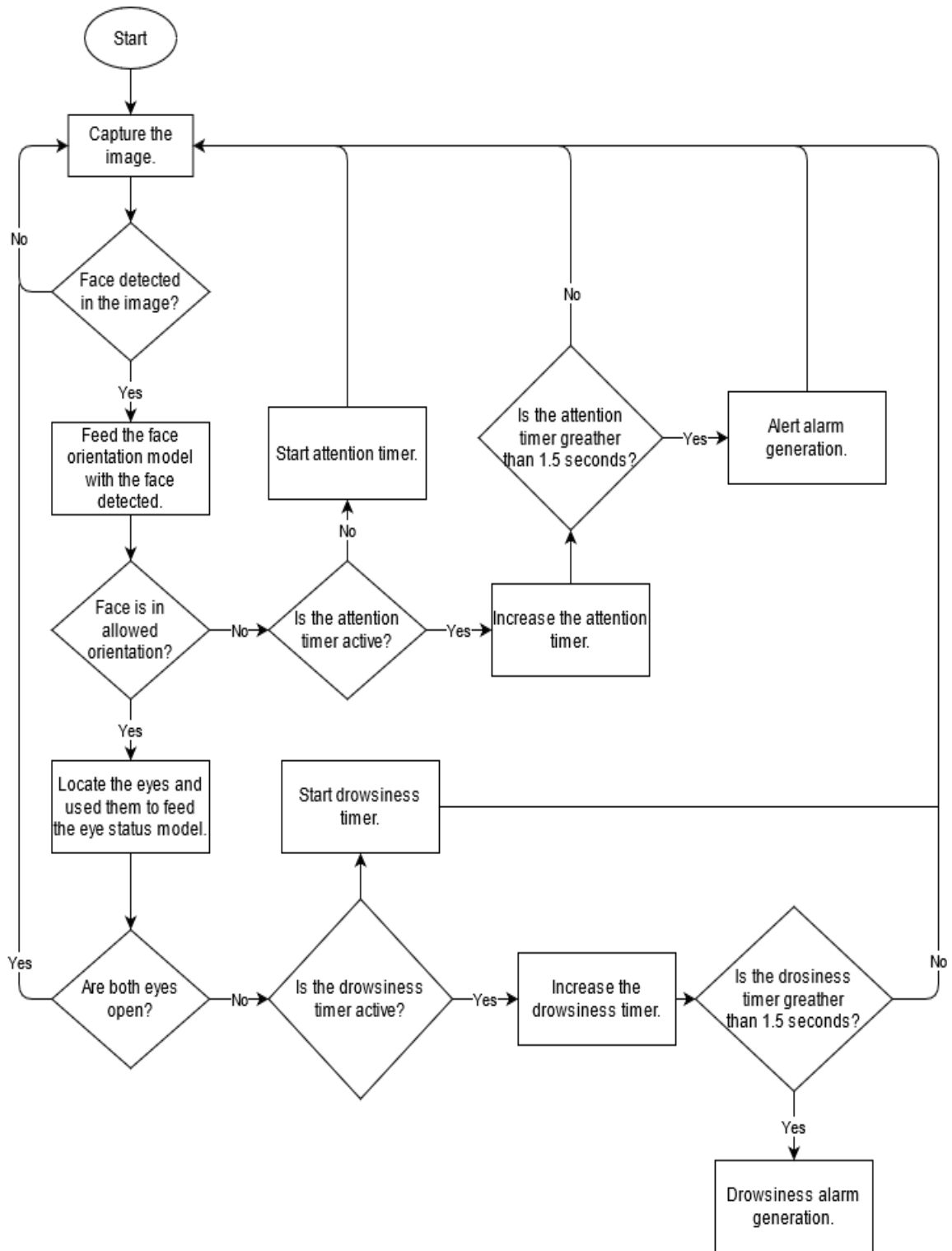


Figure 4.2 System flowchart.

## 4.3 Face Detection

### 4.3.1 Haar cascade classifier

As it was described in the literature review one option to detect drowsiness is to monitor the driver's behavior. For achieving it is important first to identify the face of the driver from the image to proceed with the desired methodology. The first approach tested was the implementation of the Haar Cascade, since the Maix Cube allows to implement it in a simple way using a native method that comes with the firmware. The mentioned method searches in the image for all areas that match with the passed in Haar Cascade and returns a list of bounding box rectangles tuples  $(x, y, w, h)$  around those features [81]. Figure 4.3 presents a series of frames of one of the dataset videos. As it can be seen from the middle and right picture the implemented classifier does not detect the face if it is not properly aligned frontally.



Figure 4.3 Frontal face recognized(left), frontal face not recognized (right), side face not recognized (middle)

Table 4.2 shows the results obtained with videos with different poses of the face. The results presented in the table provide an average value of 52,8%. It is well known that in driving conditions the driver's face constantly changes position, considering this and the average value obtained this approach was ruled out since it is very dependent to the fact that the driver's face is aligned in certain way to the camera to detect the face.

Table 4.2 Haar cascade face detection results

Subject	Video			Total Frames	Frames Detected	Accuracy
	Time	Glasses	Condition			
1	Day	Yes	NSL	1665	89	5,4%
1	Day	No	NSL	1350	362	26,8%
1	Day	No	SL	1335	683	51,2%
4	Night	Yes	MLC	2702	810	29,9%
8	Night	No	NSL	1395	1332	95,5%
10	Day	No	MLC	1120	547	48,8%
10	Night	No	MLC	8565	6613	77,2%
10	Night	Yes	MLC	6594	5936	90,0%
11	Day	No	MLC	1128	848	75,2%
21	Day	Yes	MLC	1194	970	81,2%
23	Night	Yes	SL	1410	6	0,4%
34	Night	Yes	SL	1485	22	1,5%
34	Day	No	NSL	1500	912	60,8%
37	Day	No	MLC	1553	1475	95,0%

### 4.3.2 Neural network model

The second approach for detect the drivers face was based on neural networks. Sipeed provides set of instructions in order to implement an official model based on YoloV2 provided by Sipeed [82]. Figure 4.4 presents a series of similar frames that the one used in the previous section. As it can be seen using by using this model it is possible to detect the face of the driver even when he is looking to the sides, this performance is more reliable for the purpose of this system, since it is not dependent of a frontal face aligned to detect the face.



Figure 4.4 Frontal face recognized(left), frontal face recognized (right), side face recognized (middle)

Table 4.3 shows the face detection results obtained with videos with different poses of the face. The average value of this approach was 98,2% presenting an increase rate detection in a ratio of 1,8 compared to the previous approach. Based on these results, this approach was selected to continue with the development of the system.

Table 4.3 Neural network face detection results

Subject	Video			Total Frames	Frames Detected	Accuracy
	Time	Glasses	Condition			
1	Day	Yes	NSL	1665	1639	98,4%
1	Day	No	NSL	1350	1322	97,9%
1	Day	No	SL	1335	1335	100,0%
4	Night	Yes	MLC	2702	2702	100,0%
8	Night	No	NSL	1395	1395	100,0%
10	Day	No	MLC	1120	1099	98,1%
10	Night	No	MLC	8565	8496	99,2%
10	Night	Yes	MLC	6594	6439	97,6%

Table 4.3 continued

Video				Total Frames	Frames Detected	Accuracy
Subject	Time	Glasses	Condition			
11	Day	No	MLC	1128	1128	100,0%
21	Day	Yes	MLC	1194	1154	96,7%
23	Night	Yes	SL	1410	1379	97,8%
34	Night	Yes	SL	1485	1377	92,7%
34	Day	No	NSL	1500	1494	99,6%
37	Day	No	MLC	1553	1511	97,3%

## 4.4 Eye detection

After selecting the approach that identifies the face in the images it was required to select the approach for detecting the eyes. For this stage it was decided to put aside the use of Haar Cascade classifiers and better to use the deep learning approach. It is possible to test a model trained by Sipeed that it is available on their website [83]. This model requires as an input an image of 64x64 pixels in RGB color and as output provides the coordinates of upper and lower edges of the eyes. One drawback of this model is the size since it requires at least 2739 KB of general-purpose memory to be used, this led to a memory error when using the full version of the firmware, since when the system tries to load the model there is only 808 KB of free general-purpose memory.

An attempt to try to overcome this issue was made by removing from memory the face detection model, before loading the eye detection model. Even though this release memory it still not enough memory for using the model. Therefore, it was decided to try other firmware versions. Table 4.4 presents the amount of memory free before loading the model to the general-purpose memory under two combinations loading the models at the beginning of the execution and removing from memory each model after the task is completed. Even though removing the previous model from the general purpose-memory it was only possible to get the required memory with the firmware built from source. The table does not present results for the minimum firmware with IDE support because this firmware only supports kmodel V3 and the models used are kmodel V4.



Table 4.4 Memory management

<b>Firmware</b>	<b>Memory before loading the model at the beginning</b>	<b>Memory before loading the model after removing previous model.</b>
Normal firmware	808 KB	1252 KB
Minimum firmware with IDE support	N/A	N/A
Minimum firmware with kmodel V4 support	3008 KB	3020 KB
OpenMV kmodel V4 with IDE	2172 KB	2628 KB
Custom firmware	2596 KB	3052 KB

Even though with the custom firmware and minimum firmware with V4 support it was possible to overcome the memory issued, and the model provides the results as it can be seen in Figure 4.5. Another drawback was presented by using this model, and it's related to frame rate. The fact of being loading and removing from memory each model after it's task was completed decrease the frame rate in a considerable way at the point it's not suitable for real time applications.



Figure 4.5 Sipeed model for eye detection.

Since the provided model was not suitable for a realistic implementation it was decided to train a custom model. To train the custom model the following dataset was used, [84] it is compounded by more than 7000 images labeled with the coordinates of 15

facial landmarks. However, only the landmarks related to the eye were used for this task. In this sense the dataset provides coordinates of the eye center, inner and outer corner. It was noticed that around of 50% of the samples do not have information about the eye corners, therefore it was decided to select only the center of the eyes to have a better dataset.

#### 4.4.1 Models' details

To build models that could run on the target platform the constrains mention in [85] had to be met. According to the neural network structure the number of parameters and the weight of the model changes. Figure 4.6 shows the network structure of the first model trained. All the hidden layers used ReLU as activation function while, at the output layers used Sigmoid as activation function. The input of the network is a 96x96 grayscale image and provides as an output the X, Y coordinates of the center of the eye.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 96, 96, 32)	320
max_pooling2d (MaxPooling2D)	(None, 48, 48, 32)	0
conv2d_1 (Conv2D)	(None, 48, 48, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
flatten (Flatten)	(None, 36864)	0
dense (Dense)	(None, 32)	1179680
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132
Total params: 1,198,628		
Trainable params: 1,198,628		
Non-trainable params: 0		

Figure 4.6 Model version 1 network structure.

In order to be able to run this model in the board it's required to convert the model to the format used by the board (.kmodel) using the neural network compiler (NNCase) [86]. To achieve this, it is necessary to convert the Keras (h5) file into TensorFlow (tf lite) file and then use the NNCase tool to convert the TensorFlow file into Kendryte

file (kmodel). By doing this the size of the original file is decreased, despite of this size reduction the first version gave a model of size 4,7 MB.

According to the literature [87], [88] , it's is possible to reduce the size of the model pruning it. This will create a model 3 times smaller than the original one. Also, it is possible to get a model 10 times smaller by combining pruning and quantizing the model. Both approaches where tested and the results are presented in Table 4.5. As it can be seen pruning the model affects the size of the original model, but it does not have any effect on the converted and final model. On the other hand, adding quantization to the pruning has a noticeable impact on the converted model but by doing this it's not possible to get the final model since currently NNCase doesn't support import a quantized graph [89].

Table 4.5 Model size

<b>Model</b>	<b>Normal</b>	<b>Pruned</b>	<b>Pruned and quantized</b>
Original (h5)	14,4 MB	14,4 MB	14,4
Pruned (h5)	N/A	4,8 MB	N/A
Pruned & quantized (tflite)	N/A	N/A	1,2MB
Converted (tflite)	4,8 MB	4,8MB	1,2MB
Final (kmodel)	4,7 MB	4,7MB	N/A

Figure 4.7 presents the network structure of the second version. The main changes made to the new structure compared to the previous version was the replacement of the activation function in the output layer with a ReLU function and the reduction of the filters per layer in 50%. Those modifications gave a final model of size 2,4 MB despite there is a decrease on the size of the model, it is still not suitable to overcome the memory and frame rate issues. Therefore, it was decided to use a different structure.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 96, 96, 16)	160
max_pooling2d (MaxPooling2D)	(None, 48, 48, 16)	0
conv2d_1 (Conv2D)	(None, 48, 48, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 32)	0
dropout (Dropout)	(None, 24, 24, 32)	0
flatten (Flatten)	(None, 18432)	0
dense (Dense)	(None, 32)	589856
dropout_1 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 4)	132
Total params: 594,788		
Trainable params: 594,788		
Non-trainable params: 0		

Figure 4.7 Model version 2 network structure.

A third network structure was proposed reducing again the filters per layer in 50% compared to the values of the previous models, also a new layer was added to the structure. Figure 4.8 presents the new structure. The new modifications gave a final model of size 302 kB. This size allows to overcome the memory and frame rate issues presents with the previous models. This was possible since the new model can be loaded just once into the general-purpose memory of the board at the beginning of the execution of the program. However, one important thing to point as can be seen in the corner of the images in Figure 4.9 and Figure 4.10 is the fact that the frame rate when both task are executed decreased compared to the frame rate obtained by only playing the video on the board.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 96, 96, 16)	160
max_pooling2d (MaxPooling2D)	(None, 48, 48, 16)	0
conv2d_1 (Conv2D)	(None, 48, 48, 24)	3480
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 24)	0
conv2d_2 (Conv2D)	(None, 24, 24, 32)	6944
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 32)	0
dropout (Dropout)	(None, 12, 12, 32)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 16)	73744
dropout_1 (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 4)	68
Total params: 84,396		
Trainable params: 84,396		
Non-trainable params: 0		

Figure 4.8 Model version 3 network structure.

As it is mentioned before the output of the model are the X and Y coordinates of the center of the eye. However, the main purpose of adding this model to the system is to get an image of the whole eye to feed another model that can predict the status of the eye. Therefore, it was required to perform a calibration to the values by adding some offsets in both coordinates to get a rectangle that surrounds the eye. Figure 4.9 shows the results after calibration under daylight conditions, as it can be seen the model is capable to detect the region of the eyes even if the driver is using glasses.



Figure 4.9 Model results on daytime conditions, with glasses (left) and without glasses (right)

On the other hand, under night conditions the model also could track the region where the eyes are located even if the driver is using glasses as can be seen in Figure 4.10. The performance result of this model on the other videos of the dataset can be seen in Table 4.6, these results are based on 125 random frames from each video, leading to an average performance value of 82,6%.



Figure 4.10 Model results on night conditions, without glasses (left) and with glasses (right).

Table 4.6 Eye detection performance

Subject	Video			Correctly detected eyes	Wrongly detected eyes	Accuracy
	Time	Glasses	Condition			
1	Day	Yes	NSL	98	27	78,4%
1	Day	No	NSL	97	28	77,6%
1	Day	No	SL	4	121	96,8%
4	Night	Yes	MLC	108	17	86,4%
8	Night	No	NSL	116	9	92,8%
10	Day	No	MLC	93	32	74,4%
10	Night	No	MLC	117	8	93,6%
10	Night	Yes	MLC	120	5	96,0%
11	Day	No	MLC	105	20	84,0%
21	Day	Yes	MLC	98	27	78,4%
23	Night	Yes	SL	115	10	90,7%
34	Night	Yes	SL	120	5	96,0%
34	Day	No	NSL	84	41	67,2%
37	Day	No	MLC	125	0	100.0%

Even though the performance was acceptable there were two common drawbacks in the model that was noticeable in all the videos, both related to the head orientation. As it can be seen from the images in Figure 4.11, if the driver is looking to the left or right side the model is not capable of providing the correct region where the eyes are located. The same behaviour is noticeable if the driver nods at a very sharp angle. This drawback could be mitigated using a dataset that contains images in the same conditions, creating a more robust architecture of the network but this will require more memory. Or by adding an intermediate layer that predicts the orientation of the face before the system attempts to locate the eyes. The last option was selected to mitigate the drawback. The elaboration of this layer as well as the results obtained by it are detailed in the following section.



Figure 4.11 Frames with wrong eye localization

## 4.5 Face orientation

### 4.5.1 Key points comparison

According to the documentation, the OpenMV firmware contains a function that obtains the ORB key points of an image, these points can be used to calculate how different are two images. This function was used to build the workflow presented in Figure 4.12, to calculate how different was the current face pose with respect to the first frame face pose and based on the result localize or not the eyes.

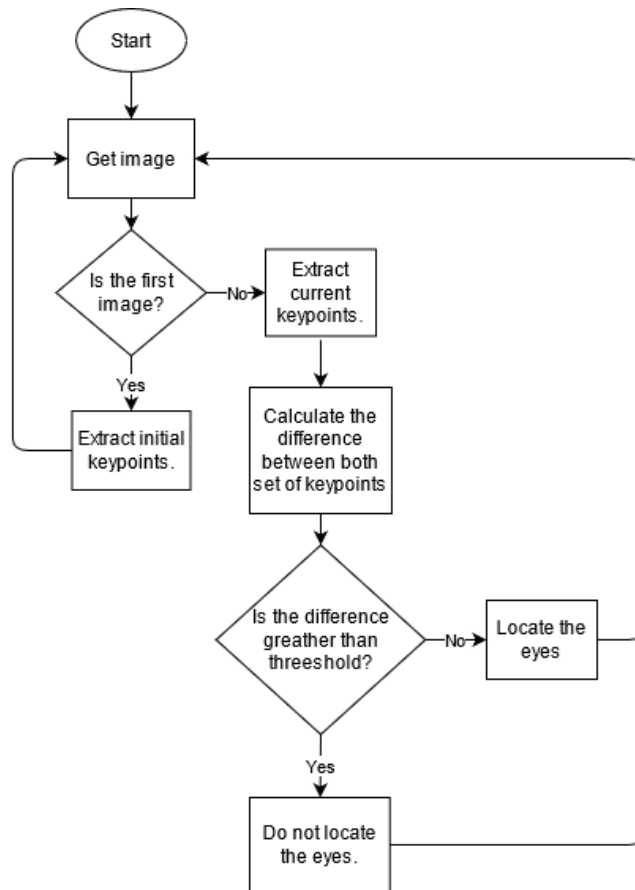


Figure 4.12 Key points comparison workflow

As it can be seen from Figure 4.13 there is a noticeable difference in the original key points (white points) and the current ones (pink points), when the face is oriented to the side compared when the face is looking to the front. Even though the results obtained by the comparisons can improve the performance of the localization of the eyes. It was decided not to use this method for the following reasons: the system does not always manage to obtain key points, leading to a sequence of frames where the calculations are not done with the correct set of key points. On the other hand, the FPS is affected by a decrease of more than 50% with respect to the previous stage.



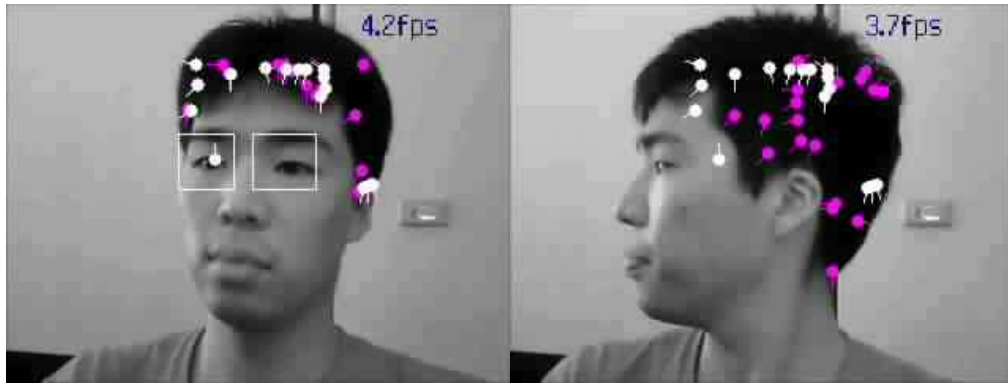


Figure 4.13 Key points difference, correct orientation (left side) and wrong orientation (right side).

### 4.5.2 Neural Network approach

The second iteration for improve the eye localization stage, was based on create and train a CNN model. For training the face orientation model the FEI face database was used [90]. The dataset contains 14 pictures per individual looking into different points as it can be seen in Figure 4.14



Figure 4.14 FEI database example. [90]

The main goal of this model was to classify the orientation of the face. To achieve this only the first 12 pictures per individual were used for making the model except for the picture 6, because it led the model into high-rate miss classification when the drivers were looking to the right side. The pictures were divided into 5 classes representing the orientation of the head. A first experiment was carried out using the online training tool presented by Sipeed, which uses transfer learning using the MobileNet V1 architecture. However, the result was an 1838 KB model [91]. Another experiment was carried out using transfer learning with the MobileNetV2 and Resnet50 architectures, but the result was the same a model with a size that was not suitable to run into the board. Therefore, it was decided to create a custom architecture.

The architecture of the network used for making the model can be seen in Figure 4.15. During the training phase this model provided an accuracy of 79,2%. By introducing this model to the system, the performance of the wrong eye localization was improved slightly. However, there was still a considerable number of frames where the region was not detected correctly. To improve its performance, it was decided to reduce the number of classes to two.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 36, 36, 16)	160
max_pooling2d (MaxPooling2D)	(None, 18, 18, 16)	0
conv2d_1 (Conv2D)	(None, 18, 18, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 9, 9, 32)	0
conv2d_2 (Conv2D)	(None, 9, 9, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 64)	0
dropout (Dropout)	(None, 4, 4, 64)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 64)	65600
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 2)	130
Total params: 89,026		
Trainable params: 89,026		
Non-trainable params: 0		

Figure 4.15 Face orientation model architecture

By reducing the number of classes to two the accuracy during the training phase improves, giving a result of 92,1%. Figure 4.16 shows how images were used for each class. Upper images were considered as wrong positions for eyes localization. Therefore, every time that the new model classifies the head position as class 1 the system does not attempt to localize the region of the eyes. The input of the model is a 36 x 36 pixels image in grayscale. By using this configuration, it was possible to obtain a final model of 94,1 KB, a suitable model size as this can also be loaded into the general-purpose memory once at the beginning of the execution of the program without having memory issues.



Figure 4.16 Face orientation classes, wrong orientation(up), correct orientation(down) [90]

As it can be seen from Table 4.7 the average performance of the eye localization model improved to 92,1% by adding the face orientation layer as a step before the system attempts to locate the eyes. Despite the improvement, there are still frames where the eyes are wrongly detected, mostly when the driver head is tilted to the front or frames where the model was capable to predict the location of the eyes correctly, but the output of the model identifies the face as a wrong position as it can be seen in Figure 4.17. This is mainly caused because the dataset used does not contain images under these conditions therefore the model is not capable to provide an accurate prediction.

Table 4.7 Eye detection performance with new model

Subject	Video			Correctly detected eyes	Wrongly detected eyes	Accuracy
	Time	Glasses	Condition			
1	Day	Yes	NSL	115	10	92,0%
1	Day	No	NSL	114	11	91,2%
1	Day	No	SL	125	0	100,0%
4	Night	Yes	MLC	122	3	97,6%
8	Night	No	NSL	121	4	96,8%
10	Day	No	MLC	113	12	89,6%
10	Night	No	MLC	71	54	56,8%
10	Night	Yes	MLC	115	10	92,1%
11	Day	No	MLC	114	11	91,2%
21	Day	Yes	MLC	123	2	98,4%
23	Night	Yes	SL	121	4	96,8%
34	Night	Yes	SL	111	14	88,8%
34	Day	No	NSL	122	3	97,6%
37	Day	No	MLC	125	0	100%

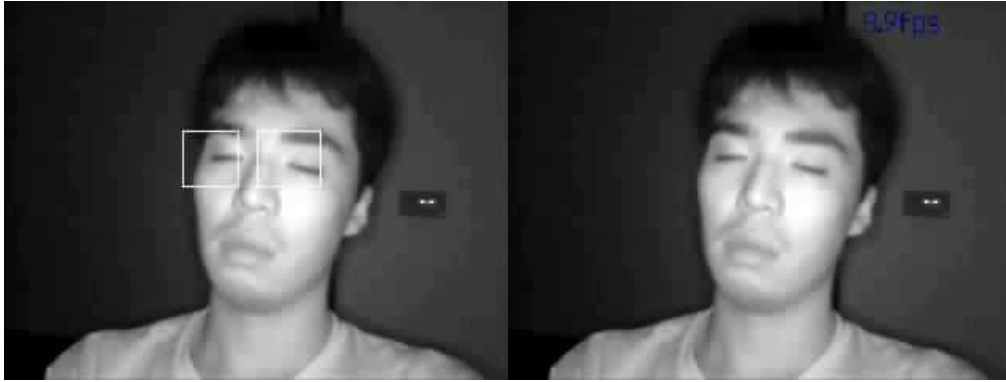


Figure 4.17 Frame with correct eye localization without head orientation model (left), with head orientation model (right).

Figure 4.18 presents the same frames as the ones presented in Figure 4.11, as it can be seen adding the face orientation layer using this approach as an intermediate layer between the face detection and eye localization improves the performance removing in a considerable number the amount of frames where the eye localization model provided wrong coordinates. Despite the good results, it is important to point that the FPS decrease compared to the ones obtained when the system used only the face detection and eye localization models. This frame rate decrease is caused because the KPU requires the images in RGB888 format for using them in the models. Therefore, it is required to convert the image to RGB565. According to the documentation [88] the instruction that performs this conversion consumes time that is added to the program execution, leading to the decrease in the frame rate. However, the decrease of FPS that this approach presents is lower compared to the one in the previous section. Therefore, it was decided to use this approach for the head orientation layer in the system.



Figure 4.18 Eye localization with head orientation model.

## 4.6 Eye status

### 4.6.1 Color detection

The next layer added to the system would have as main goal to determine the status of the eye, to analyze the amount of time the eye remain closed to determine if the driver is drowsy. Three approaches were tested for this layer. The first approach was through color detection, which consisted in trying to determine the state of the eye by detecting the black color pixels in the image of the cropped eyes, as can be seen from Figure 4.19, using the IDE threshold editor it's is possible to calibrate a threshold to identify a key difference in the pixels between the states of the eye.

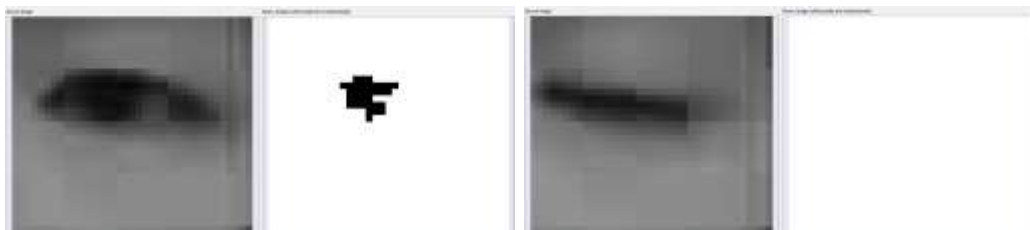


Figure 4.19 Binary image of open eye(left) and closed eye(right)

Table 4.8 presents the validation results of this approach using 150 frames as samples. The average accuracy of this approach was 51,6% as it can be seen from the table, the performance of this approach is significantly degraded when conditions are nocturnal, and the driver has more episodes of sleepiness.

Table 4.8 Layer accuracy

Subject	Video			Accuracy
	Time	Glasses	Condition	
1	Day	Yes	NSL	89,5%
1	Day	No	NSL	89,7%
1	Day	Yes	SL	57,3%
4	Night	Yes	MLC	2,2%
8	Night	No	NSL	4,3%
10	Day	No	MLC	63,0%
10	Night	No	MLC	3,6%
10	Night	Yes	MLC	2,8%
11	Day	No	MLC	85,6%
21	Day	Yes	MLC	91,5%
23	Night	Yes	SL	55,7%
34	Night	Yes	SL	66,7%
34	Day	No	NSL	84,9%
37	Day	No	MLC	25,4%

On the other hand, using this method did not affect the frame rate in a considerable way as it can be seen in the corner of the images presented below. Despite this the low value of the average accuracy obtained and that its performance is affected when drivers tend to have more drowsy episodes this technique was ruled out.

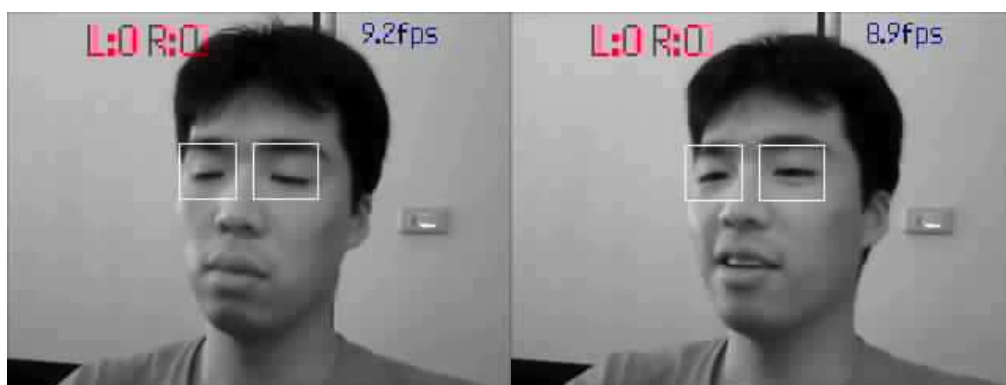


Figure 4.20 Eye status prediction using color detection.

### 4.6.2 Pupil detection

The second approach consisted in pupil tracking, according to the documentation, some versions of the firmware contain a method capable of detecting the position of the pupil in a ROI. It was intended to use this method to determine the state of the eye, since when the eye is closed there is absence of the pupil and according to the documentation the returned value should be (0,0) [92]. However, as can be seen in Figure 4.21, despite in the images of the right side the driver has both eyes closed the method identifies the presence of the pupil inside the ROI determined by the white rectangle.

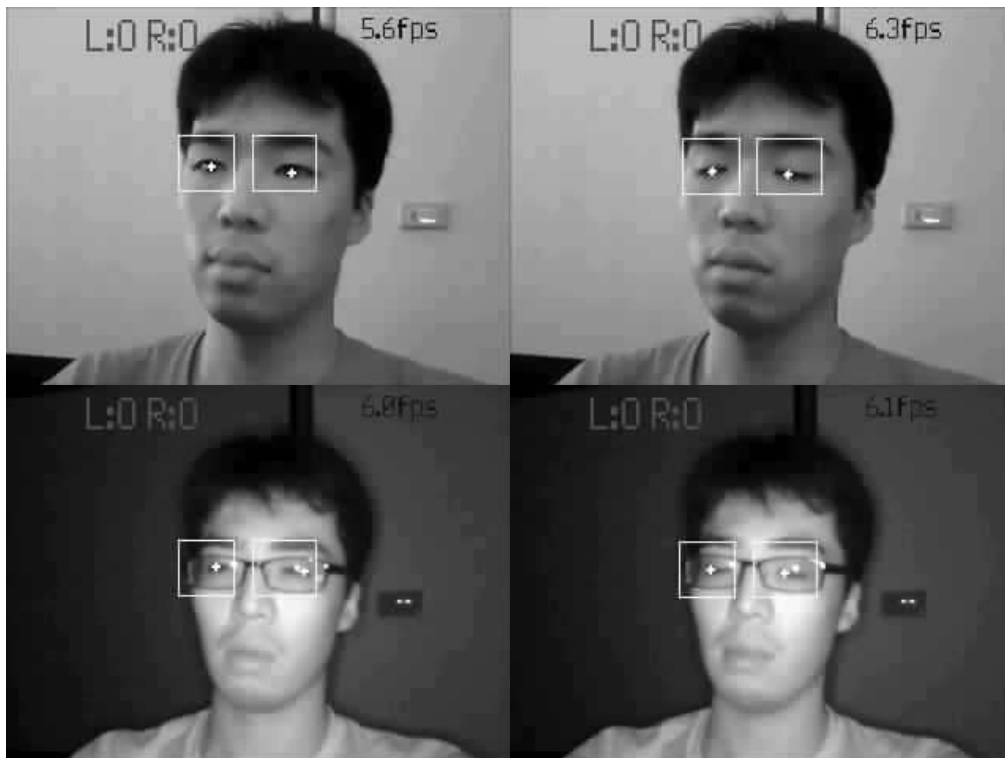


Figure 4.21 Pupil detection method open eye (left), closed eye (right)

Another major drawback that was noticeable in this method was present during night conditions as is can be seen in the images below, the method provides wrong coordinates of where it detects that the pupil is located. On the other hand, it was also noticeable that the FPS decreased to around 6FPS using this approach. Due to these results, this technique was ruled out.

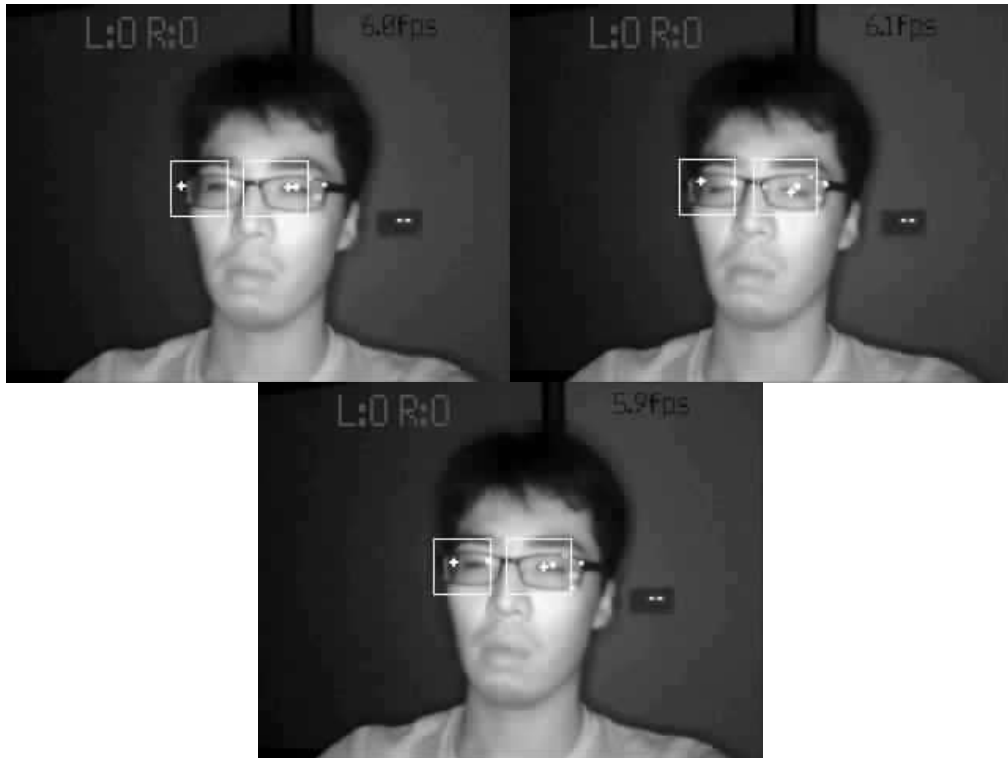


Figure 4.22 Wrong pupil detection.

### 4.6.3 Neural Network classifier

The last approach consisted in training a binary classification model to determine the status of the eye. Since this stage is considered the core of the system different architectures were proposed, for training the models the dataset presented in [93] was used. The dataset is divided in four classes. However, it was decided to merge the left and right open status in one class and the left and right closed status in the second class. Graphs in Figure 4.23 presents the validation accuracy and loss of the proposed architectures as it can be seen the red graph (architecture 5) presents the best results.



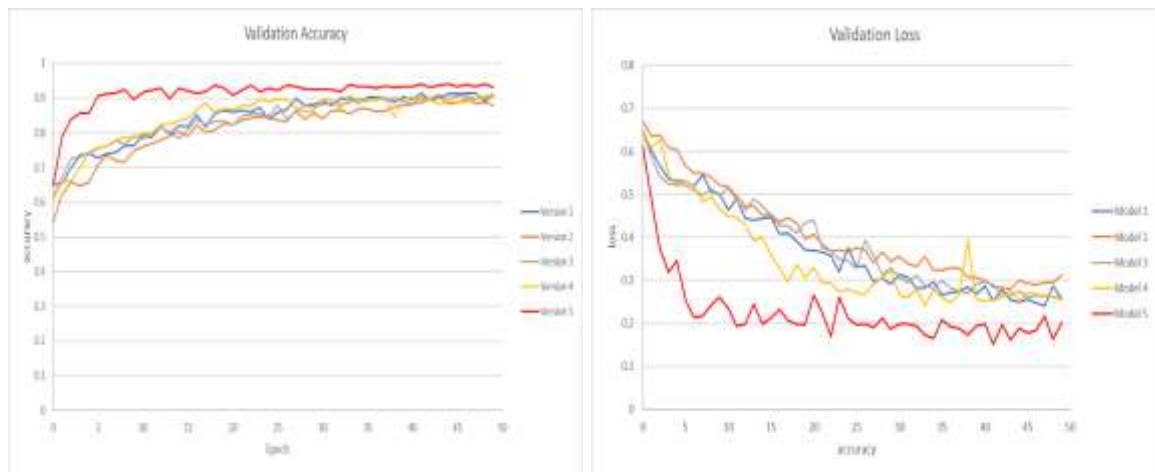


Figure 4.23 Different architectures performance, validation accuracy (left), validation loss (right)

Figure 4.24 shows the architecture that presents better training results, despite been the architecture with more convolutional layers, number of nodes per layer and with the highest size of dense layer, the total size of the trained model after its conversion to the Kmodel was of 296.6 kB. Since this approach allows to use the customized firmware, the size of the model fits perfect with the memory available in the device after performing the tasks in other layers of the system. With the mentioned size it is possible to load the model once at the beginning of the program execution, avoiding a considerable FPS reduction.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 26, 34, 1)]	0
conv2d (Conv2D)	(None, 26, 34, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 17, 32)	0
conv2d_1 (Conv2D)	(None, 13, 17, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 8, 64)	0
conv2d_2 (Conv2D)	(None, 6, 8, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 3, 4, 128)	0
flatten (Flatten)	(None, 1536)	0
dense (Dense)	(None, 32)	49184
activation (Activation)	(None, 32)	0
dense_1 (Dense)	(None, 2)	66
activation_1 (Activation)	(None, 2)	0
Total params: 141,922		
Trainable params: 141,922		
Non-trainable params: 0		

Figure 4.24 Eye status model architecture.

Despite the good accuracy during the training phase (94.1%) the model presented difficulties as it can be seen in Table 4.9, where the performance of the model under the same 150 frames used for testing the color detection model are presented. With this approach it was possible to get a performance 5% greater (56.1 %) than the one obtained with the color detection model. Although there was improvement, the performance the purposed model was not optimal to add the layer to the system, in order to improve the performance, it was decided to use a new dataset [94] with more images for training. By replacing the dataset, it was possible to obtain a performance increase of 20% (76,9%) with respect to the previous dataset, it still shows a low performance. However, as can be seen in the table, it is in specific cases when the driver's conduction is at night when the performance is affected considerably. Based on these results it was decided to use some videos from the NHTU dataset with similar conditions to crop the eyes and use them in combination with the MRL dataset [95] Adding some sample images to the MRL dataset improve the average accuracy of the model up to 79,9%. Therefore, it was decided to use the model obtained by combining both models for design the drowsiness detection system.

Table 4.9 Model performance

Subject	Video			Accuracy CEW data	Accuracy MRL data	Accuracy MRL with NHTU data
	Time	Glasses	Condition			
1	Day	Yes	NSL	74,0%	88,0%	89,3%
1	Day	No	NSL	93,4%	95,9%	94,2%
1	Day	Yes	SL	77,6%	86,5%	91,6%
4	Night	Yes	MLC	26,7%	44,6%	74,1%
8	Night	No	NSL	19,3%	93,6%	95,7%
10	Day	No	MLC	67,7%	82,3%	78,0%
10	Night	No	MLC	51,8%	78,9%	84,3%
10	Night	Yes	MLC	69,7%	57,6%	74,6%
11	Day	No	MLC	37,3%	89,4%	83,2%
21	Day	Yes	MLC	45,0%	96,6%	93,2%
23	Night	Yes	SL	39,9%	54,2%	45,7%
34	Night	Yes	SL	61,6%	82,6%	70,9%
34	Day	No	NSL	75,8%	89,3%	62,2%
37	Day	No	MLC	45,0%	37,1%	82,1%

## 4.7 Analysis and results

Figure 4.25 illustrate some tested performed in real time. As it can be seen the system has a frame rate of 19-20 FPS when the conditions in the workflow enable the drowsy detection layer. On the other hand, when this layer is not enabled the frame rate is greater in a range between 25-26 FPS.

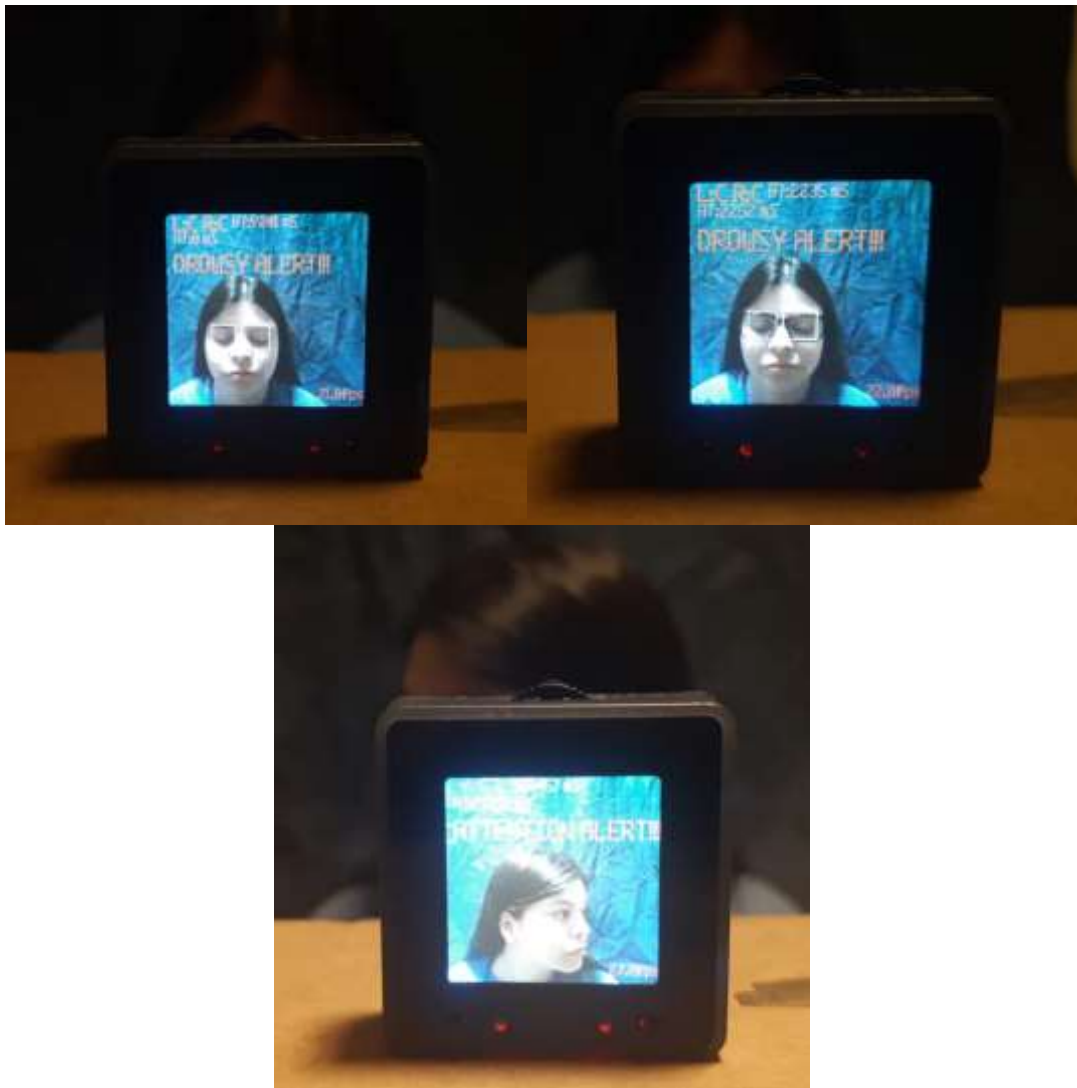


Figure 4.25 System results.

Despite the good results obtained for the frame rate, the accuracy results of the system are not optimal. As it can be seen from Table 4.10 the attention system presented an accuracy percentage higher than 80% in almost all the videos tested, resulting in an average accuracy of 85,6%, with an average F1 score of 0,87. Something important to point is the fact that it was not possible to get the accuracy for most of videos labeled under MLC condition since the dataset used does not provide the files that describe in how many frames the drivers head is oriented to the sides. On the other hand, as it can

be seen the performance of the drowsy system had a greater range of accuracy oscillation leading to an average value of 67,9%, with an average F1 score of 0,64.

Table 4.10 Detection system results.

Video				Drowsiness		Attention	
Subject	Time	Glasses	Condition	Accuracy	F1 Score	Accuracy	F1 Score
1	Day	Yes	NSL	86,6%	0,92	86,9%	0,69
1	Day	No	NSL	93,9%	0,96	78,6%	0,85
1	Day	Yes	SL	45,3%	0,23	70,0%	0,79
4	Night	Yes	MLC	26,1%	0,21	85,0%	0,91
8	Night	No	NSL	90,7%	0,95	98,8%	0,99
10	Day	No	MLC	74,1%	0,85	N/D	N/D
10	Night	No	MLC	22,3%	0,33	N/D	N/D
10	Night	Yes	MLC	69,4%	0,80	N/D	N/D
11	Day	No	MLC	95,5%	0,97	N/D	N/D
21	Day	Yes	MLC	98,1%	0,99	N/D	N/D
23	Night	Yes	SL	89,7%	0,61	93,0%	0,96
34	Night	Yes	SL	21,0%	0,23	84,6%	0,91
34	Day	No	NSL	81,1%	0,67	87,7%	0,89
37	Day	No	MLC	56,5%	0,25	N/D	N/D

The low performance of the drowsy detection system is related to the false positives generated by the attention system, as it can be seen from Figure 4.26 where a sequence of frames where the driver is drowsy are presented but the system infer that the face is not in a suitable position giving a wrong alert, this issue is also replicated when the drivers face is tilted due to the nodding produced by drowsy state. This issue may be overcome by using a dataset with a wide range of head orientation images. Another way to improve the system performance could be by modifying the structure of the networks used for training or using common architectures as Resnet50 or MobilenetV2 for transfer learning. However, both of the approaches lead to an increase of the model size. The model size increase may lead to the memory issues mentioned previously, where it is not possible to load all the models to the memory only once at the beginning of the program execution, this could be overcome loading and removing each model from memory after every inference. However, as the experiments show this cause a considerable decrease in the frame rate.



Figure 4.26 Wrong alert classification.

## **5. SUMMARY**

Current work describes software developments of vehicle driver drowsiness detection based on the face recognition. The system proposed in this thesis can process the images at a frame rate of between 19-25 FPS which allows it to infer the alert, distracted and drowsy status of the drivers in real time. The system is configured to run on a low-cost embedded CV platform. From the different software layers that compound the system the most robust layer is the face detection layer, which presents the highest accuracy (98,2%) value of all the layers. The fact that this layer can detect the face even when it is tilted frontally or laterally presented an issue to the eye localization layer, since it is not capable to provide the correct location of the eyes when the face is in the previously mentioned positions. This issue could be overcome by training different model architectures but the constrain about the platform memory limited the size of the model. Therefore, the solution to improve the results (head orientation layer) was capable to improve the wrong localization results. However, this affects the eye status prediction layer since the dataset used does not contain a wide range of head orientation images, leading into a low average accuracy value for drowsiness detection (67,9%) compared to the solutions presented in section 2.3. Based on the performance of the system it can be concluded that the Maix Cube is not suitable for the developed software solution, mainly because of its memory limitation. To get better results, it is necessary to improve the architecture of the proposed networks by increasing the number of parameters which will lead to an increase in the size of the model for doing this it is required at least 32 MiB of memory dedicated just to the inference of the models, to perform more tests.

### **5.1 Future work**

To improve the results obtained by the system proposed in this system, it is recommended to use another computer vision board with at least 32 MiB of memory dedicated to inference of the models. As second recommendation for the new development it is important to point try another approach for classifying the head orientation like perspective-n-point algorithm or merge the face detection and face orientation into the object detection layer or use a dataset that contains a wide range of images of head orientation.

## 6. KOKKUVÕTE

Käesolev töö kirjeldab tarkvaraarendusi sõidukijuhi unisuse määramisel näotuvastuse abil. Selles lõputöös välja pakutud süsteem suudab pilte töödelda kaadrisagedusega vahemikus 19 kuni 25 kaadrit sekundis, mis võimaldab reaajas hinnata juhtide erksat, häiritud ja uimast olekut. Süsteem on konfigureeritud töötama soodsal masinnägemisplatvormil. Süsteemi erinevatest tarkvarakihtidest töökindlaim on näotuvastuskiht, mis näitab kõigist kihtidest suurimat täpsust (98,2%). Asjaolu, et see tarkvarakiht suudab tuvastada nägu ka siis, kui see on ees- või külgsuunas kallutatud, tekitab silmade lokaliseerimise kihile probleemi, kuna viimane ei suuda tuvastada silmade õiget asukohta, kui nägu on eelnevalt mainitud asendites. Selle probleemi võiks lahendada teistsuguste mudeliarhitektuuride abil, end seatud platvormipiirang seadis piirid ka kasutatavale mälumahule. Seega aitas tulemusi parandada pea suuna määramise kiht, mis korrigeeris valesid lokaliseerimistulemusi. See mõjutab aga silma seisundi määramise kihti, kuna kasutatav andmekogum ei sisalda laias valikus näidispilte pea suuna tuvastamiseks, mille tulemuseks on omakorda madal täpsus uimasuse tuvastamisel (67,9%) võrreldes punktis 2.3 toodud lahendustega. Süsteemi jõudluse põhjal võib järeldada, et Maix Cube ei sobi väljatöötatud tarkvaralahenduse jaoks peamiselt mälupiirangu tõttu. Paremate tulemuste saavutamiseks on vaja pakutud võrkude arhitektuuri täiustada, suurendades parameetrite arvu, mis suurendab mudeli suurust. Järgmiste katsetuste läbiviimiseks on tarvis vaid mudelist järelduste tegemiseks ette näha vähemalt 32 MiB mälu.

### 6.1 Järgmised sammud

Parendamaks pakutud süsteemi tulemusi, on soovitatav kasutada mõnda muud tehisnägemisplatvormi, millel on vähemalt 32 MiB mälu mudelitest järelduste tegemiseks. Teine soovitus edasiarenduseks oleks proovida teist lähenemist pea suuna määramiseks, näiteks ankurpunkte ruumis vaadeldes (perspective-n-point algorithms), ühendades näo tuvastamine ning näo suuna tuvastamine objektituvastuskihis, või kasutades andmekogu, mis sisaldab suurt hulka pilte näitlikustamiseks pea suunda.

## LIST OF REFERENCES

- [1] World Health Organization, "Road Safety." <https://www.who.int/data/gho/data/themes/road-safety> (accessed Nov. 09, 2020).
- [2] T. Akerstedt *et al.*, "Sleepiness at the Wheel," 2015. Accessed: Nov. 06, 2020. [Online]. Available: [https://esrs.eu/wp-content/uploads/2018/09/Livre\\_blanc\\_VA\\_V4.pdf](https://esrs.eu/wp-content/uploads/2018/09/Livre_blanc_VA_V4.pdf).
- [3] Mobility and Transport, "Road safety: Europe's roads are getting safer but progress remains too slow ," Jun. 11, 2020. [https://ec.europa.eu/transport/media/news/2020-06-11-road-safety-statistics-2019\\_en](https://ec.europa.eu/transport/media/news/2020-06-11-road-safety-statistics-2019_en) (accessed Nov. 09, 2020).
- [4] European Council, "EU beefs up requirements for car safety ," Mar. 29, 2019. <https://www.consilium.europa.eu/en/press/press-releases/2019/03/29/eu-beefs-up-requirements-for-car-safety/> (accessed Nov. 09, 2020).
- [5] European Parliament, "Safer roads: new EU measures to reduce car accidents ," Apr. 16, 2019. <https://www.europarl.europa.eu/news/en/headlines/society/20190307STO30715/safer-roads-new-eu-measures-to-reduce-car-accidents> (accessed Nov. 09, 2020).
- [6] L. Oliveira, J. S. Cardoso, A. Lourenço, and C. Ahlström, "Driver drowsiness detection: A comparison between intrusive and non-intrusive signal acquisition methods," in *Proceedings - European Workshop on Visual Information Processing, EUVIP*, Jan. 2019, vol. 2018-November, doi: 10.1109/EUVIP.2018.8611704.
- [7] European Commission, "Road safety: Commission welcomes agreement on new EU rules to help save lives." [https://ec.europa.eu/commission/presscorner/detail/en/IP\\_19\\_1793](https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1793) (accessed Nov. 16, 2020).
- [8] European Parliament, "Regulation (EU) 2019/... of the European Parliament and of the Council," 2019. Accessed: Nov. 09, 2020. [Online]. Available: [https://www.europarl.europa.eu/meetdocs/2014\\_2019/plmrep/COMMITTEES/IMCO/DV/2019/04-01/COREPER\\_ANNEX\\_GVSR\\_EN.pdf](https://www.europarl.europa.eu/meetdocs/2014_2019/plmrep/COMMITTEES/IMCO/DV/2019/04-01/COREPER_ANNEX_GVSR_EN.pdf).
- [9] S.-W. Jang and B. Ahn, "Implementation of Detection System for Drowsy Driving Prevention Using Image Recognition and IoT," *Sustainability*, vol. 12, no. 7, p. 3037, Apr. 2020, doi: 10.3390/su12073037.
- [10] J. Solaz *et al.*, "Drowsiness Detection Based on the Analysis of Breathing Rate Obtained from Real-time Image Recognition," in *Transportation Research Procedia*, Jan. 2016, vol. 14, pp. 3867–3876, doi: 10.1016/j.trpro.2016.05.472.



- [11] J. E. Mabry, T. Laurel, G. Jeffrey, and S. Hickman, "Commercial Motor Vehicle Operator Fatigue Detection Technology Catalog and Review." Accessed: Oct. 23, 2020. [Online]. Available: [https://vtechworks.lib.vt.edu/bitstream/handle/10919/87743/NSTSCE\\_FatigueDetectionCatalog\\_Final.pdf?sequence=1&isAllowed=y](https://vtechworks.lib.vt.edu/bitstream/handle/10919/87743/NSTSCE_FatigueDetectionCatalog_Final.pdf?sequence=1&isAllowed=y).
- [12] M. Awais, N. Badruddin, and M. Drieberg, "A hybrid approach to detect driver drowsiness utilizing physiological signals to improve system performance and Wearability," *Sensors (Switzerland)*, vol. 17, no. 9, Sep. 2017, doi: 10.3390/s17091991.
- [13] M. Doudou, A. Bouabdallah, and V. Berge-Cherfaoui, "Driver Drowsiness Measurement Technologies: Current Research, Market Solutions, and Challenges," *Int. J. Intell. Transp. Syst. Res.*, vol. 18, no. 2, pp. 297–319, May 2020, doi: 10.1007/s13177-019-00199-w.
- [14] F. Guede-Fernández, M. Fernández-Chimeno, J. Ramos-Castro, and M. A. García-González, "Driver Drowsiness Detection Based on Respiratory Signal Analysis," *IEEE Access*, vol. 7, pp. 81826–81838, 2019, doi: 10.1109/ACCESS.2019.2924481.
- [15] B. Warwick, N. Symons, X. Chen, and K. Xiong, "Detecting driver drowsiness using wireless wearables," in *Proceedings - 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2015*, Dec. 2015, pp. 585–588, doi: 10.1109/MASS.2015.22.
- [16] S. E. H. Kiashari, A. Nahvi, H. Bakhoda, A. Homayounfard, and M. Tashakori, "Evaluation of driver drowsiness using respiration analysis by thermal imaging on a driving simulator," *Multimed. Tools Appl.*, vol. 79, no. 25–26, pp. 17793–17815, Jul. 2020, doi: 10.1007/s11042-020-08696-x.
- [17] G. Li, B. L. Lee, and W. Y. Chung, "Smartwatch-Based Wearable EEG System for Driver Drowsiness Detection," *IEEE Sens. J.*, vol. 15, no. 12, pp. 7169–7180, Dec. 2015, doi: 10.1109/JSEN.2015.2473679.
- [18] T. Kunding, N. Sofra, and A. Riener, "Assessment of the potential of wrist-worn wearable sensors for driver drowsiness detection," *Sensors (Switzerland)*, vol. 20, no. 4, Feb. 2020, doi: 10.3390/s20041029.
- [19] StopSleep, "StopSleep User Guide." <http://www.stopsleep.com.au/wp-content/uploads/2015/05/StopSleep-Instruction-Manual-AUS-NZ-v1.02.pdf> (accessed Oct. 22, 2020).
- [20] SmartCap Technologies, "Life." <http://www.smartcaptech.com/life-smart-cap/> (accessed Nov. 05, 2020).
- [21] CardioID, "CardioWheel." <https://www.cardio-id.com/cardiowheel> (accessed Nov. 05, 2020).

- [22] A. Lourenço, A. P. Alves, C. Carreiras, R. P. Duarte, and A. Fred, "CardioWheel: ECG biometrics on the steering wheel," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9286, pp. 267–270, doi: 10.1007/978-3-319-23461-8\_27.
- [23] M. Chai, S. W. Li, W. C. Sun, M. Z. Guo, and M. Y. Huang, "Drowsiness monitoring based on steering wheel status," *Transp. Res. Part D Transp. Environ.*, vol. 66, Jan. 2019, doi: 10.1016/j.trd.2018.07.007.
- [24] S. Arefnezhad, S. Samiee, A. Eichberger, and A. Nahvi, "Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection," *Sensors*, vol. 19, no. 4, Feb. 2019, doi: 10.3390/s19040943.
- [25] B. L. Lee, B. G. Lee, and W. Y. Chung, "Standalone Wearable Driver Drowsiness Detection System in a Smartwatch," *IEEE Sens. J.*, vol. 16, no. 13, Jul. 2016, doi: 10.1109/JSEN.2016.2566667.
- [26] M. S. Wang *et al.*, "Drowsy behavior detection based on driving information," *Int. J. Automat. Technol.*, vol. 17, no. 1, pp. 165–173, Feb. 2016, doi: 10.1007/s12239-016-0016-y.
- [27] Ford, "Lane-Keeping System." <https://www.ford.com/technology/driver-assist-technology/lane-keeping-system/> (accessed Nov. 05, 2020).
- [28] Mercedes-Benz, "Attention Assist." <https://www.mercedes-benz.es/vans/es/vito/tourer-commercial/equipment/standard-equipment-highlights/teaser-group/attention-assist> (accessed Nov. 05, 2020).
- [29] Fletcher Jones Motorcars, "What is Mercedes-Benz Attention Assist?," Mar. 11, 2019. <https://www.fjmercedes.com/mercedes-benz-attention-assist/> (accessed Nov. 05, 2020).
- [30] Volvo, "Driver Alert System." <https://www.volvocars.com/en-th/support/manuals/s60/2013w46/driver-support/driver-alert-system/driver-alert-system?id=7d874403c4f5cbffc0a801e8019b2c94-om-en-th-y283-2014-13w46> (accessed Nov. 05, 2020).
- [31] Volkswagen, "Driver alert system ." <https://www.volkswagen.co.uk/technology/car-safety/driver-alert-system> (accessed Nov. 05, 2020).
- [32] Lexus, "Lexus Safety System+ 2.0." <https://drivers.lexus.com/lexusdrivers/technology/safety#/lss-tabSlider> (accessed Nov. 05, 2020).
- [33] Subaru, "EyeSight Driver Assist Technology ." <https://www.subaru.com/engineering/eyesight.html> (accessed Nov. 05, 2020).
- [34] Honda, "Driver Attention Monitor." <https://www.hondainfocenter.com/2020/CR->

- V-Hybrid/Feature-Guide/Interior-Features/Driver-Attention-Monitor/ (accessed Nov. 05, 2020).
- [35] Bosch, "Driver Drowsiness Detection." Accessed: Nov. 05, 2020. [Online]. Available: [http://auto2015.bosch.com.cn/ebrochures2015/automated/cc/da/ddd/driver\\_drowsiness\\_detection\\_en.pdf](http://auto2015.bosch.com.cn/ebrochures2015/automated/cc/da/ddd/driver_drowsiness_detection_en.pdf).
- [36] Nissan, "Nissan's 'Driver Attention Alert' helps detect erratic driving caused by drowsiness and inattention," Apr. 01, 2015. <https://usa.nissannews.com/en-US/releases/nissan-s-driver-attention-alert-helps-detect-erratic-driving-caused-by-drowsiness-and-inattention#> (accessed Nov. 05, 2020).
- [37] Hyundai, "Connected and Caring ." <https://www.hyundai.news/eu/press-kits/new-hyundai-tucson-connected-and-caring/> (accessed Nov. 05, 2020).
- [38] Mazda, "Driver Attention Alert ." [https://www.mazda.com/en/innovation/technology/safety/active\\_safety/daa/](https://www.mazda.com/en/innovation/technology/safety/active_safety/daa/) (accessed Nov. 05, 2020).
- [39] Land Rover, "Driver Assistance ." <https://www.landrover.com/ownership/incontrol/driver-assistance.html> (accessed Nov. 05, 2020).
- [40] Kia Motors Worldwide, "Safety Technology ." [https://www.kia.com/worldwide/experience\\_kia/rnd/technology.do](https://www.kia.com/worldwide/experience_kia/rnd/technology.do) (accessed Nov. 05, 2020).
- [41] M. Y. Hossain and F. P. George, "IoT Based Real-Time Drowsy Driving Detection System for the Prevention of Road Accidents," in *2018 International Conference on Intelligent Informatics and Biomedical Sciences, ICIIBMS 2018*, Nov. 2018, pp. 190–195, doi: 10.1109/ICIIBMS.2018.8550026.
- [42] Fouzia, R. Roopalakshmi, J. A. Rathod, A. S. Shetty, and K. Supriya, "Driver Drowsiness Detection System Based on Visual Features," in *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, Sep. 2018, pp. 1344–1347, doi: 10.1109/ICICCT.2018.8473203.
- [43] J. He, W. Choi, Y. Yang, J. Lu, X. Wu, and K. Peng, "Detection of driver drowsiness using wearable devices: A feasibility study of the proximity sensor," *Appl. Ergon.*, vol. 65, pp. 473–480, Nov. 2017, doi: 10.1016/j.apergo.2017.02.016.
- [44] L. Nahar, R. Palit, A. Kafil, Z. Sultana, and S. Akter, "Real time driver drowsiness monitoring system by eye tracking using mean shift algorithm," in *2019 5th International Conference on Advances in Electrical Engineering, ICAEE 2019*, Sep. 2019, pp. 27–31, doi: 10.1109/ICAEE48663.2019.8975593.
- [45] J. Xu, J. Min, and J. Hu, "Real-time eye tracking for the assessment of driver

- fatigue," *Healthc. Technol. Lett.*, vol. 5, no. 2, pp. 54–58, 2018, doi: 10.1049/htl.2017.0020.
- [46] M. Miranda, A. Villanueva, M. J. Buo, R. Merabite, S. P. Perez, and J. M. Rodriguez, "Portable prevention and monitoring of driver's drowsiness focuses to eyelid movement using internet of things," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*, Mar. 2019, doi: 10.1109/HNICEM.2018.8666334.
- [47] J. I. Pilataxi, W. M. Vinan Robalino, and D. Chavez Garcia, "Design and Implementation of a Driving Assistance System in a Car-like Robot When Fatigue in the User is Detected," *IEEE Lat. Am. Trans.*, vol. 14, no. 2, pp. 457–462, Feb. 2016, doi: 10.1109/TLA.2016.7437179.
- [48] I. Teyeb, O. Jemai, M. Zaied, and C. Ben Amar, "A Drowsy Driver Detection System Based on a New Method of Head Posture Estimation," in *Intelligent Data Engineering and Automated Learning -- IDEAL 2014*, 2014, pp. 362–369.
- [49] W. Zhang and J. Su, "Driver yawning detection based on long short term memory networks," in *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings*, Feb. 2018, vol. 2018-January, pp. 1–5, doi: 10.1109/SSCI.2017.8285343.
- [50] M. Jafari and M. Soryani, "Driver Drowsiness Detection by Yawn Identification Based on Depth Information and Active Contour Model," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2019*, Jul. 2019, pp. 1522–1526, doi: 10.1109/ICICICT46008.2019.8993385.
- [51] Z. Jie, M. Mahmoud, Q. Stafford-Fraser, P. Robinson, E. Dias, and L. Skrypchuk, "Analysis of yawning behaviour in spontaneous expressions of drowsy drivers," in *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, Jun. 2018, pp. 571–576, doi: 10.1109/FG.2018.00091.
- [52] W. Deng and R. Wu, "Real-Time Driver-Drowsiness Detection System Using Facial Features," *IEEE Access*, vol. 7, pp. 118727–118738, Aug. 2019, doi: 10.1109/access.2019.2936663.
- [53] O. Khunpisuth, T. Chotchinasri, V. Koschakosai, and N. Hnoohom, "Driver Drowsiness Detection Using Eye-Closeness Detection," in *Proceedings - 12th International Conference on Signal Image Technology and Internet-Based Systems, SITIS 2016*, Apr. 2017, pp. 661–668, doi: 10.1109/SITIS.2016.110.
- [54] Optalert, "Ealge Industrial and Eagle Light." <https://www.optalert.com/explore-products/compare-products/> (accessed Nov. 04, 2020).
- [55] Optalert, "What is the Optalert JDS™ and why is it leading the drowsiness

- monitoring field? ." <https://www.optalert.com/what-is-the-optalert-jds-and-why-is-it-leading-the-drowsiness-monitoring-field/> (accessed Nov. 04, 2020).
- [56] Vigo, "Vigo Smart Headset." <https://www.wearvigo.com/> (accessed Oct. 23, 2020).
- [57] Smart Eye, "Driver Monitoring System." <http://smarteve.se/automotive-solutions/> (accessed Nov. 04, 2020).
- [58] Tobii Tech, "Tobii IS5 Integration Platform ." <https://tech.tobii.com/technology/is5-integration-platform/> (accessed Nov. 05, 2020).
- [59] NVIDIA, "Nvidia Drive IX ." <https://developer.nvidia.com/drive/drive-ix> (accessed Nov. 05, 2020).
- [60] SeeingMachines, "Guardian." <https://www.seeingmachines.com/guardian/guardian/> (accessed Nov. 04, 2020).
- [61] Bosch, "Camera-based life-saver: Bosch helps cars keep an eye on their passengers - Bosch Media Service." <https://www.bosch-presse.de/pressportal/de/en/camera-based-life-saver-bosch-helps-cars-keep-an-eye-on-their-passengers-204288.html> (accessed Nov. 04, 2020).
- [62] Hyundai Mobis, "Hyundai Mobis Develops Driver State Warning System to Prevent Big Traffic Accidents," Aug. 14, 2019. <https://en.mobis.co.kr/communityid/34/view.do?pageIndex=1&idx=4647> (accessed Nov. 05, 2020).
- [63] Jabil Automotive & Transportation, "Driver Monitoring System." Accessed: Nov. 06, 2020. [Online]. Available: <https://www.jabil.com/dam/jcr:b2ad3d1b-1c60-411a-9b0f-b289dc121cf1/fact-sheet-driver-monitoring-system.pdf>.
- [64] Meitrack, "Driver Fatigue Monitoring System." <http://www.meitrack.com/en/driver-fatigue-monitoring-system/> (accessed Nov. 16, 2020).
- [65] Stonkman CO., "Driver Fatigue Monitoring System ." <https://www.stonkam.com/products/Driver-Status-Detection-System-DMS11.html> (accessed Nov. 16, 2020).
- [66] Jimi IoT, "JC400D EdgeCam 2." <https://www.jimilab.com/products/jc400d-aivision-cam.html> (accessed Nov. 17, 2020).
- [67] Q. Abbas, "HybridFatigue: A real-time driver drowsiness detection using hybrid features and transfer learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 585–593, 2020, doi: 10.14569/ijacsa.2020.0110173.
- [68] J. Gwak, A. Hirao, and M. Shino, "An investigation of early detection of driver drowsiness using ensemble machine learning based on hybrid sensing," *Appl. Sci.*, vol. 10, no. 8, pp. 4–7, Apr. 2020, doi: 10.3390/APP10082890.

- [69] M. Sunagawa, S. I. Shikii, W. Nakai, M. Mochizuki, K. Kusakame, and H. Kitajima, "Comprehensive Drowsiness Level Detection Model Combining Multimodal Information," *IEEE Sens. J.*, vol. 20, no. 7, pp. 3709–3717, Apr. 2020, doi: 10.1109/JSEN.2019.2960158.
- [70] L. Giordano, "AIoT: a Kendryte K210 proof of concept," Universidad Complutense de Madrid, Madrid, 2020.
- [71] Sipeed Ltd, "Maix Cube Datasheet," 2020. [https://maixpy.sipeed.com/maixpy/en/develop\\_kit\\_board/maix\\_cube.html](https://maixpy.sipeed.com/maixpy/en/develop_kit_board/maix_cube.html) (accessed Feb. 21, 2021).
- [72] Sipeed Ltd, "MaixUI: UI framework.," Sep. 01, 2020. <https://github.com/sipeed/MaixUI> (accessed Feb. 17, 2021).
- [73] Sipeed Ltd, "MaixCube - MaixPy." [https://maixpy.sipeed.com/maixpy/en/develop\\_kit\\_board/maix\\_cube.html#get-started](https://maixpy.sipeed.com/maixpy/en/develop_kit_board/maix_cube.html#get-started) (accessed Feb. 17, 2021).
- [74] Sipeed Ltd, "Sipeed download center." <https://dl.sipeed.com/> (accessed Feb. 17, 2021).
- [75] Sipeed Ltd, "Build Maixpy from source code." <https://github.com/sipeed/MaixPy/blob/master/build.md> (accessed Feb. 17, 2021).
- [76] C.-H. Weng, Y.-S. Lai, and S.-H. Lai, "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network," 2016, [Online]. Available: <http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/>.
- [77] R. Ghoddoosian, M. Galib, and V. Athitsos, "A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, 2019, Accessed: Apr. 26, 2021. [Online]. Available: <https://github.com/rezaghoddoosian>.
- [78] MaixPy, "Video," Feb. 03, 2021. [https://maixpy.sipeed.com/en/libs/machine\\_vision/video.html](https://maixpy.sipeed.com/en/libs/machine_vision/video.html) (accessed Feb. 17, 2021).
- [79] Sipeed Ltd, "Getting frame from video," Jun. 02, 2020. <https://github.com/sipeed/MaixPy/issues/254> (accessed Feb. 17, 2021).
- [80] National Highway Traffic Safety Administration, "An Analysis of Driver Inattention Using a Case-Crossover Approach On 100-Car Data: Final Report," 2010. Accessed: Mar. 31, 2021. [Online]. Available: [www.ntis.gov](http://www.ntis.gov).
- [81] D. Gerge, P. Sokolovsky, and LLC OpenMV, "Image - Machine Vision," Jan. 28, 2021. [https://docs.openmv.io/library/omv.image.html?highlight=haar\\_cascade#image.image.Image.image.find\\_features](https://docs.openmv.io/library/omv.image.html?highlight=haar_cascade#image.image.Image.image.find_features) (accessed Feb. 17, 2021).
- [82] Sipeed Ltd, "Face detection."

- [https://maixpy.sipeed.com/maixpy/en/course/ai/image/face\\_detect.html](https://maixpy.sipeed.com/maixpy/en/course/ai/image/face_detect.html)  
(accessed Feb. 19, 2021).
- [83] Sipeed, "MaixPy support new nncase & Kmodel V4 Now."  
<https://en.bbs.sipeed.com/t/topic/1790> (accessed Apr. 08, 2021).
- [84] Kaggle, "Face Images with Marked Landmark Points."  
<https://www.kaggle.com/drgilermo/face-images-with-marked-landmark-points>  
(accessed Feb. 21, 2021).
- [85] Sipeed Ltd, "Basic knowledge of MaixPy AI hardware acceleration."  
[https://maixpy.sipeed.com/maixpy/en/course/ai/basic/maixpy\\_hardware\\_ai\\_basic.html#model-conversion](https://maixpy.sipeed.com/maixpy/en/course/ai/basic/maixpy_hardware_ai_basic.html#model-conversion) (accessed Feb. 24, 2021).
- [86] Kendryte, "NNCase." <https://github.com/kendryte/nncase> (accessed Feb. 24, 2021).
- [87] Tensorflow, "Pruning comprehensive guide | TensorFlow Model Optimization."  
[https://www.tensorflow.org/model\\_optimization/guide/pruning/comprehensive\\_guide.md](https://www.tensorflow.org/model_optimization/guide/pruning/comprehensive_guide.md) (accessed Feb. 24, 2021).
- [88] Tensorflow, "Pruning in Keras example | TensorFlow Model Optimization."  
[https://www.tensorflow.org/model\\_optimization/guide/pruning/pruning\\_with\\_keras#train\\_a\\_model\\_for\\_mnist\\_without\\_pruning](https://www.tensorflow.org/model_optimization/guide/pruning/pruning_with_keras#train_a_model_for_mnist_without_pruning) (accessed Feb. 24, 2021).
- [89] Kendryte, "Using a quantized model with nncase."  
<https://github.com/kendryte/nncase/issues/73> (accessed Feb. 24, 2021).
- [90] C. E. Thomaz, "FEI Face Database." <https://fei.edu.br/~cet/facedatabase.html>  
(accessed Feb. 27, 2021).
- [91] Sipeed, "Maixpy model online training."  
<https://www.maixhub.com/ModelTraining> (accessed May 06, 2021).
- [92] Sipeed Ltd, "Image (Machine Vision)."  
[https://maixpy.sipeed.com/maixpy/en/api\\_reference/machine\\_vision/image/image.html#imagegrayscale\\_to\\_rgbg\\_value](https://maixpy.sipeed.com/maixpy/en/api_reference/machine_vision/image/image.html#imagegrayscale_to_rgbg_value) (accessed Mar. 08, 2021).
- [93] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes Closeness Detection from Still Images with Multi-scale Histograms of Principal Oriented Gradients," 2014. Accessed: Mar. 08, 2021. [Online]. Available: [http://parnec.nuaa.edu.cn/\\_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html](http://parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html).
- [94] K. Cortacero, T. Fischer, and Y. Demiris, "RT-BENE: A Dataset and Baselines for Real-Time Blink Estimation in Natural Environments," *Proc. IEEE Int. Conf. Comput. Vis. Work.*, 2019, Accessed: Mar. 15, 2021. [Online]. Available: [www.imperial.ac.uk/Personal-Robotics/](http://www.imperial.ac.uk/Personal-Robotics/).
- [95] E. Sojka, J. Gaura, M. Krumnikl, T. Fabián, M. Němec, and R. Fusek, "MRL Eye Dataset." <http://mrl.cs.vsb.cz/eyedataset> (accessed Mar. 16, 2021).