PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

University Mohammed Khider BISKRA

Report

Presented to obtain the diploma of academic master in Artificial intelligence

Path: (Computer science)

Driver's Drowsiness and fatigue detection system

By:

Hamed LAOUZ

Academic year: 2019/2020

Acknowledgment

First of all, I thank God for giving me strength and patience,
the courage and will during my studies to reach this point
And achieve my goals to complete this work.

My deep thanks and gratitude go back to:

Mr. Sohyeb AYAD.

To direct this work in the right direction and for its accurate and valuable advice.

I would like to thank the members of the jury, by name, for their agreement to adjudicate this
Work in these exceptional situations.

Dedication

I dedicate this modest work

*To my dear **parents** for their attention and support and encouragement*

*To my dear brothers: **Ayoub, Abd Al Hakim** and his wife **Rima.***

*My dear sister **Hiba**.*

To my dear friends;

MANCER M'hammed** and **SNOUSI chems eddine

*To botth **LAOUZ** and **HADRI** family*

To the people I will never forget, my colleagues at

Computer science group 2020.

To all those who have helped me from near or far to carry out this work.

ملخص

تتسبب حوادث المرور دائمًا في خسائر مادية وبشرية كبيرة. أحد أهم أسباب هذه الحوادث هو العامل البشري الذي ينتج عادة عن التعب أو النعاس. لمعالجة هذه المشكلة ، تم اقتراح عدة طرق للتنبؤ بحالة السائق. بعض الحلول مبنية على قياس سلوك السائق مثل: حركة الرأس ، مدة طرفة العين ، ملاحظة تعبير الفم ... إلخ ، بينما تعتمد الحلول الأخرى على القياسات الفسيولوجية للحصول على معلومات حول الحالة الداخلية لجسم السائق.

يهدف هذا العمل إلى التوصل الى ايجاد حل لعلاج هاته المشكلة عن طريق إنشاء نظام مدمج في السيارة يستطيع الكشف عن نعاس السائق وإرهاقه في الوقت الفعلي باستخدام كاميرا مثبتة في السيارة تلتقط إيماءات وحركة السائق ليقوم النظام عندها بتحليل حالة العين لمعرفة ان كانت العين مفتوحة أم لا ،كما عزز النظام بوضع لمراقبة حالة الفم للتحقق مما إذا كان السائق يتثاءب أم لا والذي قد يساعد في التنبؤ بالنعاس والتعب مبكرًا.

الكلمات المفتاحية:

تعب ونعاس السائق, المقاييس السلوكية للسائق ,طرفة العين, القياسات الفيسيولوجية.

Résumé

Les accidents de la circulation entraînent toujours de grandes pertes matérielles et humaines. L'une des principales causes de ces accidents est le facteur humain, qui résulte généralement de la fatigue ou de la somnolence. Pour résoudre ce problème, plusieurs méthodes ont été proposées pour prédire l'état du conducteur. Certaines solutions reposent sur la mesure du comportement du conducteur comme : le mouvement de la tête, le temps de clignotement, la note d'expression de la bouche ... etc., tandis que d'autres reposent sur des mesures physiologiques pour obtenir des informations sur l'état interne du conducteur.

Dans ce travail nous avons réalisé une solution pour traiter ce problème en créant un système embarqué capable de détecter la somnolence / fatigue du conducteur en temps réel à l'aide d'une caméra intégrée dans la voiture qui capte les gestes et les mouvements du conducteur, afin que le système analyse ensuite l'état de l'œil pour voir si l'œil est ouvert ou non, le système est également assisté par un mode de surveillance de l'état de la bouche, pour vérifier si le conducteur bâille, ce qui peut aider à prédire la somnolence tôt.

Mots clés : *somnolence et fatigue du conducteur, mesures du comportement du conducteur, clignement des yeux, mesures des signaux physiologiques, système intégré.*

Abstract

Traffic accidents always cause great material and human losses. One of the main causes of these accidents is the human factor, which usually results from fatigue or drowsiness. To address this problem, several methods have been proposed to predict the driver's condition. Some solutions are based on measuring the driver’s behavior such as: head movement, blink time, mouth expression note ... etc., while other solutions rely on physiological measurements to obtain information about the driver’s internal condition.

This work aims to find a solution to treat this problem by creating a system that can detect driver drowsiness/fatigue in real time using a built-in camera in the car that captures the driver's gestures and movement, so that the system then analyzes the eye's condition to see if the eye is open or not, and the system has also strengthened By monitoring the condition of the mouth, to check whether the driver is yawning, which can help predict sleepiness early.

Key words: *Driver’s drowsiness and fatigue, Driver behavioral measures, eye blink, physiological signals measures, embedded system.*

Contents

Tables list

Figures list

Algorithms list

Abbreviations list

ML: Machine learning.

DL: Deep learning.

AI: Artificial intelligence.

DBN: Deep Belief network.

RBM: Restricted Boltzmann Machine.

SVM: Support vector machine.

ANN: Artificial neural network.

CNN: Convolution neural network.

RNN: Recurrent neural network.

NHTSA:US National Highway Traffic Safety Administration.

NSF: National Sleep Foundation.

HRV: Heart rate variability.

KSS: Karolinska Sleepiness Scale.

SSS: Stanford Sleepiness Scale.

EAR: Eye Aspect ratio.

MAR: Mouth Aspect ratio.

PERCLOS: Percentage of Eyelid Closure Over the Pupil Over Time.

IR: InfraRed.

ECG: Electrocardiogram.

EMG: Electromyogram.

EEG: Electroencephalogram.

EOG: Electro-oculogram.

KNN: K-Nearest Neighbors.

General Introduction

General introduction

The drowsiness is a normal biological state could happen for every human being while the scientists defines it as a state of impaired awareness associated with a desire or inclination to sleep **[1].**

But in some cases, this type of state can harm human life, especially while driving and it could be a cause for traffic accident because according to the recent statistics one out of every six (16.5%) deadly traffic accidents, and one out of eight (12.5%) crashes requiring hospitalization of car drivers or passengers is due to drowsy driving, also ,a recent AAA survey found that two out of every five drivers (41%) admitted to having fallen asleep at the wheel at some point, with one in ten drivers (10%) reporting they did so within the past year, and more than one-quarter of drivers (27%) admitting they had driven while they were “so sleepy that [they] had a hard time keeping[their] eyes open” within the past month **[2].** In term of cost, one analysis estimated the cost of automobile accidents attributed to sleepiness to be between $29.2 to $37.9 billion **[3].**

That is why we are interested in finding a solution to this problem by developing an intelligent system built into the car that can detect driver drowsiness or fatigue in real time using the best possible method that can detect drowsiness with the least error and is more appropriate to our environment, so that the system then alerts the driver in order to prevent potential accidents.

The objectives of our work are:

- Find the different methods used to detect the drowsiness from the driver who set behind the wheel.
- Develop an embedded system that can detect efficiently the driver’s drowsiness in real time.

We will present through our manuscript four chapter; The first chapter is talking about the basics of machine learning. The second chapter is a Literature Review on Driver’s Drowsiness and Fatigue Detection which talk about the recent methods used to detect the drowsiness. The third chapter is the design and conception of our system, in this chapter we will present our system architecture and components, also the methods used to in our system detect the drowsiness. And the last chapter present our system that we had developed with the obtained results we get it.

Chapter 1:
Generalities on machine learning

1.1 Introduction

Machine Learning (ML) is undeniably one of the most influential and powerful technologies in today's world. ML is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data. This mass of data is useless unless we analyze it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making **[4].**

But in the other side, Deep learning (DL) is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making.

DL has proven its strength in multiple fields requiring complex tasks compared to machine learning because: Machine learning algorithms are designed to "learn" to act by understanding labeled data and then use it to produce new results with more datasets. However, when the result is incorrect, there is a need to "teach them "unlike the deep learning networks because they do not require human intervention, as multilevel layers in neural networks place data in a hierarchy of different concepts, which ultimately learn from their own mistakes **[5].**

A few years back, deep learning changed the paradigm for ML, and with that we saw many new products emerge in various verticals including health care, education, ...etc. Most of these are possible because of the unlimited compute of the cloud, along with the intersection of a vast amount of data. However, this has also led to thoughts and questions on the challenges around privacy and data governance, and it's opened a greenfield for various companies looking to bring the intelligence closer to the data, on users' devices. On-device ML also brings other benefits like real-time performance and personalization, and therefore is an important and natural space for growth in the ML world, despite its associated challenges **[6].**

1.2 Machine Learning

1.2.1 Definition

Machine learning has many definitions:

Definition 1:

• The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience **[7].**

Definition 2:

• Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed **[8].**

Definition 3:

• A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E **[7].**

Definition 4:

• Machine learning is a branch of study in which a model can learn automatically from the experience based on data without exclusively being modeled like in statistical models. Over a period and with more data, model predictions will become better **[9].**

1.2.2 Types of machine Learning

Machine learning systems can be classified according to the amount and type of supervision the get during training. There are four major categories:

- Supervised learning.
- Unsupervised learning.
- Semi supervised learning.
- Reinforcement learning.

1.2.2.1 Supervised learning: Learning with a labeled training set

The defining characteristics of supervised learning is the availability of annotated training data. The name invokes the idea of a 'supervisor' that instructs the learning system on the labels to associate with training examples. Typically, these labels are class labels in classification problems. Supervised learning algorithms induce models from these training data and these models can be used to classify other unlabeled data.**[10].**

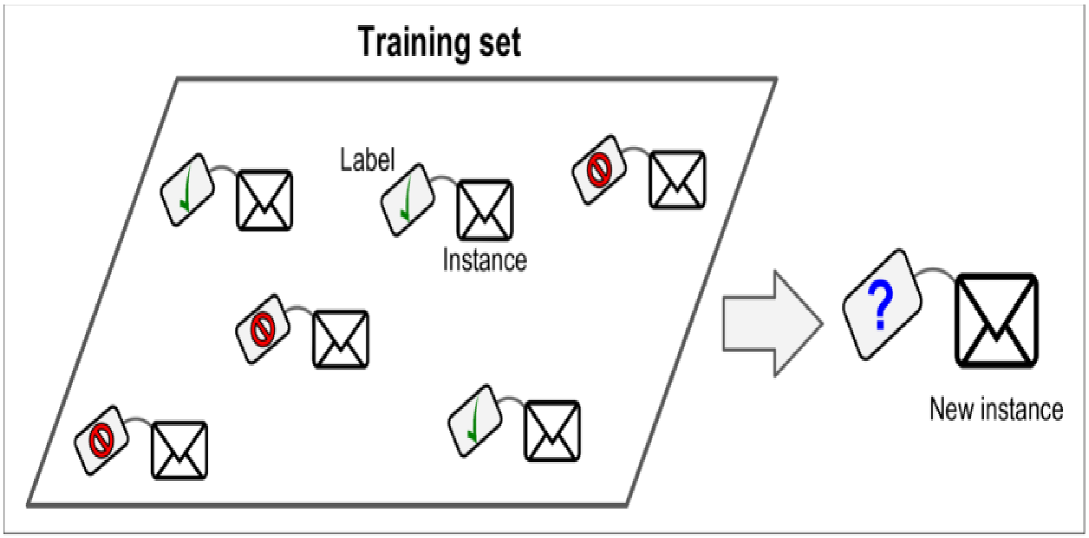


Figure 1. A labeled training set for supervised learning (e.g., spam classification) **[13].**

1.2.2.2 Unsupervised learning: Discover patterns in unlabeled data

When a model does not require labeled data, we refer to unsupervised learning. These types of models try to learn or extract some underlying structure in the data or reduce down to its most important features. Clustering, dimensionality reduction, and some forms of feature extraction, such as text processing, are all unsupervised techniques **[11].**

The goal of unsupervised learning is for a machine to process a set of training data and then extract structure which cognitive scientists believe to be relevant to perception and conception .For example ,given a training set containing examples of handwritten digits ,we would like our machine to learn represent digit class without being told what the digit classes are .when the machine is given a new test image of, say , a 3,we hope that somewhere within the machine there will be a flashing light saying "I think this is a 3!" **[12]**.

1.2.2.3 Semi supervised learning

Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called semi supervised learning.

Some photo-hosting services, such as google photos, are good examples of this. Once you upload all your family photos to the service, it automatically recognizes that the same person A shown up in photos 1,5and 11,while another person B shows up in photos 2,5 and 7.this is the unsupervised part of the algorithm(clustering).now all the system needs is for you to tell it who these people are. Just one label per person and it is able to name everyone in every photo, which is useful for searching photos.

Most semi supervised learning algorithms are combinations of unsupervised and supervised algorithms. for example, deep belief networks (DBNs) are based on unsupervised components called restricted Boltzmann machines (RBMs) stacked on the top of one another. RBMs are trained sequentially in an unsupervised manner, and then the system is fine-tuned using supervised learning techniques **[13].**

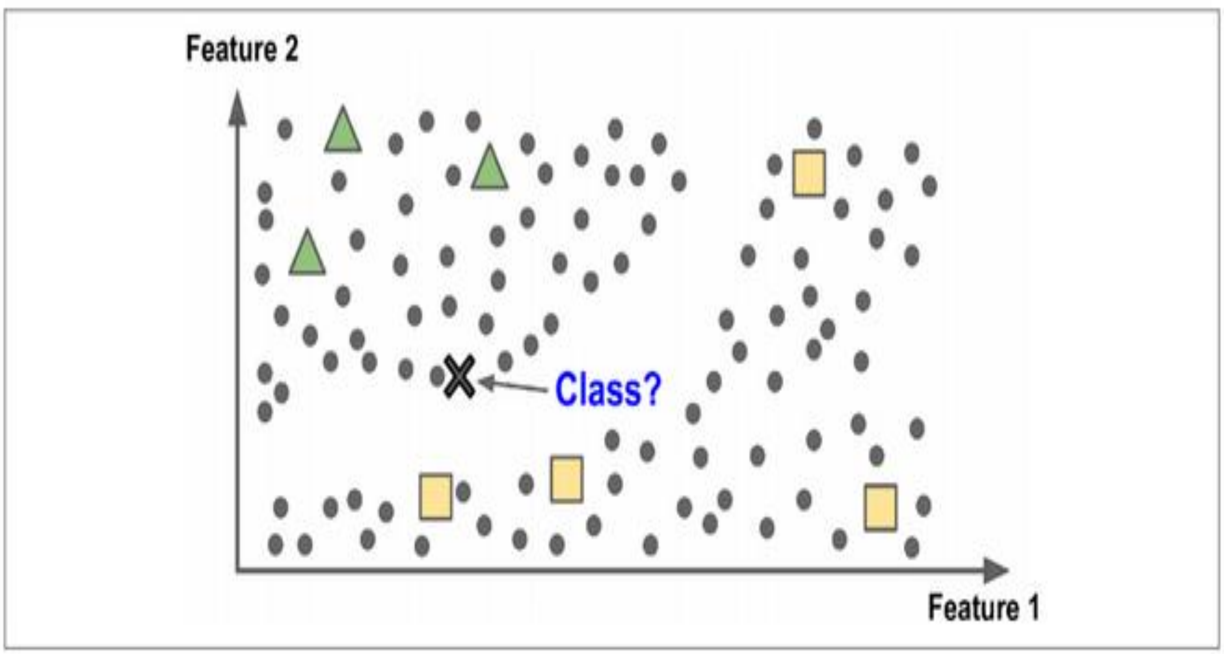


Figure 2. semi-supervised learning with two class (triangles, squares): the unlabeled examples(circles)help classify an instance (the cross cross) into the triangle class rather than the square class, even though it is closer to the labeled squares **[13].**

1.2.2.4 Reinforcement learning: learn to act based on feedback/reward

Reinforcement learning is learning what to do-how to map situation to actions-so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them.in the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics-trial and-error search and delayed reward-are the two most important distinguishing features of reinforcement learning **[14].**

Example: learn to play Go, reward: *win or lose.*

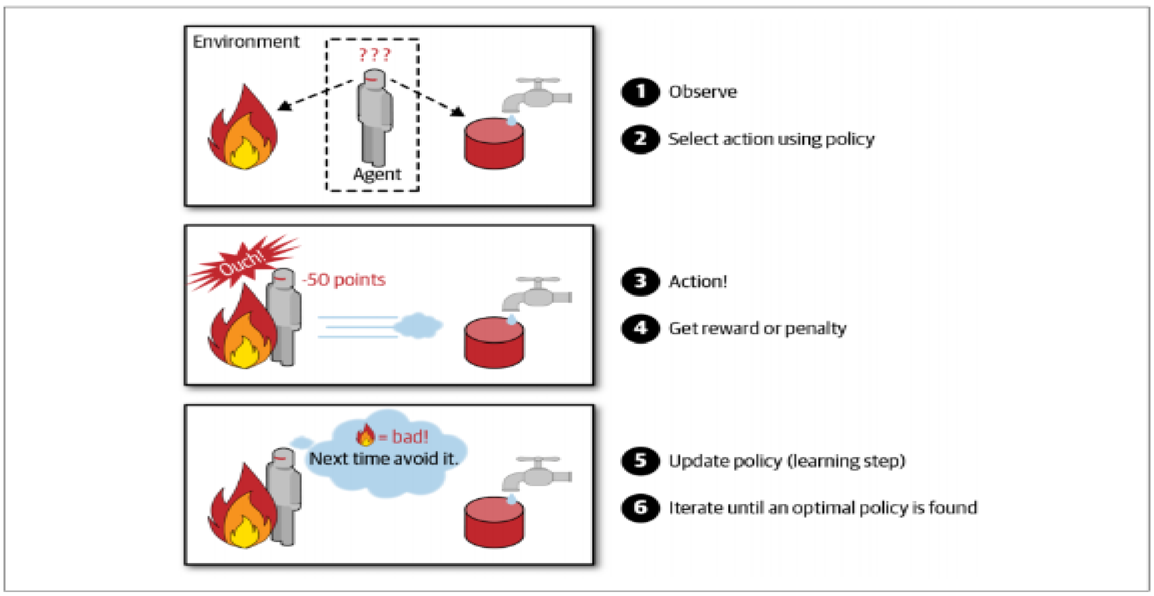


Figure 3.Reinforcement learning **[13].**

1.2.3 Machine learning process

In **[15]** the author said that the machine learning process is an iterative process. It cannot be completed in one go. The most important activities to be performed for a machine learning solution are as follows:

- ❖ Define the machine learning problem (it must be well-defined).
- ❖ Gather, prepare, and enhance the data that is required.

- Use that data to build a model. This step goes in a loop and covers the following sub steps. As time, it may also lead to revisiting Step 2 on data or even require the redefinition of the problem statement:

    - Select the appropriate model/machine learning algorithm.
    - Train the machine learning algorithm on the training data and build the model.
    - Test the model.
    - Evaluate the results.
    - Continue this phase until the evaluation result is satisfactory and finalize the model.

- Use the finalized model to make future predictions for the problem statement.

1.2.4 Machine learning models

1.2.4.1 Support vector machine

The best way of introducing Support vector machines (SVMs) is to consider the simple task of binary classification. Many real-world problems involve prediction over two classes. An SVM is an abstract learning machine which will learn from a training data set and attempt to generalize and make correct predictions on novel data. For the training data we have a set of input vectors, denoted xi, with each input vector having a number of component features. These input vectors are paired with corresponding labels. Which we donate yi, and there is m such pairs (*i* =1..., m) **[27].**

The concept of Support Vector Machines was first developed by researchers of AT&T Bell Laboratories as" a training algorithm that maximizes the margin between the training patterns and the decision boundary". Further research would develop the base algorithm into a flexible and accurate machine learning technique providing the ability for classification, and it would see use in financial forecasting, predicting medication adherence, and mobile communications. The algorithm can be trained (i.e. learn by example much like a human being), which gives SVMs the ability to label and categorize information presented to it. Metrics representing the entity in question are used by the SVM for classification. This is performed by projecting the metric" data from a low-dimensional space to a space of higher dimension" to increase the ability of the SVM to determine decision boundaries, and thus increase the accuracy of classification **[16].**

1.2.4.2 Genetic algorithms

The most popular technique in evolutionary computation research has been the genetic algorithm. In the traditional genetic algorithm, the representation used is a fixed-length bit string. Each position in the string is assumed to represent a particular feature of an individual, and the value stored in that position represents how that feature is expressed in the solution. Usually, the string is “evaluated ad a collection of structural features of a solution that have little or no interactions”. The analogy may be drawn directly to genes in biological organisms. Each gene represents an entity that is structurally independent of other genes **[17].**

1.2.4.3 Artificial neural network (ANN)

Artificial neural networks are mathematical inventions inspired by observations made in the study of biological systems, though loosely based on the actual biology. An ANN can be described as mapping an input space to an output space. This concept is analogous to that of a mathematical function. The purpose of a neural network is to map an input into a desired output. While patterned after the interconnections between neurons found in biological systems, ANNs are no more related to real neurons than feathers are related to modern airplanes. Both biological systems, neurons and feathers, serve a useful purpose, but the implementation of the principals involved has resulted in man-made inventions that bear little resemblance to the biological systems that spawned the creative process **[18].**

1.2.5 Limitations of machine learning

Despite its wide applications in general education such as curricular integration, learner interaction. ML shown some limitations in term of how it can promote learner’s DL. Research in machine learning has been primarily focused on the system functions like individualized cognitive diagnosis, error detection, expert model overlay, and learning path determination. Few efforts have been directed toward understanding the role of machine learning pertaining to scaffolds of deep and surface learning. It is obvious that teaching learners to solve problems without facilitating their abilities to transfer knowledge to new learning may render the learning experience less relevant and meaningful since learners tend to repeat the problem steps without questioning the meaning of the steps and understanding their connection to new learning. another reason why ML has been slow in moving forward to deep and surface cognitive scaffolds is the technical challenges in terms of solutions to deep learning and knowledge transfer scaffolds **[19].**

1.3 Deep learning

Deep Learning is part of ML in which we use models of a specific type, called deep artificial neural networks. Since their introduction, ANNs have gone through an extensive evolution process, leading to a number of subtypes, some of which are very complicated.

1.3.1 Definitions

DL has many definitions:

Definition 1:

Deep Learning is a new area of ML research, which has been introduced with the objective of moving ML closer to one of its original goals: Artificial Intelligence **[20].**

Definition 2:

DL can be defined as the recursive of the learning of the features in the dataset by the algorithms itself as against the traditional machine learning algorithms which require features to be specified in advance. DL models works on the principles of ANN which consists of number of layers arranged in sequence (hierarchical order) and each layer act as output to subsequent layer. DL models use drop out techniques (dropping some of the nodes in the network during the training process) to overcome the problem found in artificial neural network. The depth of the deep learning models depends on the number of layers it contains. DL models are capable of processing linear and nonlinear data **[21].**

Definition 3:

DL is a class of ML techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification **[22].**

Definition 4:

DL is a connectionist approach to AI. A connectionist system is composed of a large number of simple components that collectively exhibit complex behavior. DL specifically employs multiple layers of components arranged in an acyclic graph (i.e. without loops). There are many kinds of layers but the common property is that the layer is differentiable. Another way of saying this is, the gradient can be calculated for any layer **[23].**

1.3.2 Deep learning models

1.3.2.1 Convolution neural network (CNN)

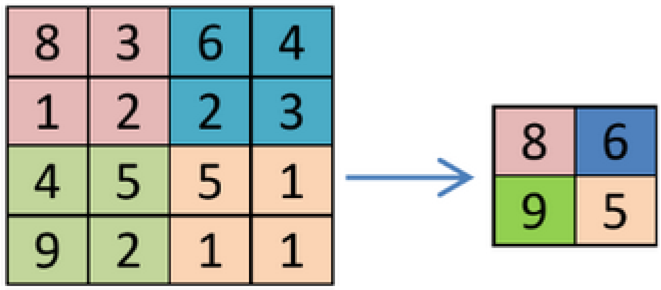
CNN are useful to process and classify images efficiently.it is similar to the feed forward networks except the fact that it can handle more weights and layers can handle three dimension such as width, height and depth during the transformation process. Convolution network involve the following layers: Input, ConV, ReLU, Pooling and Fully connected layers. The image is processed by the network as below:

I. **Input layer** receives the inputs such as raw image of size x1*x2**x3 where x1 is the height, x2 is the width and x3 is the channel or depth and pass it to the next layer i.e. Convolution layer.

Ii. **Conv layer** transforms the image into three-dimensional object using filters. The convolution layer use j filters of size y1*y1*x3 where the filters height and width will be smaller than the original image but the channel or the depth will be the same (i.e. y1<x1) the filters produce x1-y1+1 feature map.

iii. **ReLU layer** applies activation function to the received input.

iv. **Pooling layer** will reduce the size of the imagelike. width and height of the image through down sampling technique but the not the depth of the image which is retained as same. Here the feature maps are subsampled using mean or max pooling or l2 norm pooling and filters of size g*g and with stride h.



Here the max pooling of size 2*2 and stride2 is used.

v. **Fully connected layer** acts as a classifier using the features received from the previous layers. The fully connected layer multiplies the features(input) with weight matrix and adds error(vector) to **[21].**

1.3.2.2 Recurrent Neural networks (RNNs)

The RNN was first developed in the 1980s. Its structure consists of an input layer, one or more hidden layers, and an output layer. RNNs have chain-like structures of repeating modules with the idea behind using these modules as a memory to store important information from previous processing steps. Unlike feedforward neural networks, RNNs include a feedback loop that allows the neural network to accept a sequence of inputs. This means the output from step t - 1 is fed back into the network to influence the outcome of step t, and for each subsequent step. Therefore, RNNs have been successful in learning sequences. **Fig 4** shows the sequential processing in RNN **[28].**

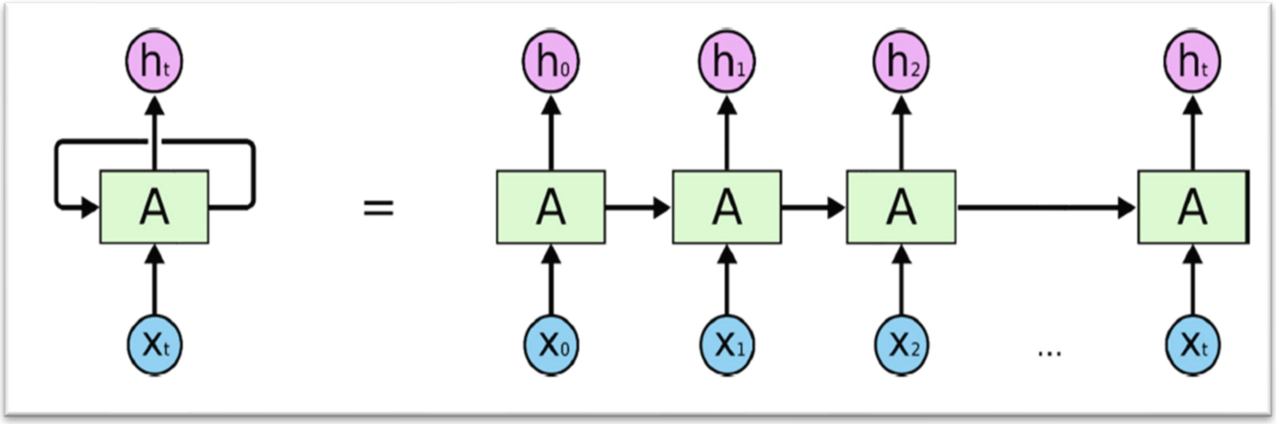


Figure 4. Sequential processing in an RNN **[29].**

Recurrent neural networks are a class of neural networks that exploit the sequential nature of their input. Such inputs could be a text, a speech, time series, and anything else where the occurrence of an element in the sequence is dependent on the elements that appeared before it **[26].**

1.3.2.3 Deep Belief networks

The DBN contains two parts. RBM and Directed sigmoid network. DBN can work well with the unlabeled and unsupervised inputs using the probabilistic approach. Layers in the network have latent stochastic variables which are used to extract the feature from the input and acts as a hidden unit. The state of stochastic variables changes to 1 or 0 based on the weights and error value received from the previous latent stochastic variables. Here the layers are connected not the units within the layers are connected between them. The top two layers in the network are undirected connections among them wherein the lower layers have directed connections from the top layers. RBM is bipartite graph which contains two groups of units one is visible unit and another is hidden unit and it connects two nodes from different groups but not from same group.

DBNs are formed through stacking of the RBM layers. Learning is done through Greedy learning process which selects the optimum parameters during each learning process aim at overall optimum at the end. Greedy learning algorithm uses generative weights and the top down approach. The values of the latent variables can be determined by using bottom up pass approach which uses the generative weights in opposite direction. DBNs are used in image and voice recognition video and motion capture, classification and clustering problems **[21].**

1.4 Mobile (On-device) machine learning

1.4.1 Definition of an Edge Device

Before we elaborate on on-device learning, it is helpful to define what we mean by a device, or specifically an edge device, in the context of on-device learning. We define an edge device to be a device whose compute, memory, and energy resources are constrained and cannot be easily increased or decreased. These constraints may be due to form-factor considerations (it is not possible to add more compute or memory or battery without increasing the size of the device) or due to cost considerations (there is enough space to add a GPU to a washing machine but this would increase its cost prohibitively). This definition of an edge device applies to all such consumer and industrial devices where resource constraints place limitations on what is available for building and training AI models. Any cloud solution such as Amazon AWS, Google Cloud Platform, Microsoft Azure, or even on-premise computing clusters do not fit the edge definition because it is easy to provision additional resources as needed. Likewise, a workstation would not be considered an edge device because it is straightforward to replace its CPU, add more memory, and even add an additional GPU card. A standard laptop on the other hand would be considered an edge device as it is not easy to add additional resources as needed, even though their resources generally far exceed what is normally considered as available in a consumer edge device **[24].**

1.4.2 On device learning definition

Mobile machine learning is a growing field of ML that doesn’t involve data centers and giant clusters of high-powered GPU machines. Instead, we can now run machine learning operations on mobile devices to avoid the network bottleneck Applications such as Spotify and Apple Music, leaders in the music industry use a combination of on-device network and cloud services to generate results. However, on-device deep learning techniques have significantly evolved over the past two years, and they now cover a lot of everyday use cases without making any network calls. These include speech recognition, image recognition, object detection, gesture recognition, translation and text classification.

Choosing a deep learning strategy for mobile applications might be harder because of the lack of relevant resources to get started **[25].**

1.4.3 On device machine learning improvement approaches

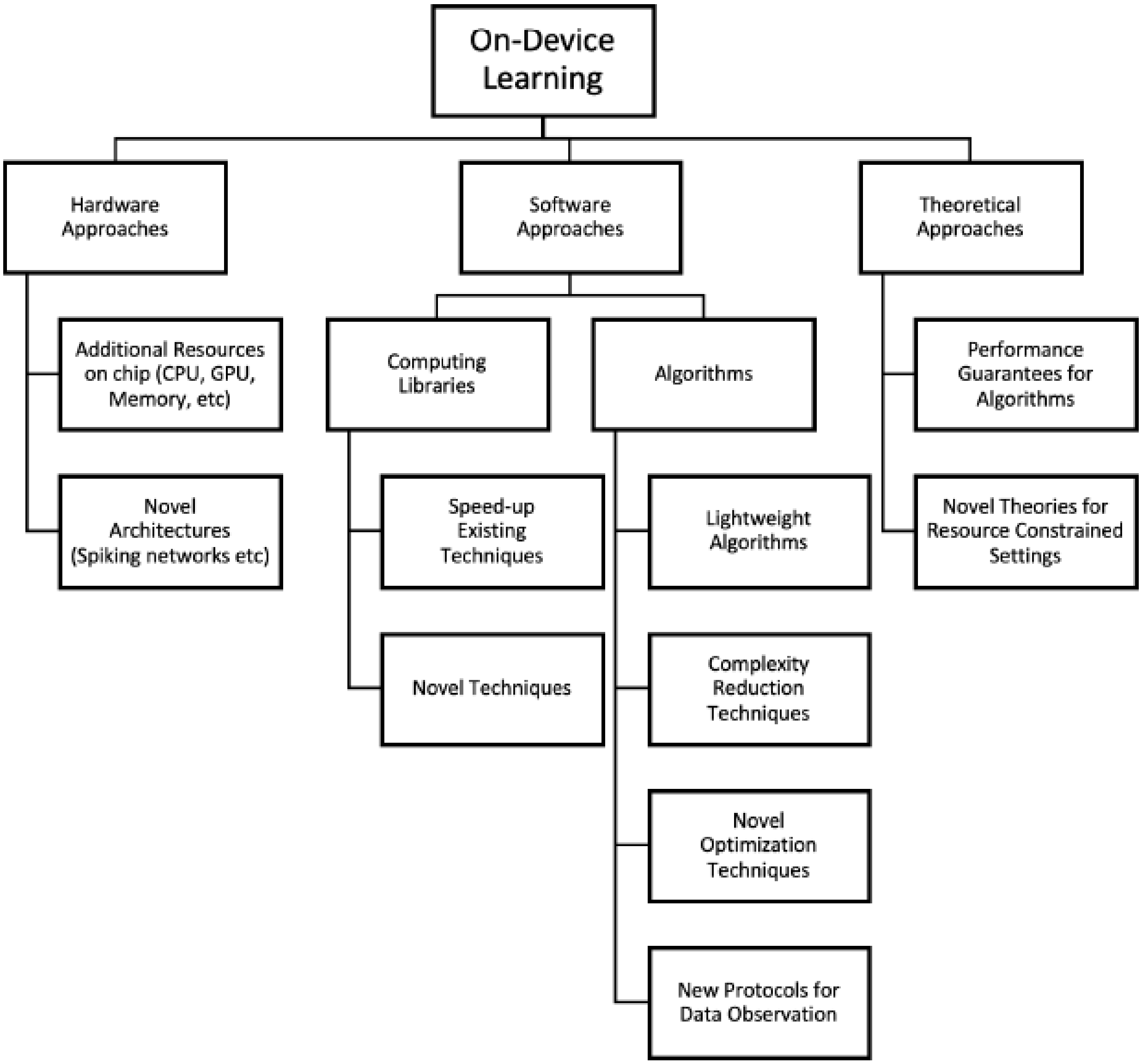


Figure 5. Different approaches to improving on-device learning **[24].**

Figure 5 shows an expanded hierarchical view of the different levels of the edge learning stack and highlight different ways to improve the performance of model training on the device at each level. The hardware approaches involve either adding additional resources to the restricted form-factor of the device or developing novel architectures that are more resource efficient.

Software approaches to improve model training involve either improving the performance of computing libraries such as OpenBLAS, Cuda, CuDNN or improving the performance of the machine learning algorithms themselves. Finally, theoretical approaches help direct new research on ML algorithms and improve our understanding of existing techniques and their generalizability to new problems, environments, and hardware **[24].**

1.5 Why use machine learning on mobile devices:

In **[15]** the author said that ML is needed to extract meaningful and actionable information from huge amounts of data. A significant amount of computation is required to analyze huge amounts of data and arrive at an inference. This processing is ideal for a cloud environment. However, if we could carry out ML on a mobile, the following would be the advantages:

• ML could be performed offline, as there would be no need to send all the data that the mobile has to the network and wait for results back from the server.

• The network bandwidth cost incurred, if any, due to the transmission of mobile data to the server is avoided.

• Latency can be avoided by processing data locally. Mobile machine learning has a great deal of responsiveness as we don't have to wait for connection and response back from the server. It might take up to 1-2 seconds for server response, but mobile machine learning can do it instantly.

• Privacy-this is another advantage of mobile machine learning. There is no need to send the user data outside the mobile device, enabling better privacy.

Some classic examples of ML on mobile devices are as follows:

- Speech recognition.
- Computer vision and image classification.
- Gesture recognition.
- Translation from one language into another.
- Interactive on-device detection of text.
- Autonomous vehicles, drone navigation, and robotics.
- Patient-monitoring systems and mobile applications interacting with medical devices**.**

1.6 Conclusion

In short, ML and DL are one of the major and biggest areas of AI. But DL is an advanced form of machine learning that came to deal with unstructured and overwhelming information. Thus, DL can satisfy more problems easily and more efficiently. Also, the ML- or DL- application on devices occupied a large place in the tasks and fields of artificial intelligence because it provides some benefits such as privacy and confidentiality of information inside the devices because there is no need for servers, which is one of the features required in modern applications. But this concept requires some conditions that must be verified and some methods that must be followed before implementation due to the limited resources and computing time in these devices which must be taken into consideration.

Chapter 2:
Related works

2.1 Introduction

According to available statistical data, over 1.3 million people die each year on the road and 20 to 50 million people suffer non-fatal injuries due to road accidents **[30].** Based on police reports, the US National Highway Traffic Safety Administration (NHTSA) conservatively estimated that a total of 100,000 vehicle crashes each year are the direct result of driver drowsiness. These crashes resulted in approximately 1,550 deaths, 71,000 injuries and $12.5 billion in monetary losses **[31].**

In 2009, the US National Sleep Foundation (NSF) reported that 54% of adult drivers have driven a vehicle while drowsy and 28% of them actually fell asleep **[32].** The German Road Safety Council (DVR) claims that one in four highway traffic fatalities are a result of momentary driver drowsiness **[33].** These statistics suggest that driver drowsiness is one of the main causes of road accidents. When the driver is drowsy, he may not pay attention to the road and this may cause to an accident. In order to prevent these types of accidents, there are two main measures that can detect drowsiness:

Driver behavioral measures: this measure mainly concerned with the extent of the driver's focus on his driving, by observing the driver's head movements, eye blink, yawning or facial expression, etc. Which mostly detected by image processing techniques **[34, 35].**

Physiological signals measures: this category of measures focuses on analyzing the physiological signals of the driver to detect the drowsiness. Also, it provides an accurate measure of drowsiness because of the strong relationship of physiological signals with driver fatigue **[36, 37].** Beside to that, the physiological signals results are obtained in earlier stages of drowsiness **[36]** which helps to prevent the accidents.

In literature, many researchers have analyzed the driver's drowsiness/fatigue and they have proposed different methods based on group of metrics, furthermore, there are many cars industry which have developed their own driver's drowsiness/fatigue system to improve the quality and the security of their products and reduce the losses caused by the drowsiness. In this chapter, we will first present some of the works that have been established in the car industry. Then, we will present, analyze, compare and discuss recent related works in literature.

2.2 Industry

Currently in the automotive field, there are many companies that have developed devices used to detect drowsiness/fatigue efficiently. Also, the leading car signs have been developed from their systems to follow the driver’s condition accurately, although these companies do not state the methods used to detect drowsiness/fatigue and their accuracy in daily use, this companies have proven the efficacy of their products through user experiences.

2.2.1 Warden

Warden, a heart rate analysis-based system for driver’s drowsiness detection. Warden uses a range of sensors placed on the back of the driver's seat which analyze the changes in electric potential in the human body It detects the electrical impulses of the heart and returns the accurate R peak signal from the ECG users, which, in turn, can be used to measure the heart rate variance (HRV), which allows the device to alert the driver at an early stage of drowsiness via a smartphone app **[38].**

2.2.2 Vigo

Vigo is a Bluetooth headset that is used to measure the driver's alertness by tracking eye and head movement. When Vigo senses the driver is sleepy, it stimulates him with a mixture of pulses, music or flashing light. Since data is sent to the cloud, the device has the ability to check alert levels for any other user. The Vigo app can be used to find out the driver's alert history, directions and suggestions on how to focus during long trips **[38].**

2.2.3 Stop sleep device

Stop sleep is a device worn on finger to measures the driver’s levels of awareness and concentration continually by using 8 built-in cutaneous sensors which monitor his electrodermal activity. Stop sleep device analyze the driver’s brain activity to define his levels of concentration and awareness. if these levels start to drop, the system will alert the driver via the alarm system **[39].**

2.2.4 Leading cars landmarks

In the automotive industry field, many cars companies such as Mercedes-Benz, Volvo, Saab, Nissan, and Hyundai used the Drowsy Driver Detection System technology which can detect the driver’s drowsiness to prevent possible accidents **[38].**

Nissan

When the car speed exceeds 60 km/h, the system analyzes the driver's driving behavior in order to detect baseline-style changes using statistical analysis of steering correction errors. The system tries to adapt to each person's driving style. If the system finds that the driver is exhausted or discovers that his attention is decreasing, the information "Taking a break" appears in the car screen. The system resets automatically when the engine is turned off **[38].**

Mercedes

Mercedes Attention assist uses a sensor and detailed algorithm to detect fatigued driving behavior. Attention assist analyzes driving behavior in the first few minutes of the driver ride and assesses to his personal driving techniques using over 70 parameters. While he is driving, Attention assist identifies certain steering corrections that indicate drowsiness and fatigue. Attention assist also considers external factors, including road conditions, crosswinds, and driver interaction with vehicle controls. After considering these external factors, Attention assist will to the driver an alert that suggests him to take a break from driving if it determines that his driving behavior is due to fatigue **[40].**

Volvo

When the car is traveling above 65 km/h Driver alert is activated and still active as long as the speed is over 60 km/h. Using a camera that senses the surface lines drawn on the road and matches the lane segment with the driver's steering wheel motions. If the driver’s vehicle does not follow the road evenly it will alert him **[41].**

2.3 Scientific research works

In [36], the authors classified the recent methods for driver’s drowsiness detection into four main measures:

- Vehicle-based measures.
- Subjective measures.
- Driver behavioral measures.

- Driver physiological measures.

2.3.1 Vehicle-Based Measures

Vehicle-based measures provide a full evaluation on the driving performance by analyzing the changes in the vehicle's environment including changes in speed, steering wheel movement, etc **[42, 43].** But the changes in the vehicle's environment are often the result of the last stage of drowsiness when the driver sleeps and loses control of his car, which does not help to prevent the accident. Another problem that limits the use of this measure in real world is the difficulty of adapting these systems to the changing nature of methods **[44, 45].**

2.3.2 Subjective Measures

In the subjective measure, the driver himself evaluate his level of sleepiness. There are 2 common scales: the Stanford Sleepiness Scale (SSS) **[46, 47]** and the Karolinska Sleepiness Scale (KSS) **[46].** But because of its subjective nature, this measure is useless in realistic driving conditions. This comes in cause of asking the driver to assess his level of sleep may stimulate his awareness, and thus error in classification **[48].**

But in this work, we are interested of analyzing the drowsiness by extraction the data from the driver himself using the last two metrics.

2.3.3 Driver's behavior measures

The driver behavioral measures are mainly concerned with the extent of the driver's focus on his driving, by observing the driver's head movements, eye, yawning, or facial expression, etc. **[34, 35]** which mostly detected by image processing techniques. Although the driver's behavioral measures are non-intrusive, these measures suffer due to the instability of the vehicle's environment conditions, which directly affect the validity of these measures in detecting drowsiness **[49, 50].** For example, changing lights inside or outside the car while driving during day and night, or the use of glasses by the driver leads to reduced image processing capabilities and the effectiveness of the detection in the system **[48].**

Moreover, studies of this measure are usually tested on people trying to simulate the state of drowsiness by mimicking the movement of yawning and tilting the position of the head, which may not reflect the true state of drowsiness, which negatively affects the results **[48].**

In **Table 1** we presented and compared some related researches about how can the driver's drowsiness detected by using two major metrics: observing eye and the mouth expression. The main reason which made these two metrics a major metrics is because that the driver's behaviors can be treated and analyzed by observing the facial expressions, especially verifying if his eyes are on the wheel and see if he is not drowsy, also observing his mouth changes to detect yawning state.

Table 1. A comparison between driver drowsiness detection works using behavioral measures signals.

Ref	metrics	sensors	Classifier method	accuracy
51	Eye blinking	camera	STASM model (Based on neural network)	94%
52	Eye blinking	camera	Eye aspect ratio (EAR)	Too high (not mentioned)
53	Eye blinking features extraction	camera	Hierarchical Multiscale Long Short-Term Memory network (HMLSTMN)	Over 80%
54	PERCLOS	Camera with IR Illuminator	SVM	99%
55	Pupil (PERCLOS)	CCD micro camera +IR Illuminator	Ratio of eye-height and eye-width	92%
56	Yawning detection from the face	camera	circular Hough transform (CHT)	98%
57	yawning	camera	Ratio of mouth-height and mouth-width	94%
58	Fusion between eye and mouth detection	camera	Multi-task convolutional neural network+ Optical flow	97.06%

Discussion:

The works in **Table 1** generally gave a high accuracy results (>> 90 %, close to 100% in some cases), also these methods are the most realistic and the easiest methods to implement because they using only camera with/without IR Illuminator to capture the driver's face behavior.

From this table we can notice that analyzing the driver's behavior is determine by two major metrics: Analyzing the eye state and detecting the yawning from driver's mouth.

Analysis the eye state (eye blinking, PERCLOS, pupil analysis, eye detection models):

Many features have applied in the eye state to detect the drowsiness like detect the eye blinking **[51,52,53]** to determine if the closed eye is just blinking state or it's reflected a really state of closed eye (Lack of attention on the road). This condition is verified by analysis the average time for eye blinking by analysis eye blinking extracted from a dataset (collect of videos) and determine the threshold of blinking then in the test phase if the blinking time reach to the threshold then the system will alert the driver to get his eye on the road.

The closed eye is detected by calculating the EAR which is a constant value when the eye is open, but rapidly falls to 0 when the eye is closed **[59].**

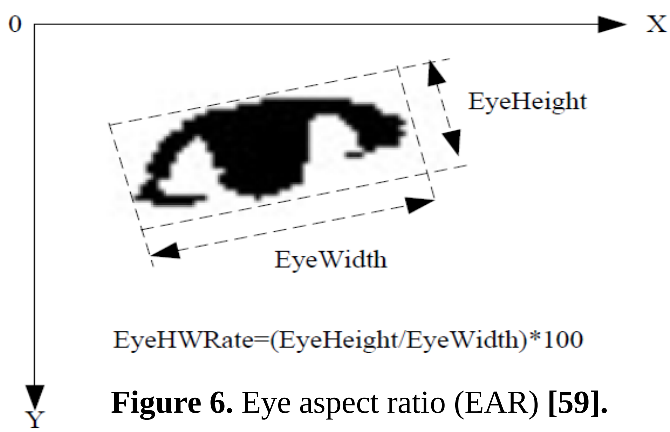


Figure 6. Eye aspect ratio (EAR) **[59].**

In **[58],** the authors can detect the closed eye by passing the eye image to image classification model to identify if the eye is open or close by extracting the features from the eyes.

The PERCLOS (Percentage of Eyelid Closure Over the Pupil Over Time) metric is used to assess the state of the pupil. PERCLOS is the mean of in the unit time the eye closed time occupies proportion. PERCLOS has been tested and proven to be the most accurate eye parameter for fatigue monitoring. Fatigue can be detected by the eye closure/opening speed. It's known as the amount of time required to open or close the eyes completely **[55].**

The PERCLOS detected by calculated the degree of opening/closing eye (EAR) per time by applying the formula (1).

$$ŋ = \frac{t_3 - t_2}{t_4 - t_1} \times 100\% \qquad (1)$$

while:

ŋ: the value of PERCLOS.

t_1: the time that eyes closed from the largest to 80%.

t_2 : the time that eyes closed from 80 % to 20 %.

t_3: the time from 20% closed to 20% open.

t_4 : the amount of time spent that that eyes open from 20% to 0%.

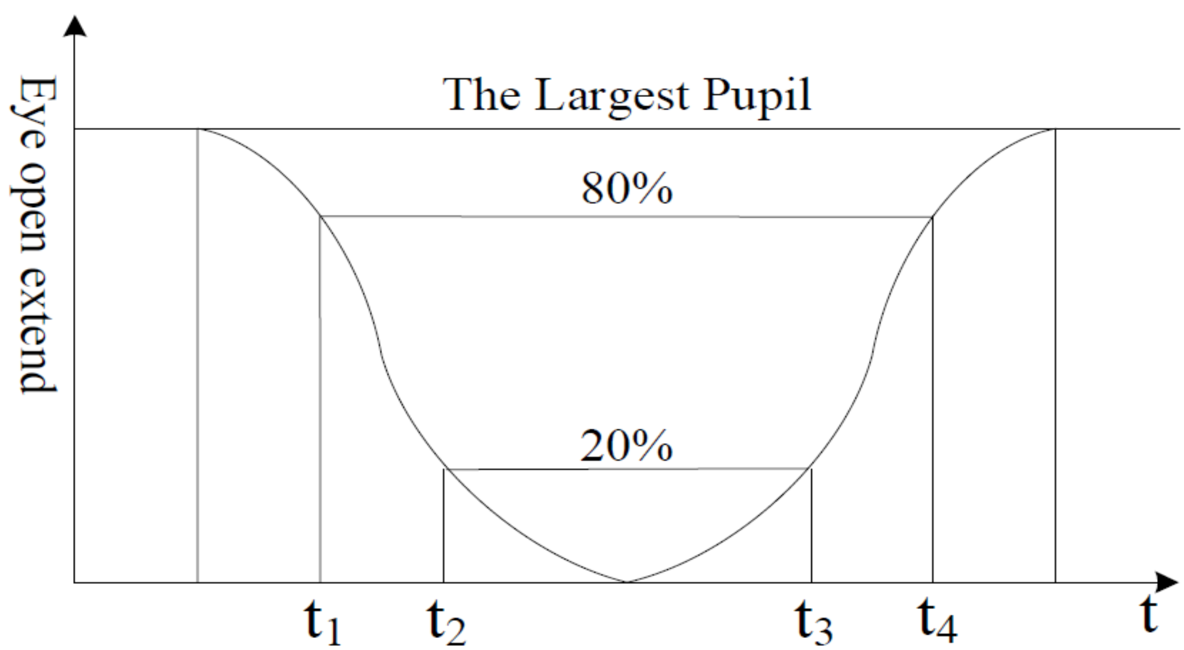


Figure 7. PERCLOS principle **[55].**

a) Detect the yawning from driver's mouth:

Some works as it shown in **Table 1** have determine the drowsiness and fatigue based on yawning detection by detecting the mouth aspect ratio (MAR) which mean the distance between the two lips or verify if the system detect a blob hole in the mouth region. But in this metric, there is one big problem which is:

Usually When the driver yawning, he put his hand on his mouth?!

But in **[57]** the problem was treated by a continuous analysis of images frames and see if after the system was detecting the increase of opening mouth, suddenly the system does not give any value of mouth distance (no mouth detects) then it will presume that there is a yawning state.

Finally, we notice that in the driver's behavior measures, all previous work suffers from poor night lighting or lack of lighting, and as a result, the face/eye/mouth detection will not work properly.

2.3.4 Driver Physiological Measures

The previous measures can only be used after the driver starts to sleep, unlike this measure, which can prevent an accident because the ability of physiological signals to detect drowsiness in its early stages **[36].**

This last category of measures focuses on analyzing the physiological signals of the driver (Electrocardiogram (ECG), Electromyogram (EMG) Electroencephalogram (EEG) and Electro-oculogram (EOG)) to detect the drowsiness and provide an accurate measure of drowsiness because of the strong relationship with driver fatigue **[36,37].**

Some researchers have used the EOG signal to identify driver drowsiness by analyzing the electric field generated by the electric potential difference between the cornea and retina. This electric field represents the orientation of the eyes, which allows to detect any drowsiness **[36].**

EEG is a common physiological signal that measure the driver's drowsiness. The EEG signal has various frequency bands, including the delta band (0.5-4 Hz), which corresponds to sleep activity, the theta band (4-8 Hz), which is related to drowsiness, the alpha band (8-13 Hz), which represents relaxation and creativity, and the beta band (13–25 Hz), which corresponds to alertness

[36]. There some works that have used the EEG signals to detect the drowsiness **[34,37].** For example, Picot, A et al. used the **α** and **θ** and **β** activities of the EEG signals extracted from the driver's brain in order to classify his state to one of this three cases: "awake, drowsy, very drowsy" **[60].** Also, ECG signals can be used in order to detect drowsiness by analyzing the HRV extracted from the ECG sensors **[61,62,64].**

The physiological signals are more accurate and suitable to detect the drowsiness with less error rate because the physiological signals start to change in earlier stages of drowsiness, this gives the driver a good time to alert him and thus may prevent the road accidents **[36].**

In general, the physiological signals firstly go through a preprocessing phase which used to filter the entered data from the artefacts and noises of the signals in order to improve the results and eliminate the unimportant data.

Table 2. A comparison between driver drowsiness detection works using physiological signals.

Ref	**Sensors**	**Classification method**	**Classification accuracy**
69	EEG, EOG	Fuzzy logic	80,6% (20 subjects)
62	ECG sensor	SVM, K-nearest neighbors (KNN), Ensemble classifier	From 58% to 100%
63	EOG, EEG, ECG	Case-based reasoning (CBR), SVM, Random Forest	From 72% to 93% (30 subjects)
64	ECG sensor	ANN	90% (12 subjects)
65	EEG, ECG, EOG	Linear Discriminant Analysis (LDA), LIBLINEAR, KNN, SVM	95–97% (31 drivers)
66	EEG, EMG	ANN +Back Propagation Algorithm (awake,drowsy, sleep)	98–99% (30 subjects)
67	EOG, MG	SVM	90% (37 subjects)
68	EEG, EOG, EMG	ANN	97–98% (10 subjects)

The **Table 2** represents a comparison that we have made between some studied works related to driver drowsiness detection using physiological signals.

Discussion

In general, these works have a high accuracy and stable results in most conditions, also the works that uses two or more sensors give higher accuracy results compared to the works that uses only one sensor. It can also be seen that the size of the dataset used in the model is a main factor for the results quality where if the dataset contain data for more subjects or the datasets collect more data on driver's state, the results quality will be significantly higher.

In **[62]** the authors did a comparative works of detecting the drowsiness using the ECG signals. They used in their work 3 classifiers: SVM and KNN and Ensemble classifiers to detect the drowsiness while the last classifier gave the better results. In part of classify sleepiness, the authors used 2 types of classification, the first divides the sleepiness into 2 classes (normal‑drowsy, normal‑visual inattention, normal‑fatigue and normal‑cognitive inattention). The second is used a fusion of the 5 classes. The results show that the 2 classes division gives a higher result (In normal-drowsy case the accuracy results was 100%) While the fusion of 5 classes gives a worse result (provides an overall accuracy of 58.3%). In **[63],** the authors used the three classifiers which are SVM, CBR, Random forest to find the best method that gives a higher result accuracy which was the SVM classifier with 93 % for the binary classification(alert, sleepy) and 79% for multi-class(alert, sleepy, somewhat sleepy) while subject-dependent classification exhibited a 10% improvement in the system performance compared to the subject-independent classification. Also, they have embedded a contextual information, and that improved the classification performance by 5%. Besides, they have analysis other driver's states such as stress and cognitive states.

The authors in **[60]** used EEG signals to obtain the brain information by analyzing the increase of **α** and **θ** activities that characterize the drowsiness case and analyze the decrease of **β** activity which is linked to active concentration. They also used the EOG signals to obtain the visual information about the blinking analysis. They use in their system 3 cases of driver's state (awake, drowsy, very drowsy). And to analyze and extract the driver's state they use a fuzzy logic based on merge of medical rules for the two physiological signals. The obtained results were about 80% which is a lower result compared to the other works.

In **[64],** the authors have proposed an approach that used the ECG sensor to analysis the HRV (Heart Rate variability). Before the model analysis the ECG signals, the data go through a filter phase in order to eliminate the artifacts and noises in the signals (< 5Hz and > 40 Hz ➔ Ignore), then this filtered data enter to the ANN model as a input layer and go to a single hidden layer to determine the state of the subject (output), The ECG signals are enough-as they said, however the accuracy of this method is low compared to the others because they using a limited data for the prediction (12 subjects) so their model is insufficient for all cases.

In **[68]** Artificial neural network was applied for detecting the state of a subject. Delta and Theta and Alpha (EEG sub bands) and LEOG and REOG and EMG was considering as an input layer of this model, also their model use 2 hidden layers to analysis the signals to determine the state of the subject (“awake”, “sleep”, ”drowsy”).Their model have a high accuracy caused by the using of multiple physiological sensors .Also they use of filter of artifacts and noises to extract the important data for the model .

The authors in **[66]** had also used the ANN based approach, their work is much similar to the work in **[68]** but with a higher accuracy results because they collected more data (30 subjects instead of 10).

But even if these methods have proven relatively effective for analyzing the physiological states of driver, the problems of wearbility, have limited the feasibility of using these systems in real-world driving conditions **[69].**

2.3.5 Hybrid measurement

The hybrid between the driver’s behavior and the physiological signals is a good solution for driver’s drowsiness detection, and if it is implemented perfectly, it will give a high accuracy then using one of each metric. There are some works have used the concept of hybrid measures to detect the drowsiness **[70, 71 and 72].** For example, Cheng et al. combined between the behavioral and the vehicle-based measures and the result of the hybrid method was significantly higher than those results using single sensor **[72].** Guosheng et al. have mixed the subjective measures with the behavioral measures using PERCLOS metric with the physiological measures by analyzing the driver’s ECG and EEG signals to detect the drowsiness, and they found that the obtained results were higher success then the results obtained from the use of the measures individually **[70].**

2.4 Best implementation

establish the best implementation of the driver's drowsiness and fatigue detection system. To do, we need to compare between all the analyzed metrics (driver's behavior, physiological signals, hybrid measurement) for the same environment and for the same data.

We have seen that some of the previous methods are used in the automotive industry. While each measure has its own weaknesses, making it incomplete solutions for use in the real world. But it seems that the concept of hybrid measures is a good solution to achieve the best possible performance because the detection of drowsiness in the real world needs to use non-intrusive acquisition methods.

Considering this, L. Oliveira and al. **[73]** have compared drowsiness detection performance using a combination of a non-intrusive camera-based method with an EOG-based physiological method and an ECG signals video to obtain the best possible scenario by combining multiple data sources. In part of data, they have used in their work data from a real road experiment with sleep deprived drivers In order to classify sleepiness, the dividing cases of sleepiness levels idea was inspired by the 9-point Likert scale of the KSS scale to divide the problem into awake and drowsy cases in the binary division and for the multi class (3 classes) division they uses 3 cases which are awake and medium and drowsy **[73].**The division of sleepiness levels for both cases are displayed in the **Figure 8.**

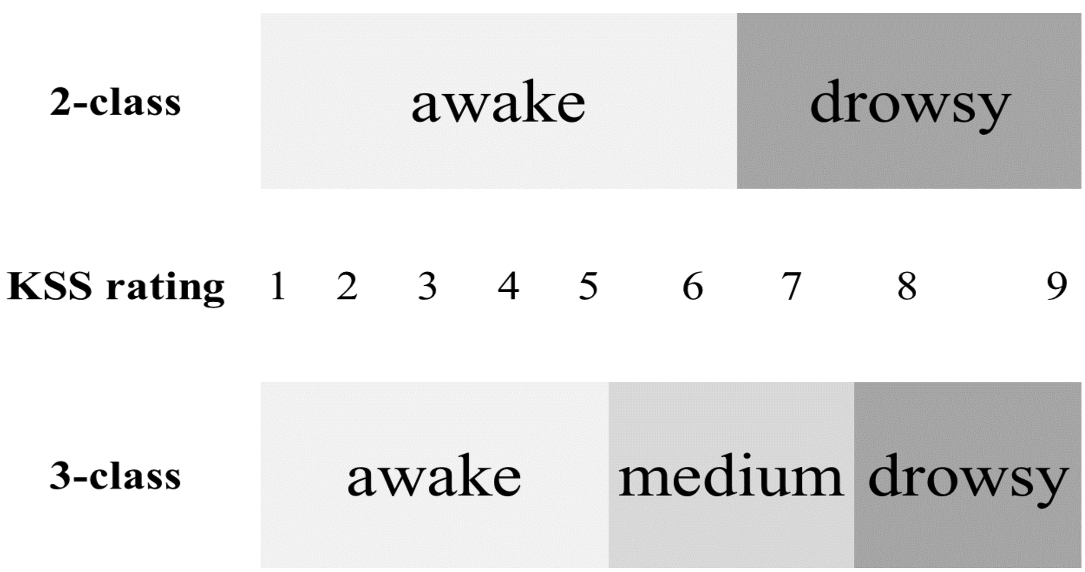


Figure 8. KSS rating and their corresponding states for 2-class and 3-class **[73].**

a) The binary division case (Awake, Drowsy)

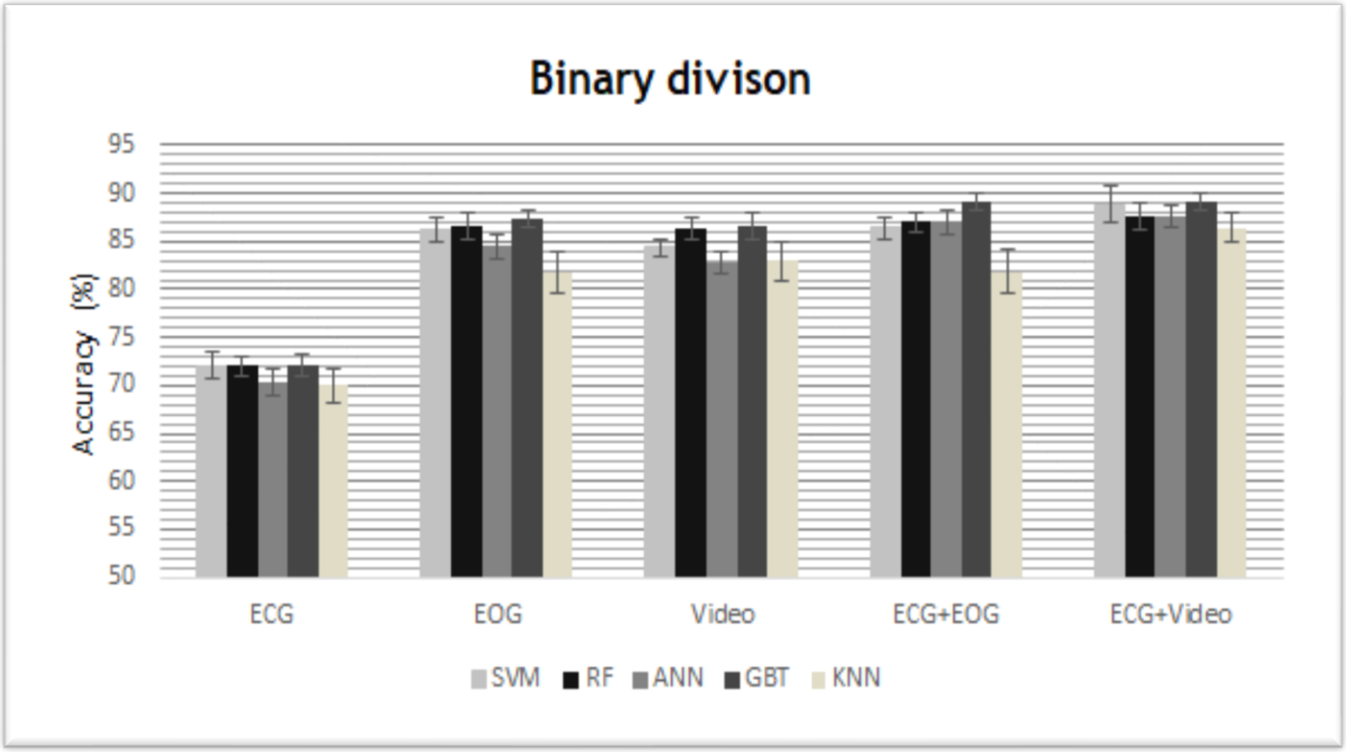


Figure 9. Accuracy results (average and standard deviation) for all classifiers for binary division case **[73].**

Discussion:

It is recognizable that the ECG alone shows poor performance **Figure. 9.** Also, the overall performance obtained using EOG features is similar to the performance using video features. The same observation can be made when comparing ECG+EOG and ECG + video features. There is a slight improvement when using a hybrid approach including ECG features in addition to the EOG or video features. Regarding the different classifiers, there are some minor differences in their performance’s; however, no classifier seems to be far superior to the others.

b) **The Multiclass division case (Awake, Medium, Drowsy)**

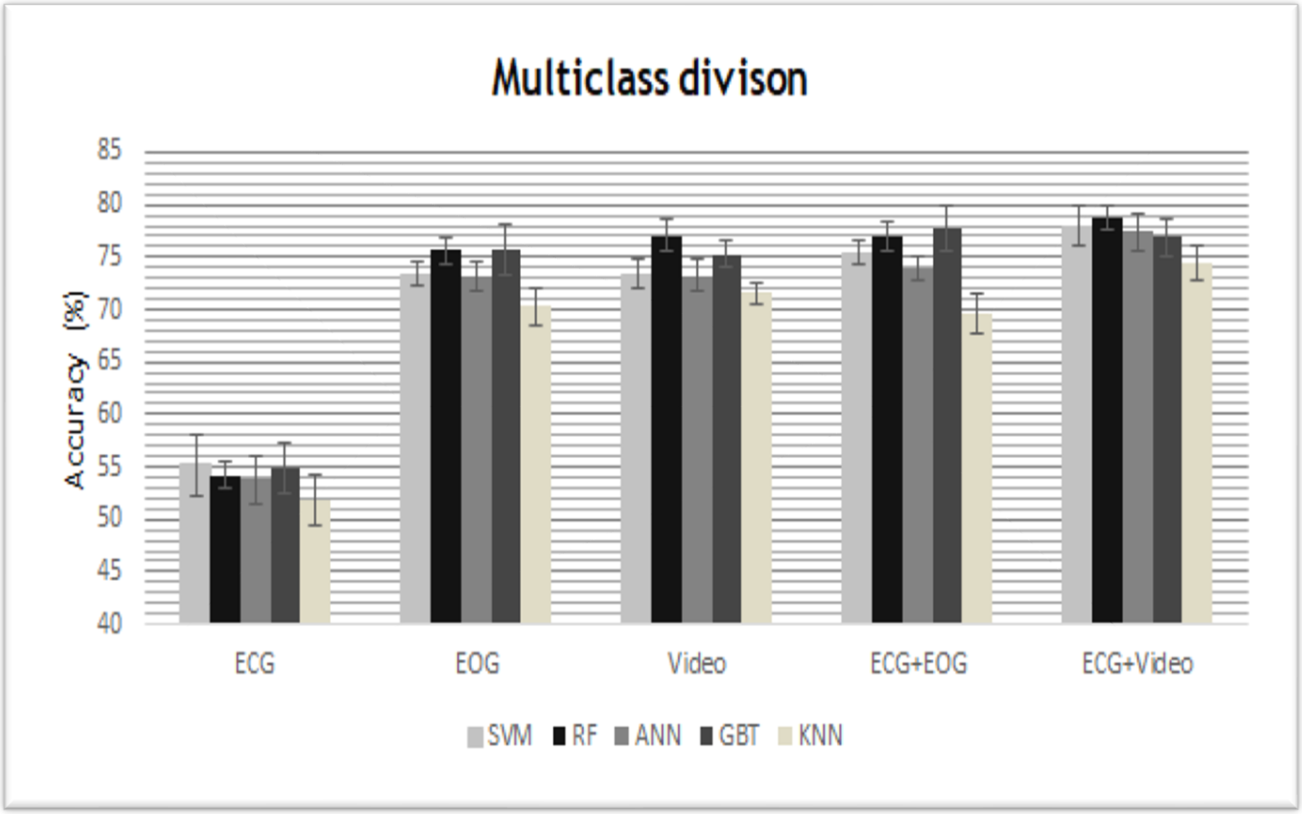


Figure 10. Accuracy results (average and standard deviation) for all classifiers for multiclass division case **[73].**

Disscucion:

In the multi class division case the results obtained by applying the same methods as the binary case are displayed in the **Figure 10.** In this case, the same observations of the binary problem division can still be applied; but the overall accuracies are obviously lower compared to the previous case (about 10%).

2.5 Conclusion

In this chapter, we have presented a review on different measurement approaches of driver's drowsiness and fatigue detection systems. Then, we performed an analytical comparison of those approaches. We noticed that methods for measuring the driver behavior give high-resolution results in the case of good lighting but these results become less good in the light of night because it suffers from the detection of facial expressions. On the other hand, the physiological signal measurements give a stable result in most cases. The only drawback of this measure is that it is an intrusive method for the driver.

Despite the difficulty of implementation and the requirements of much data to be collected for both measures. The hybrid metric appears promising to remedy the flaws of the two measures, and may yield impressive results.

Chapter3:
System Design

3.1 Introduction

After analyzing the recent methods used to detect sleepiness, we will present in this chapter our developed system. We start by explaining the principal architecture and all its components. After that, we will introduce the methods that we used to detect drowsiness and discus about the choices used in our system.

3.2 System architecture

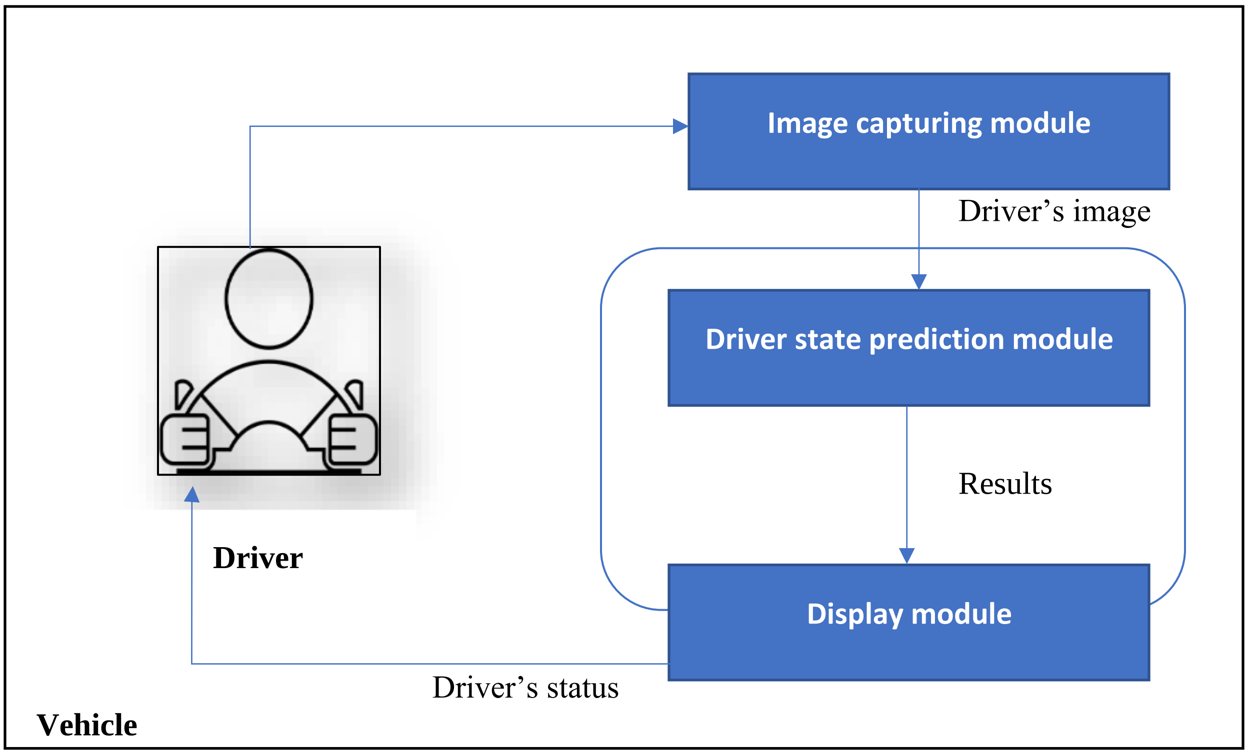


Figure 11. System architecture.

From the previous figure, we can see that we used an embedded system hidden in the vehicle (no cloud user part) where all actions take place within the vehicle's environment. The system consists of three main modules: the image capture module that captures the image of the driver who sits behind the steering wheel to send the image to the second module, which is the driver's state prediction module, which can determine the driver's condition to send the obtained results to the last module, which is the display module that displays the driver's general condition. In a simple form that can be read and understood easily.

3.3 System components

3.3.1 Driver

The driver represents the person who sits behind the steering wheel who controls the management of the car and is the person whose condition is to be examined to prevent any possible accident resulting from any drowsiness or fatigue that may affect him.

3.3.2 Image capturing module

The only role of this unit is to constantly take pictures of the person sitting behind the wheel (the driver) and then send the pictures to the next unit (driver's state prediction module).

3.3.3 Drivers' state prediction module

Once the image capturing module sent the image to the recurrent module, this last start to identify the driver's state using the methods disused in the previous chapter.

3.3.3.1 Driver’s state prediction module architecture

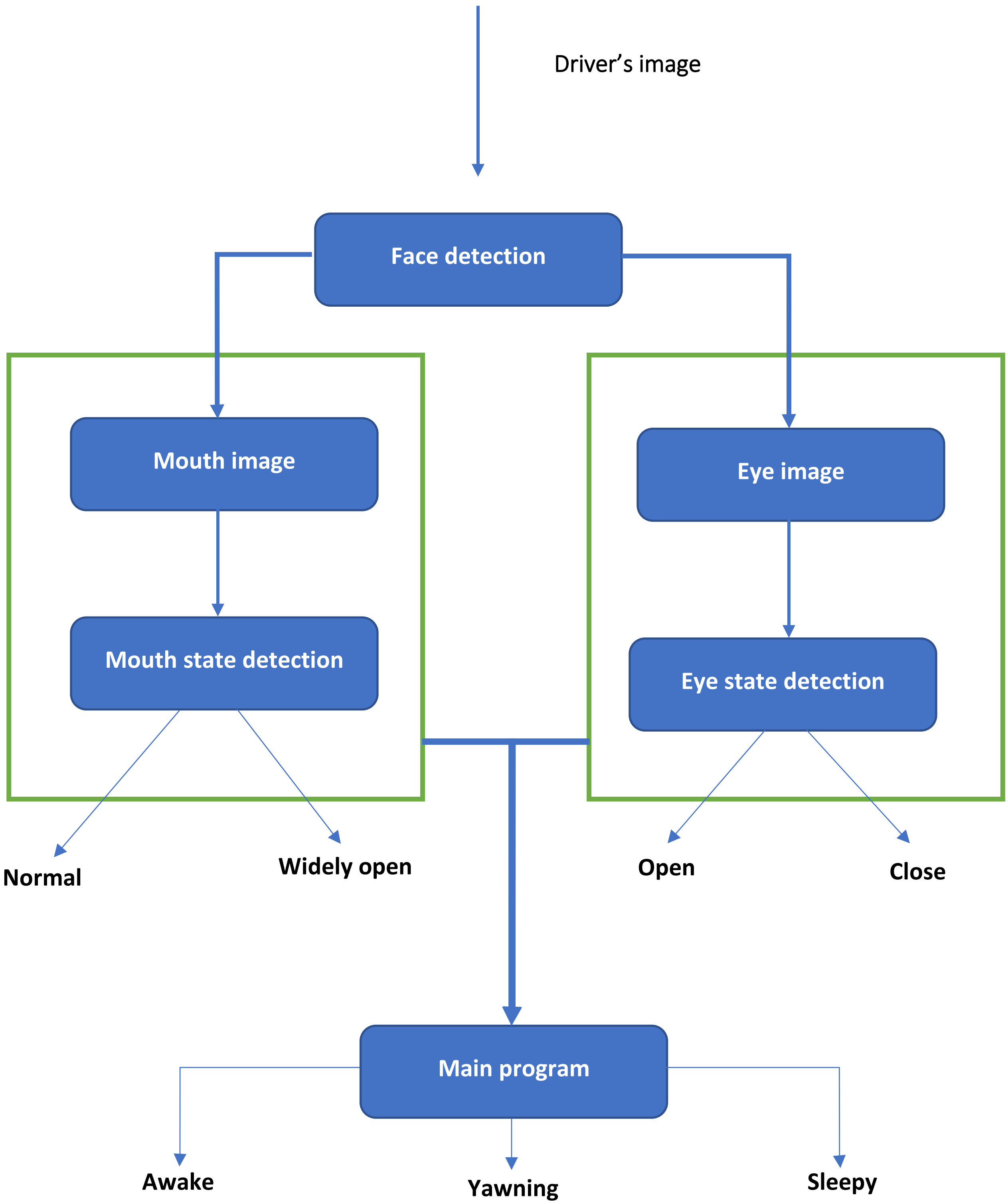


Figure 12. Driver's state prediction module architecture.

The previous figure shows the architecture of the driver’s condition prediction, where we can see that after obtaining the captured image from the previous unit, this process begins with the face detection phase to give as a result the precise location of the driver’s conscious mouth, so that these results are sent to the next two stages. Also, we can see that our system detects the driver’s condition based on the driver’s behavioral metrics (detection of eye and mouth condition). Oral condition to spot any potential cases of yawning that may help determine the driver's exact condition. As a result of this stage, the system gives two possible classes of oral condition which may be normal or open wide. Ultimately, the main program collects the results of both stages to determine the driver's overall condition, which may be one of the three categories (Awake / Yawn / Sleepy).

3.3.3.2 Driver’s state prediction module phases

3.3.3.2.1 Face detection phase

After obtained the images captured from the previous module, the face detection start works to find the exact location of the driver’s face, and the location of the mouth and eyes inside the driver’s face.

In our system, we need to get this result from the face detection model as it’s shows in this figure. Our desired model must get the image of the driver’s face as an input and gave the mouth and eye sub images as an output.

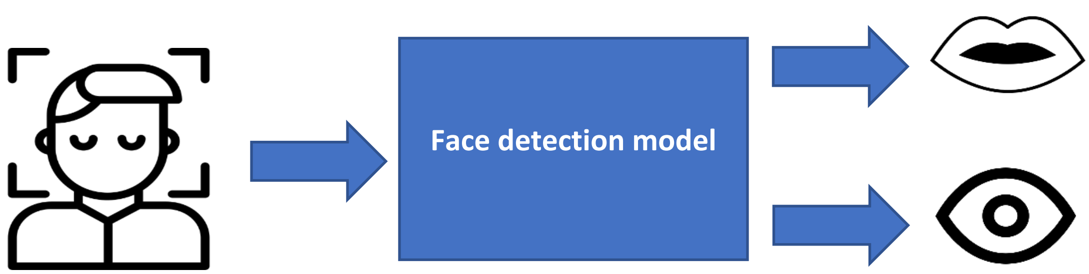


Figure 13. Face detection phase process.

There are many models and techniques to detect the face with a high accuracy in short time. But in our system, we need to find the best technique which can detect the face and find inside it the eyes and the mouth localization in real time (leger technique), so because of that we have decide to use the facial landmarks model to extract the 68 facial landmarks points as shown in the algorithm below.

Algorithm 1 Face detection

Input: *model* : The 68 facial landmark model
Input: *frame* : The captured image
Output: *Array* (The 68 driver facial landmark points coordinates(x,y))

```
Array_X ← ∅
Array_Y ← ∅
Faces ← model.detect faces(frame)          ▷ Find all faces inside the frame
if length(Faces)== 0 then                  ▷ No face detected Case
    return ∅
else                       ▷ Case where there is at least one face detected
    Driver_Face ← ∅
    Max_width ← 0
    for Face in Faces do                   ▷ Find the driver's face
        if Max_width > Width(Face) then
            Driver_Face ← Face
            Max_width ← Width(Face)        ▷ the driver has the max width.
        end if
    end for
    landmark_Points_Array ← model.detect_Points(Driver_Face)
    for point in landmark_Points_Array do
        ▷ Get all the 68 coordinates (X,Y) of the driver
        Array_X .add(point.X)
        Array_Y .add(point.Y)
    end for
    return (Array_X,Array_Y)   ▷ Return the (X,Y) coordinates of the 68
points of the driver
end if
```

Algorithm 1. Face detection Algorithm.

The main reason that we have chosen the Facial landmark model is:

Facial landmark detection algorithms aim to automatically identify the locations of the facial key landmark points on facial images or videos. Those key points are either the dominant points describing the unique location of a facial component (e.g., eye corner) or an interpolated point connecting those dominant points around the facial components and facial contour. Formally, given a facial image denoted as I, a landmark detection algorithm predicts the locations of D landmarks $x = \{x_1, y_1, x_2, y_2, ..., x_D, y_D\}$, where x_i and y_i resentment the image coordinates of the facial landmarks **[76].**

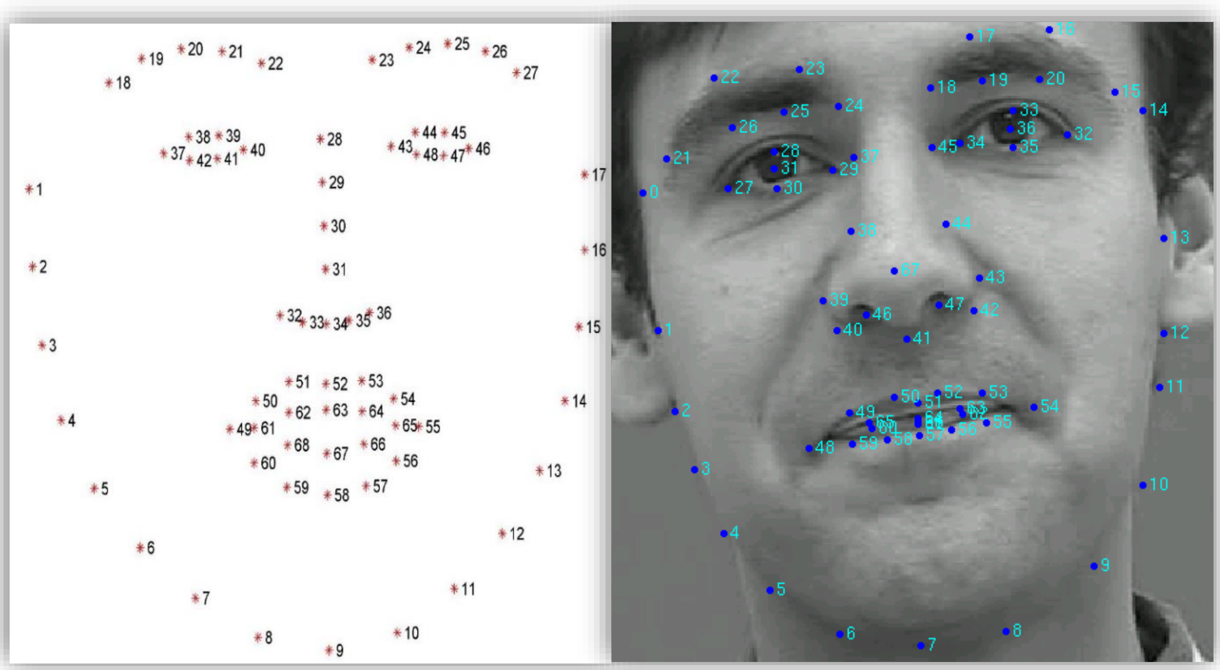


Figure 14. The 68 points of the facial landmarks model **[75].**

3.3.3.2.2 Eye state detection phase

After finding the eye zone location, the system detects the eye state using a CNN model for image classification to identify the eye state(open/close).

There are many methods and techniques to verify the eye state to check if the driver focuses at the road or not, like using a prediction models or calculate the eye aspect ratio which is a constant value when the eye is open. But rapidly falls to 0 when the eye is closed (previous chapter) to distinct between the blinking and the sleepiness states of driver.

In our system we have develop a CNN model for image classification for the eye image of driver. The main reason that we have choose a CNN technique is:

The CNN performs better in image classification and identification purposes because taking matrix as input reduce the number of parameters and increase the reusability of weights [77].

Also, the usage of the methods that verified the eye state by calculating the EAR or analysis the PERCLOS have problems especially at the night where the eye movement could be very difficult to analyses and the face detection model will have problems to find the location of the eye.

In our system we have built a typical CNN model with a sequence of layers which are:

- Convolution layer with F1 filters and “ReLU” activation function.
- Pooling layer.
- Convolution layer with F2 filters and “ReLU” activation function.
- Pooling layer.
- Convolution layer with F3 filters and “ReLU” activation function.
- Pooling layer.

Then we flatten the outputs of the previous layer (pooling layer) to a single Dense layer with “ReLu” activation function. As an output layer, we have used a Dense layer with Sigmoid activation function with 1 class (Binary classification refer to eye is opened or not).

Our CNN model is a binary image classification model, that’s mean that it will give us as a result one of two states of the eye: open or close.

The architecture of our model is represented in the next figure **(Figure 15).**

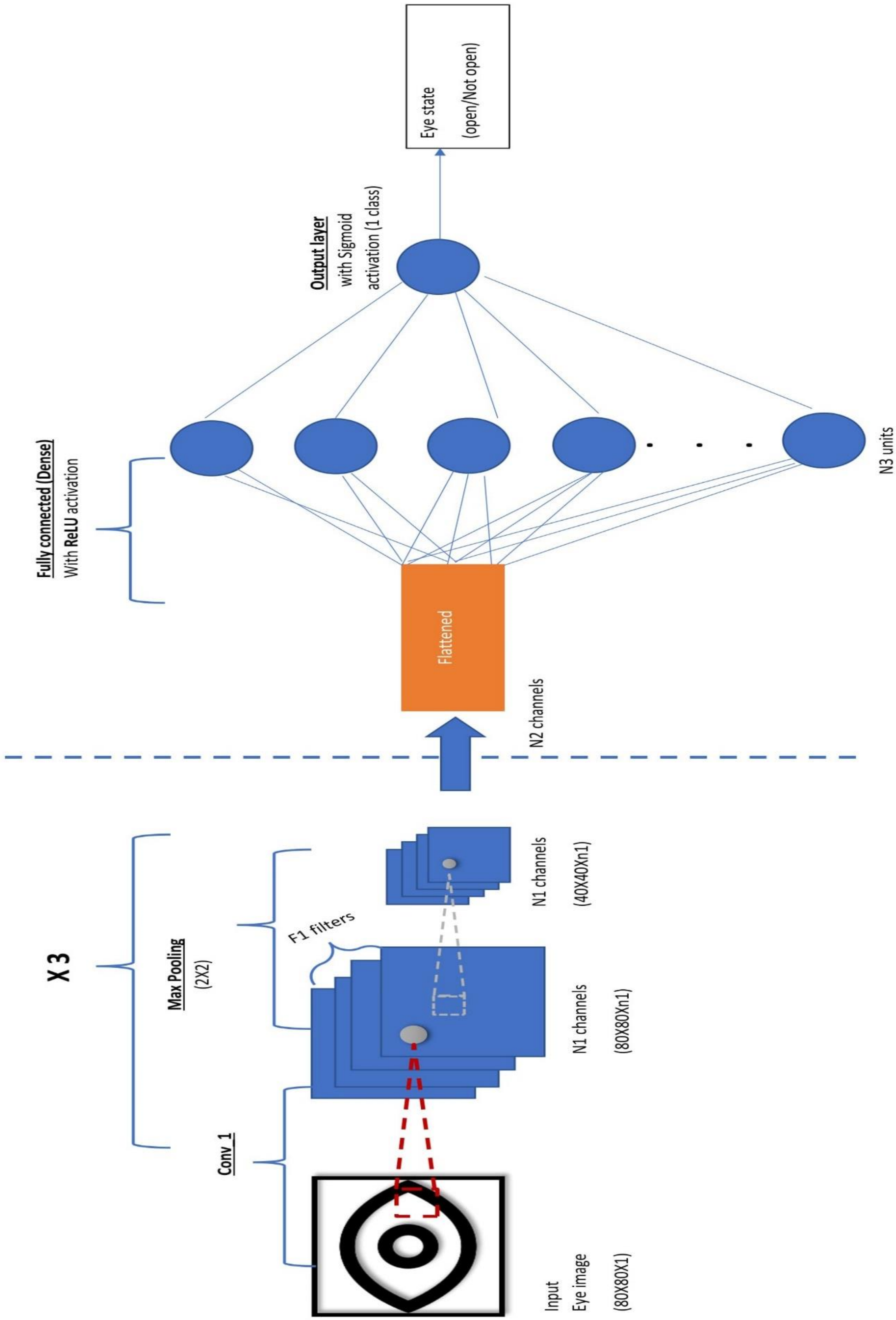


Figure 15. CNN model Architecture for eye state.

3.3.3.2.3 Mouth state detection phase

At this point, we determine whether the driver's mouth is wide open (yawning) or in the normal state (speaking / singing / closing) by calculating the distance between the lips to check if the distance reaches a certain point that indicates a yawning case.

After finding the exact location of the mouth, the yawning can be detected by calculated the mouth width and height by horizontal and vertical grayscale projection of the mouth area **[78].**

Ashlesha Singh and al say that yawning can be detected from the MAR which can calculated by computing the Euclidean distance between the landmark which can determine the yawning state if this value reach to a certain threshold **[79].**

So, in general, the yawning can be detected by checking if the mouth is just widely open (reach to a certain threshold). For that we detect the yawning using the process shown in the following figure **(Figure 16).**

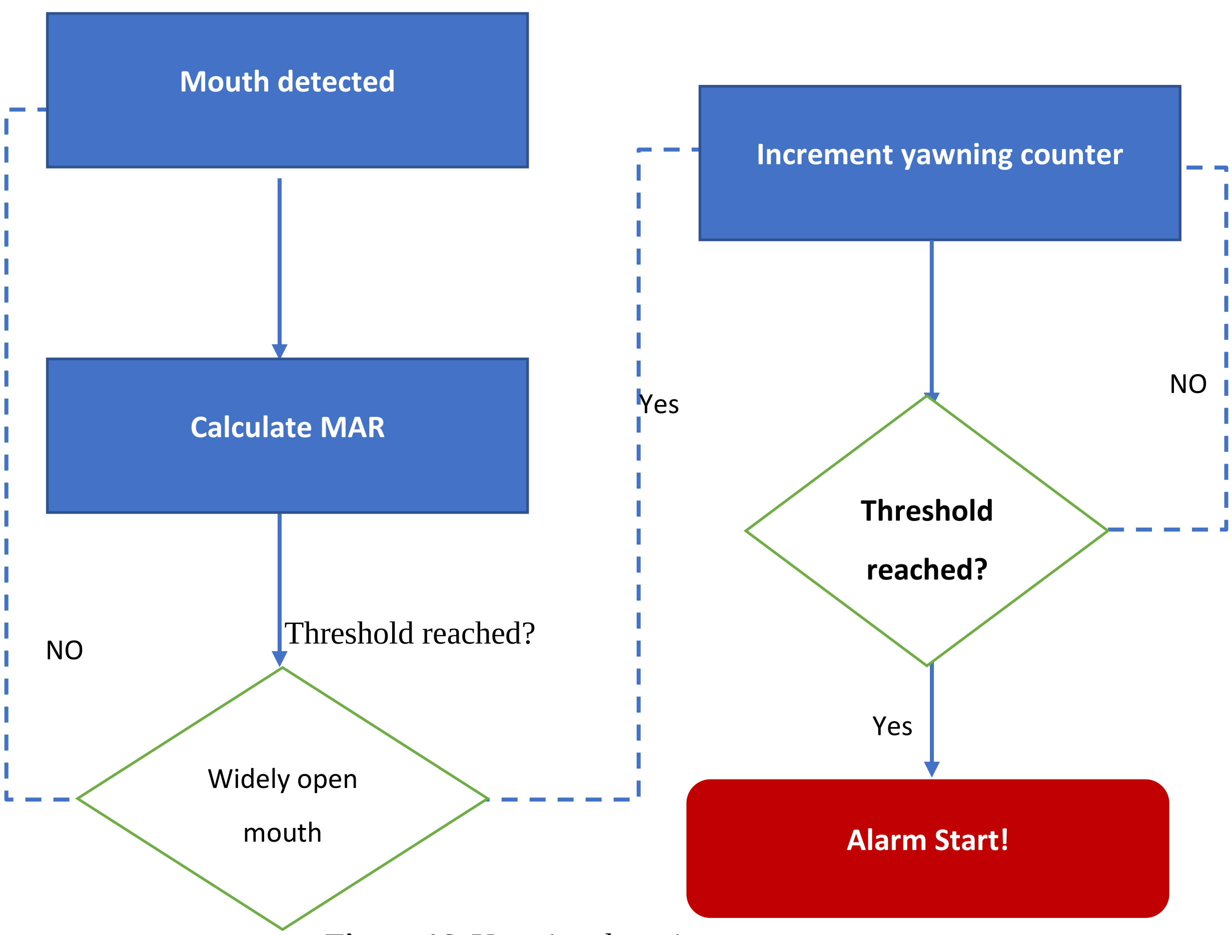


Figure 16. Yawning detection process.

In our process, after we find the location and extract the mouth image, we calculate the MAR which can be defined by calculating the Euclidean distance between the two lips like it's explained in the **Figure 17 [80].** Then we compare this calculated value with a fixed value called "yawning threshold". The yawning threshold is a fixed value that which calculated and fixed after analysis a yawning dataset to see the average value (distance between lips) of the widely open mouth which lets us to distinct between the talking or singing or closing mouth states, also we calculate another value called "yawning threshold time", the yawning time threshold refer to the average time that the mouth tacked while yawning.

So, if the MAR reach to threshold then we will increment the time value until we reach to the yawning threshold time (Yawning state➔ Alarm start) or the MAR will decrease above the threshold before reached to time yawning threshold (Return to the first step).

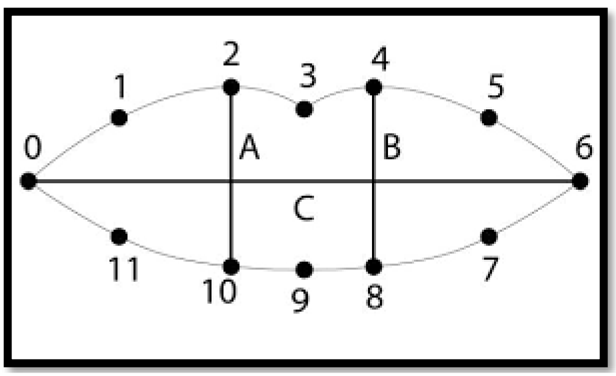


Figure 17. Mouth region with reference coordinate points **[80].**

A=d (Mouth [2], Mouth [10])

B=d (Mouth [4], Mouth [8]

C=d (Mouth [0], Mouth [6])

d is the Euclidian distance between two points.

A and B measure the vertical distance the eye and C calculates the horizontal dimensions of eye.

$$MAR=(A+B) / (2.0 * C) \qquad (2)$$

So, in general we have calculating the MAR using the following algorithm.

Algorithm 1 Calculate MAR

Input: $facial\_Points\_Array$: Array of the mouth region points obtained from the face detection phase *(Points from 49 to 68)*
Output: Mouth aspect ratio value

$Dist\_A \leftarrow$ euclidean_Distance($facial\_Points\_Array[2]$, $facial\_Points\_Array[10]$)
$Dist\_B \leftarrow$ euclidean_Distance($facial\_Points\_Array[4]$, $facial\_Points\_Array[8]$)
$Dist\_C \leftarrow$ euclidean_Distance($facial\_Points\_Array[0]$, $facial\_Points\_Array[6]$)
$MAR \leftarrow \dfrac{(Dist\_A + Dist\_B)}{2 \times Dist\_C}$ ▷ *MAR rule*
return MAR ▷ *Return the Mouth aspect ratio value*

Algorithm 2. Calculate Mouth Aspect Ratio Algorithm.

Algorithm 1 Mouth state detection

Input: $MAR\_threshold$: the average MAR value reached by the mouth to distinguish between the normal and yawning state
Input: $MAR\_Time\_Threshold$: the average time taken by the mouth when it's open
Input: m_Frame : the mouth image at the current time
Output: $Mouthstate$ (Yawn/Normal)

if Calculate_MAR(m_Frame)== $MAR\_threshold$ **then**
    $Still\_Open \leftarrow True$ ▷ *mouth state(open/Close) checker*
    $time\_Open \leftarrow 0$ ▷ *Counter for mouth open time*
    $m\_Frame{+}{+}$ ▷ *Jump to next image*
    while $Still\_Open$ **do**
        if Calculate_MAR(m_Frame) $< MAR\_threshold$ **then**
            $Still\_Open \leftarrow False$ ▷ *Case where the mouth begin to close up*
        else ▷ *Increment mouth open time counter if mouth still open*
            $Time\_Open{+}{+}$
        end if
        $m\_Frame{+}{+}$ ▷ *jump to next image*
    end while
    if $Time\_Open \geq MAR\_Time\_Threshold$ **then**
        return $Yawn$ ▷ *Yawning state detected*
    else
        return $Normal$ ▷ *Mouth does not reach to time threshold*
    end if
else
    return $Normal$ ▷ *Mouth in normal Case*
end if

Algorithm 3. Mouth state detection Algorithm.

Using the MAR algorithm (**Algorithm 2**), we have developed our own algorithm to detect the mouth state as it shows in previous algorithm (**Algorithm 3**).

3.3.3.2.4 Main program

This is the last phase of our system, it gathers all the results obtained from the last two phases (eye state detection, yawning detection), then the system will decide the driver's state based on using multiple metrics (Eye state, MAR,EAR) to classify the state into one of 3 classes which are:

Awake: the normal condition where the driver is noticing on the road (eyes open / not yawning).

Yawn: If the driver yawns while driving, the system will ask him if he is feeling sleepy or not and start closely monitoring his condition (by analyzing the aspect ratio in his eye, PERCLOS analysis, etc).

Sleepy: If the system detects that the driver is not paying attention on the road (closing his eyes frequently or for a long time, moving his head a lot), the system will start trying to wake him up by triggering an alarm siren to ask him to stop aside to rest a little.

3.3.4 Display module

After the Driver Condition Predictor determines the general condition of the driver, it sends the obtained results to the display unit in abstract form. The main role of this unit is to convert these results from their abstract form (the logical variables of the eye condition, the real values of the mouth state) into a graphic form that is easy for the driver to read and understand.

3.4 Discussion and argumentation

We use an embedded system in the car to detect the drowsiness instead of using a connected system because in **[74]** the author said that:

- Algeria greatly lags behind in access to technological resources including the internet. In the region of North Africa, Algeria is ranked behind Morocco and Tunisia. Indeed, the internet penetration rate is estimated to be 14% while neighbors Morocco and Tunisia are respectively about 51% and 39% according to a case study conducted by Google North Africa in 2013. This lack of penetration is explained by the vast space of the country compared to its neighbors.

- The infrastructure of the internet in Algeria is weak and requires many things to make the situation better for Algerian internet users. This poor quality of internet connectivity is due mostly to selling the same bandwidth to more than 40 families at the same time, which causes major online blockage and traffic. Users wonder why they are getting such a slow internet connection even if they registered for better service. By this logic, it became impossible to give them good internet service.

- This situation has created a huge delay in online services as e-commerce and e-payment have not been implemented yet. Despite the existence of many e-commerce sites in Algeria like Echrily.com and Guiddini.com, traditional methods for payment by bank transfer, cash on delivery or even by cheque are used.

- A recent official report published in Algerian newspaper “Echourok” disclosed that Algeria telecom was unable to make a global and obvious plan for fiber optic utilization and exploitation in a better way or to raise the capacity and the speed of the local and international access to the Internet. Algeria is considered to be one of the countries that has the slowest internet connectivity in the world, ranked 179th with a rate of 3.3 mbps/sec, according to a net index website.

So, it is difficult and risky to use a connected system to detect the driver’s condition due to the poor internet service in Algeria and its failure to cover all regions.

Also, we have combined between the eye state analysis and yawning metrics to get more information about the driver’s state.

3.5 Conclusion

In this chapter, we introduce the concept and design of our system by introducing the system architecture and discussing the main components of this system.

We also introduced the methods used in our system, which are methods based on the driver's behavioral measures that can detect drowsiness by analyzing eye and mouth conditions to classify the driver's condition within the three categories we used (Awake / Yawn / Sleepy).

In the end we explain why we chose an embedded system and why we used CNN models rather than other methods to detect eye condition.

Chapter 4: Implementation and results

4.1 Introduction

After analyzing and introducing the concept of our system in the previous chapter, we have developed a prototype capable of detecting driver's drowsiness/fatigue. Therefore, in this chapter, we will provide the required materials, tools and platforms used to build and develop our prototype. We will also show how we implement the system concept on the ground, and then we will discuss the results obtained. In the end we will show some pictures of our developed system.

4.2 System overview

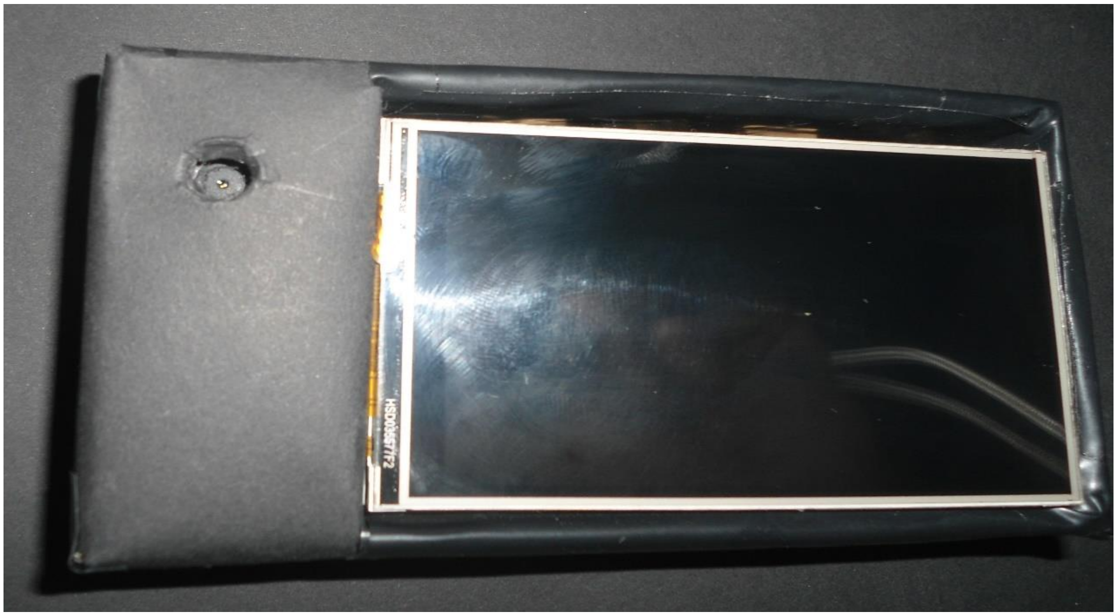


Figure 18. System prototype.

This figure shows the prototype of our system, where this system is characterized by its simplicity as it consists of a camera used to capture the driver's movements (on Top left) in addition to the screen used to display the driver's condition. The system is also characterized by being lightweight, easy to carry and portable.

4.2.1 Hardware

In this section we will present the required equipment that we will use in our system.

4.2.1.1 Raspberry pi

We have used in our system the raspberry pi 3 model B vol2 where the Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python [81].

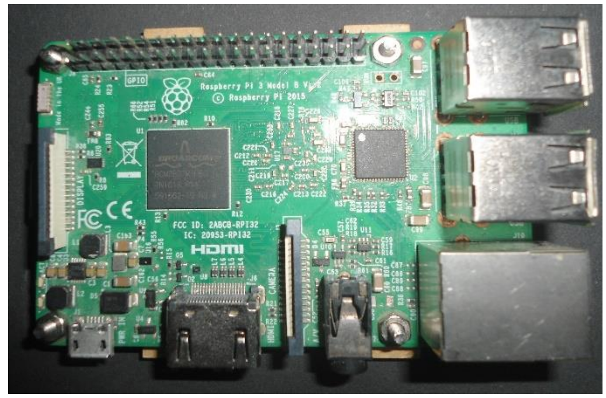


Figure 19. Raspberry pi 3 model b vol2.

4.2.1.2 Pi Camera module

Also, we have used a 5 MP pi camera version 1.13 as an image capturing module to detect the driver’s movements. This pi camera features a 5MP (2592×1944pixels) Omni vision 5647 sensor, also it supports 1080p/30fps, 720p/60fps and 640x480p 60/90 video recording.

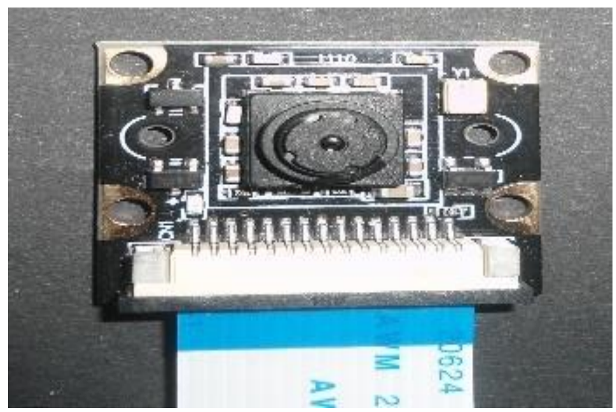


Figure 20. Pi Camera module.

4.2.1.3 Raspberry pi LCD Touchscreen

For the Display module we have choose to use a 3.5-inch raspberry pi LCD touchscreen with resolution of 320*480 dots.

Figure 21. Raspberry pi LCD touchscreen.

4.2.2 System characteristics

Property	**Description**
OS	Raspbian 2020 (Buster version)
CPU	ARM Cortex-A53 Quad-Core 1.2 GHz
RAM (GB)	1
Storage (GB)	16
Camera	5 MP (2592×1944pixels) Omni vision 5647 sensor

Table 3. System characteristics.

The table shows the characteristics of our system. As we can see, our system has limited and weak resources, and this may negatively affect the effectiveness of the system. All these obstacles have made it necessary for us to work on improving our system in order to address these problems by working to exploit these resources in the best possible way. Also, using the pi camera module without any infrared sensor as the main imaging module for capturing was a problem for us in night light.

4.2.2 Software tools

In this section we will present the tools that we have used to develop our system.

4.2.2.1 Python

In our system we have used as a main developing language python with 3.7 version.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed **[82].**

4.2.2.2 TensorFlow

We have developed our CNN model of eye state using TensorFlow version 2.3.

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications **[83].**

4.2.2.3 Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research **[84].**

4.2.2.4 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products **[85].**

4.3 Results

In this section, we will discuss the results we obtained as we developed the system.

4.3.1 Face detection

For the face detection, we have used the pre-trained facial landmark detector inside the “dlib” library**,** which based on the 68 facial landmark model that used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face. This detector is an implementation of the One Millisecond Face Alignment with an Ensemble of Regression Trees **[86].**

4.3.2 Eye state

As we seen in the previous chapter, we have used a CNN model for identifying the eye state, also we have seen the model architecture. Using python language and TensorFlow, we have implemented this architecture into a real CNN model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 98, 98, 16)        448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 49, 49, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 47, 47, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 21, 21, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 6400)              0
_________________________________________________________________
dense (Dense)                (None, 128)               819328
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 843,041
Trainable params: 843,041
Non-trainable params: 0
_________________________________________________________________
```

Figure 22. Eye state model configuration.

The previous figure shows our eye state model architecture configuration which is a summary implementation of the model architecture presented in the previous chapter.

In part of data, we have found many datasets for the driver's eye. But we have used the MRL Eye Dataset to train it with our model where this dataset is a large-scale dataset of human eye images. This dataset contains infrared images in low and high resolution, all captured in various lightning conditions and by different devices. The dataset is suitable for testing several features or trainable classifiers. The eye images are obtained by using the eye detector based on the histogram of oriented gradients combined with the SVM classifier **[87].**

The MRL dataset contains 12000 eye images and we divided them into 2 sets (training and validation sets) at a rate of 80% (10000 images) for training phase and 20% (2000 images) for validation phase.

The model was trained using the previous dataset for 20 epochs, and the results are shows in the next figure.

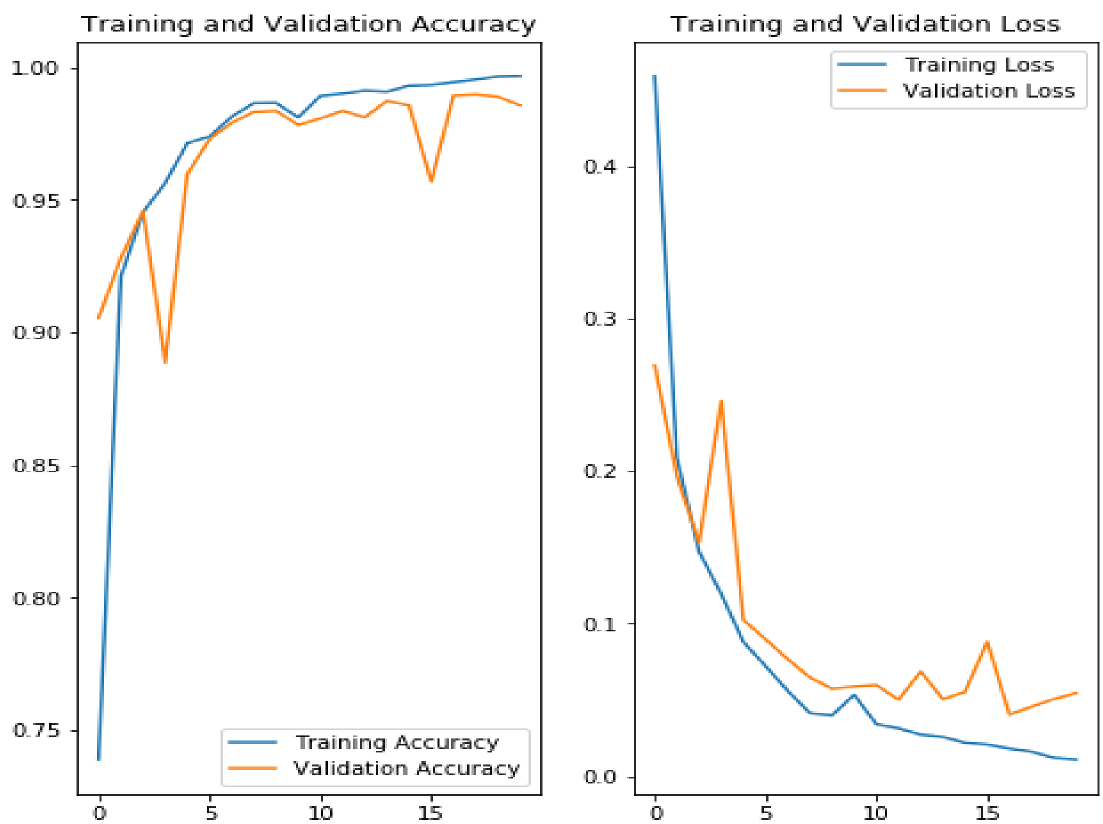


Figure 23. Accuracy and loss results for both training and validation sets.

After analyzing the obtained results. We found that these results present an overfitting problem which happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the model's ability to generalize **[88].** The overfitting problem can be treated by some solutions like using the data augmentation by augment the dataset so that it has a sufficient number of training examples. Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks. **[89].**

But this technique is generally used in a restricted and poor dataset case, and our system, we used around 12000 images from the MRL dataset which is very sufficient number of data(images).

Another technique that use to fix the overfitting problem is the "Dropout" technique. Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel. During training, some number of layer outputs are randomly ignored or "dropped out." This has the effect of making the layer look-like and be treated-like a layer with a different number of nodes and connectivity to the prior layer. In effect, each update to a layer during training is performed with a different "view" of the configured layer. Dropout has the effect of making the training process noisy, forcing nodes within a layer to probabilistically take on more or less responsibility for the inputs. This conceptualization suggests that perhaps dropout breaks-up situations where network layers co-adapt to correct mistakes from prior layers, in turn making the model more robust **[90].**

In our model, we have used the dropout technique with 0.2(set 20% of the neurons to zero during each training epoch) to the last max pooling layer. The new results are shown in next figure **(Figure 24).**

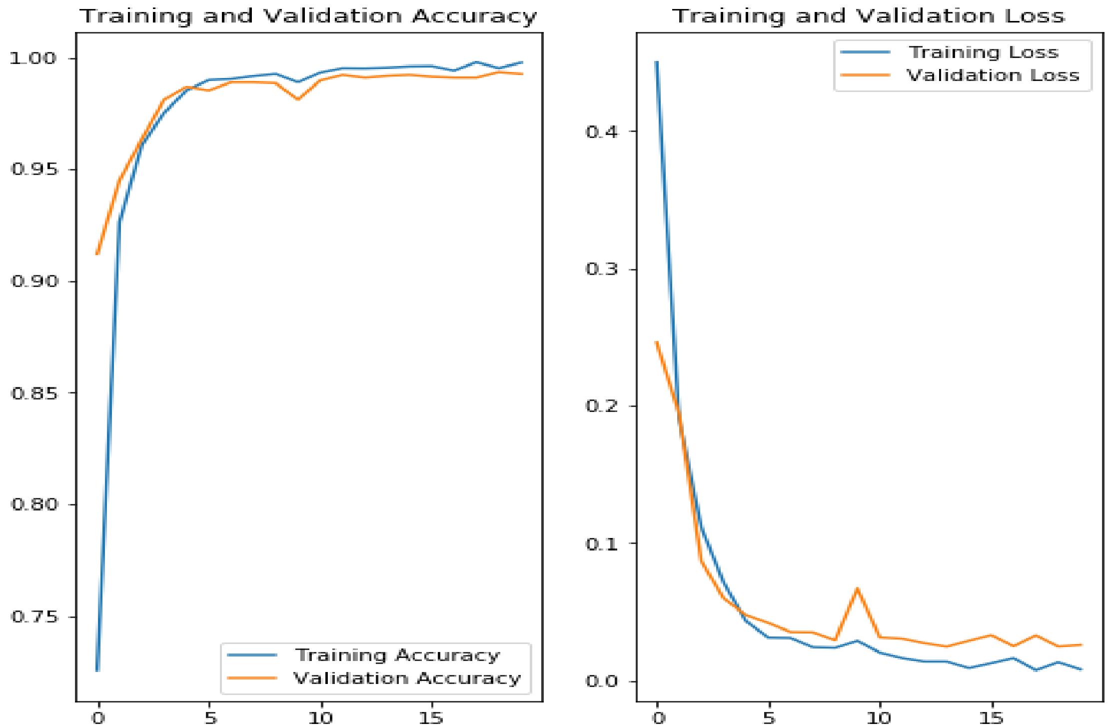


Figure 24. Accuracy and loss results for both training and validation sets after improvement.

After developing the CNN model that allows the determination of the state of the eye (open / closed), we need to find the average duration of an eyelash that allows us to distinguish between normal blinking or the condition in which the eye is already closed. For that, we found that Human adults blink approximately 12 times per minute and one blink lasts about 1/3 second **[91].**

The eye condition detection process is also helped by analyzing the EAR to help the system give a better result, the idea of the EAR is the same as the mouth condition process but with a difference with the eye area (points from 37 to 48) where we can detect About closing the eye when the EAR value drops to 0.

4.3.3 Mouth state

In Mouth state part, the story is a little bit simple because the accuracy and the results quality is related directly to the face (and mouth) detection results. And this due to that we are in this phase just calculating the mouth aspect ratio from the mouth points that have been extracted from the output of the face detection phase. The only work here is to find the yawning threshold which let us distinct between the normally open mouth state and the widely open mouth state which reflect the yawning state where the distance between the two lips reach to its max value and find the yawning time threshold which reflect to How long does it take for the mouth to be widely open(on max value). So, we have analysis the YawDD dataset which offers a variety videos with different conditions. YawDD contains two video datasets of drivers with various facial characteristics. The videos are taken in real and varying illumination conditions **[92].**

We have chosen the second video dataset because the camera is installed on the driver's dash which is similar to what we have installed in our system. In this dataset Each participant has a single video containing scenes with driving, driving while talking, and driving while yawning. This dataset provides 29 videos consisting of both male and female drivers, with and without glasses/sunglasses, from different ethnicities **[92].**

After We have calculated the mouth aspect ratio of all the drivers during time, we have analyzed all the obtained results, we have concluded that:

- In the yawning state, the mouth reaches its maximum opening (MAR>0.75) and stays at this threshold for a short time (3-4 seconds) before it begins to go close.
- Contain multiple state of the mouth in normal case (talking, singing, smiling, or just closed). Where the MAR<0.75.

4.4 System Interfaces

4.4.1 Home page

this page shows the global status of the driver (eye/mouth state and the global state) beside to the driver image.

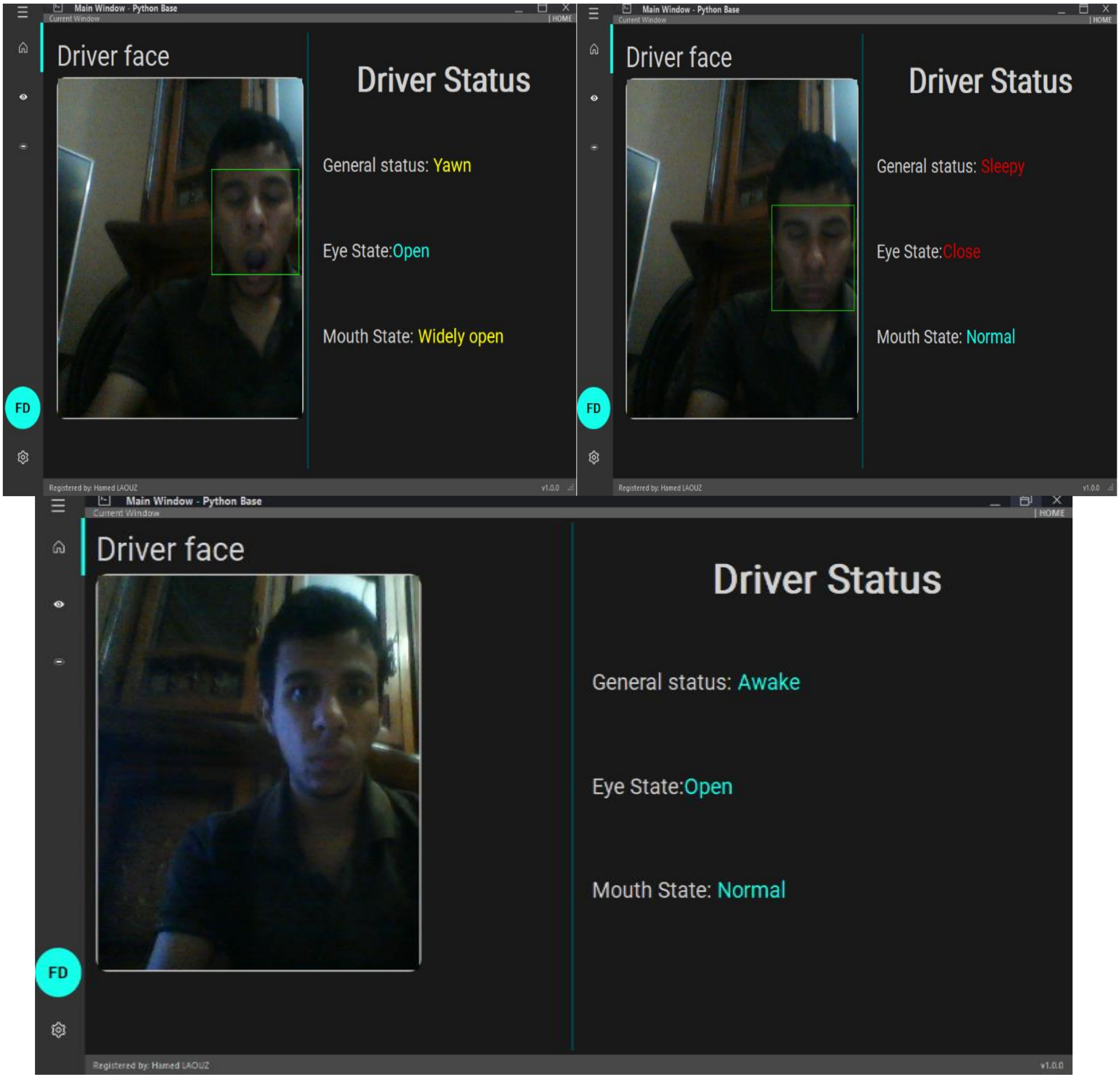


Figure 25. Home page interface of all cases.

4.4.2 Eye state page

In this page we can follow the eye state side by side with eye aspect ratio curve.

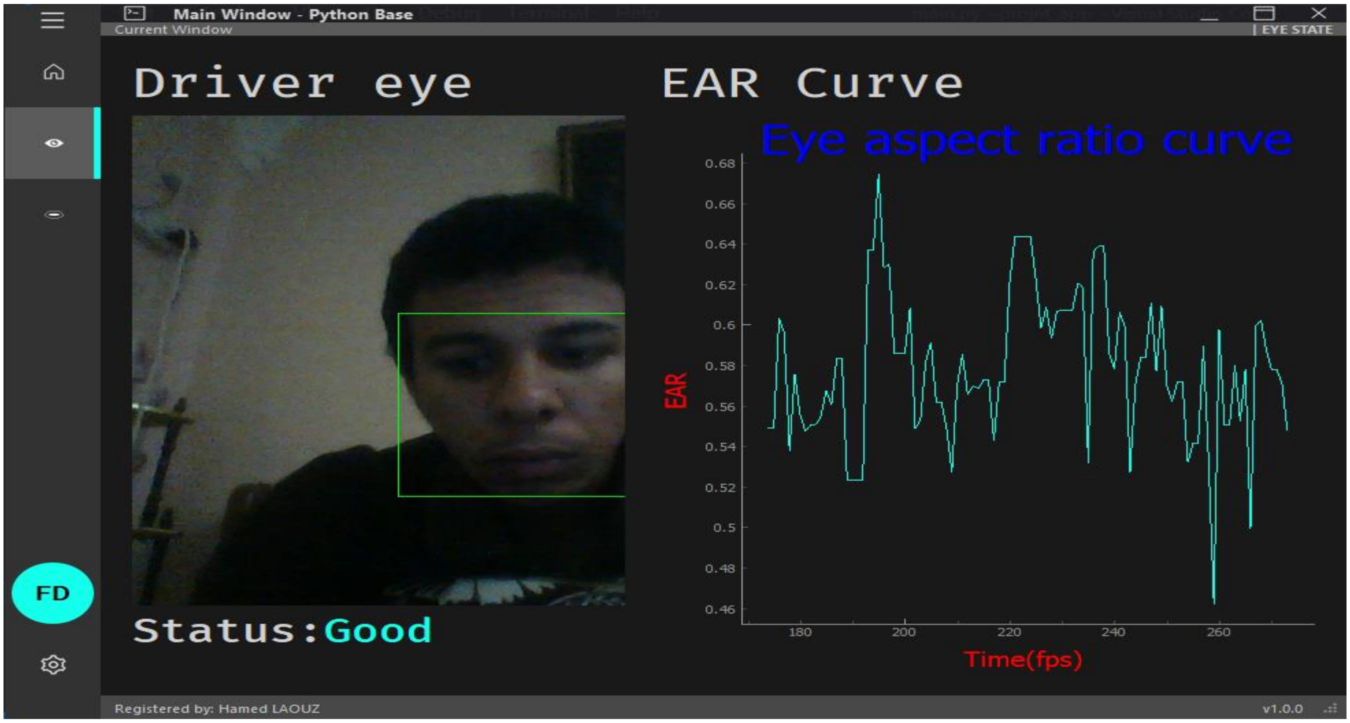


Figure 26. Eye state page interface.

4.4.3 Mouth state page

In this page we can follow the mouth state side by side with mouth aspect ratio curve

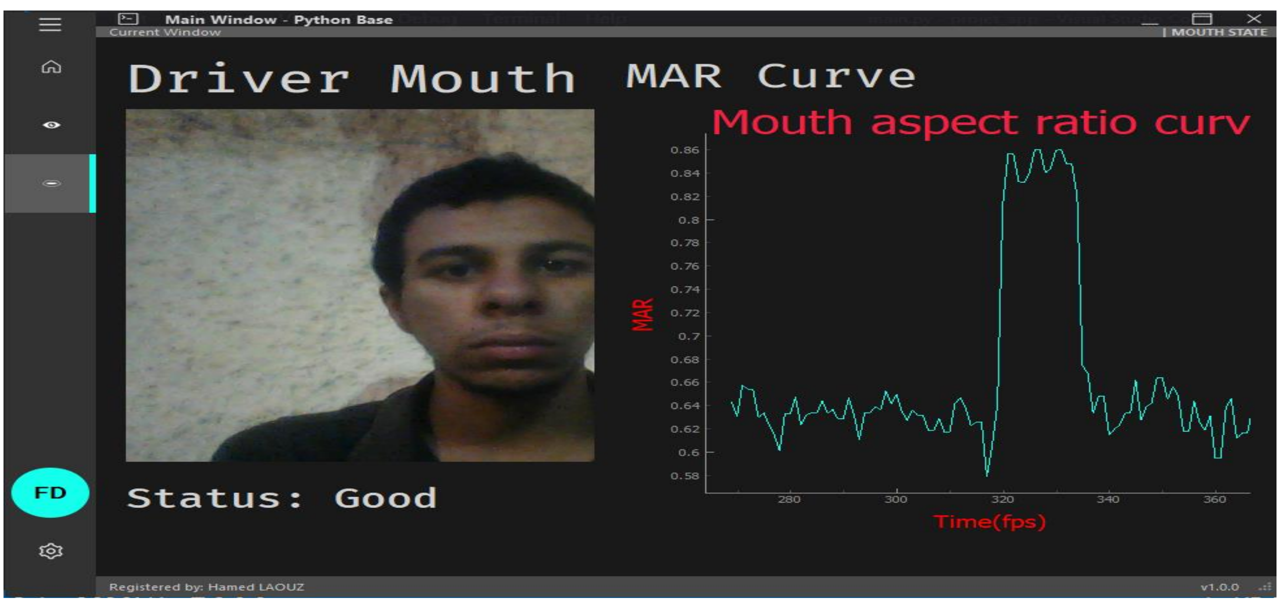


Figure 27. Mouth state page interface.

4.4.4 Settings page

In this page we can manipulate the settings of our system by showning or hiding the eye and mouth curves.Also, we can activate the mouth state to detect any yawning state or not.

Figure 28. Settings page interface.

4.5 Conclusion

In summary, we have developed an embedded system that can detect sleepiness in real time using methods based on the driver's behavioral metrics by analyzing both eye and mouth conditions to classify the driver's condition into one of three states (Awake / Yawn / Sleepy). Although our system suffers from a lack of resources and a problem with night vision, we have developed a system that is able to determine the eye's condition with 93% accuracy in addition to the ability to detect any yawning condition, which may help to better identify the driver's condition.

General Conclusion

General Conclusion

The drowsiness is a serious problem and one of the most important and biggest causes of road accidents, which may cost a lot of material losses and may lead to great human losses as well.

In literature, there are four measurements that can detect the drowsiness: vehicle-based measures, subjective measures, driver behavioral measures, physiological signals measures. Methods for measuring the driver behavior give high-resolution results in the case of good lighting but these results become less good in the light of night because it suffers from the detection of facial expressions. On the other hand, the physiological signal measurements give a stable result in most cases. The only drawback of this measure is that it is an intrusive method for the driver. The hybrid between measures is hard to implement but, it could give a better result and may fix the limitation of each measure.

We have developed a system that can detect the driver's drowsiness by analyzing 2 metrics of driver's behaviors which are: analyze the eye state using a CNN model to verify if the eye is opened or not, the second metric used to augment the system accuracy by analyzing the mouth state to see if the driver is yawning or not which help to predict if the driver is drowsy. Because the system analyzes the data captured from the camera, our system suffering from the problem of the night vision where the light is poor the system can't detect the face of the driver properly.

In term of perspective, we suggest:

- Augment the system accuracy by adding the physiological signals (ECG signal for because it gives the better combination with the eye analysis in term of results and best implementation in real world condition).
- Create a comprehensive dataset from our environment to gives a better result.

List of publications

Laouz, Hamed and Ayad, Soheyb and Terrissa Labib Sadek. (2020). Literature Review on Driver's Drowsiness and Fatigue Detection. The fourth International Conference on Intelligent Systems and Computer Vision on Fez, Morocco. DOI:978-1-7281-8041-0.

Bibliographic part

References

1. drowsy. (n.d.) Farlex Partner Medical Dictionary. (2012). Retrieved August 11 2020 from https://medical-dictionary.thefreedictionary.com/drowsy.

2. American Automobile Association Foundation for Traffic Safety, 2010. Asleep at the wheel: the prevalence and impact of drowsy driving, http://www.aaafoundation.org/pdf/2010DrowsyDrivingReport.pdf, accessed 1/5/11.

3. Leger, D., 1994. The cost of sleep-related accidents: a report for the National Commission on Sleep Disorders Research, Sleep 17(1):84-93.

4. Gavin, E. (2018b, November 18). Machine Learning | An Introduction. Retrieved May 8, 2020, from https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0.

5. Parsers.Inc. (2019, May 20). Deep learning & Machine learning: what's the difference? Retrieved January 5, 2020, from https://parsers.me/deep-learning-machine-learning-whats-the-difference/.

6. Jain, D. (2019, October 29). Addressing the Challenges of On-Device Machine Learning. Retrieved May 1, 2020, from https://medium.com/adobetech/addressing-the-challenges-of-on-device-machine-learning-1f71ebcedd69.

7. Mitchell, T.M .1997. Machine Learning. McGraw-Hill.

8. A. L. Samuel (1959). Some Studies in Machine Learning Using the Game of Checkers IBM Journal of Research and Development, 3(3), 210-229.

9. Dangeti, P.2017. Statistics for Machine Learning. Packt Publishing.

10. Cord, M., & Cunningham, P. (2008). Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval. Springer Berlin Heidelberg.p21.

11. Dua, R., Ghotra, M., & Pentreath, N. (2017). Machine Learning with Spark. Packt Publishing.p107.

12. Frey, B., Brendan, J., & Frey, P. (1998). Graphical Models for Machine Learning and Digital Communication. Bradford book, p. 4.

13. Géron, A .2017. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

14. Sutton, R., & Barto, A. (2018). Reinforcement Learning: An Introduction. MIT Press.p1-2.

15. Gopalakrishnan, R. and Venkateswarlu, A. 2018. Machine Learning for Mobile: Practical guide to building intelligent mobile applications powered by machine learning. Packt Publishing, Birmingham. pp 10_24.

16. B. T. Davis, J. M. Caicedo, V. A. Hirth, B. M. Easterling .2019. Acceleration Signal Categorization Using Support Vector Machines.

17. Sivanandam, S.N. and Deepa, S.N.2007.Introduction to Genetic Algorithms. Springer Berlin Heidelberg.

18. Priddy, K.L. and Keller, P.E..2005. Artificial Neural Networks: An introduction. SPIE Press, Bellingham, WA.

19. Zhou, J. and Chen, F.2018. Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent. Springer International Publishing.

20. Caglar Gulcehre.Deep learning site. http://deeplearning.net.3/20/2020.

21. IJSMI, E. 2019. Deep Learning Models and its application: An overview with the help of R software: Second in series (Machine Learning). Amazon Digital Services LLC - Kdp Print Us.

22. Deng, L. and Yu, D..2014. Deep Learning: Methods and Applications. Now Publishers, Boston.

23. Perez, C. 2017.The deep learning A.I. playbook. Lulu.com Selbstverlag, Buchdruck und Online Publishing,Cham.

24. Dhar, S., Guo, J., Liu, S., Kurup, U., & Shah, M. (2019). On-Device Machine Learning: An Algorithms and Learning Theory.PerspectivearXiv e-prints, arXiv:1911.00623.

25. Manjunath M. 5 Challenges for Developing Mobile Apps with AI & Machine Learning Capabilities. https://heartbeat.fritz.ai/5-challenges-for-developing-mobile-apps-with-ai-machine-learning-capabilities-483668704a60.3.28.2020.

26. Gulli, A., & Pal, S. (2017). Deep Learning with Keras. Packt Publishing, p.61.

27. Campbell, C., & Ying, Y. (2011). Learning with Support Vector Machines. Morgan & Claypool.p1.

28. Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. Water, 11(7), 1387.

29. Olah, C. Understanding LSTM Networks. Available online: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 26 June 2018).

30. Global Status Report on Road Safety 2009; World Health Organization (WHO):Geneva,Switzerland, 2009.

31. Rau, P. Drowsy Driver Detection and Warning System for Commercial Vehicle Drivers: Field Operational Test Design, Analysis, and Progress; National Highway Traffic Safety Administration: Washington, DC, USA, 2005.

32. Drivers Beware Getting Enough Sleep Can Save Your Life This Memorial Day; National Sleep Foundation (NSF): Arlington, VA, USA, 2010.

33. Husar, P. Eyetracker Warns against Momentary Driver Drowsiness. Available online: http://www.fraunhofer.de/en/press/researchnews/2010/10/eye-tracker-driver-drowsiness.html (accessed on 3 May 2020).

34. Azim, T.; Jaffar, M.A.; Mirza, A.M. Automatic Fatigue Detection of Drivers through Pupil Detection and Yawning Analysis. In Proceedings of the 4th Innovative Computing, Information and Control (ICICIC), Kaohsiung, Taiwan, 7–9 December 2009.

35. Ying, Y.; Jing, S.;Wei, Z. The monitoring method of driver's fatigue based on neural network. In Proceedings of the International Conference on Mechatronics and Automation, Harbin, China, 5–8 August 2007.

36. ahayadhas, A.; Sundaraj, K.; Murugappan, M. Detecting driver drowsiness based on sensors: A review.Sensors 2012, 12, 16937–16953.

37. Mardi, Z.; Ashtiani, S.N.M.; Mikaili, M. EEG-based drowsiness detection for safe driving using chaotic features and statistical tests. J. Med. Signals Sens. 2011, 1, 130

38. Szeszko, E. (2017, September 30). Technology against drowsy driving. Retrieved May 4, 2020, from https://medium.com/vorm/technology-against-drowsy-driving-72ede9265b84.

39. Stop Sleep. (n.d.). StopSleep: Anti-sleep alarm. Retrieved May 4, 2020, from https://www.stopsleep.co.uk/.

40. Fletcher Jones Motorcars Newport. (2019, March 11). What is Mercedes-Benz ATTENTION ASSIST? Retrieved May 4, 2020, from https://www.fjmercedes.com/mercedes-benz-attention-assist.

41. Volvo Car Corporation. (2018, July 23). Driver alert Control.Retrieved May 4, 2020, from https://www.volvocars.com/enth/support/manuals/s60/2018/driver-support/driver-alertystem/driver-alert-control-dac.

42. Tunçer, O.; Güvenç, L.; Co¸skun, F. Vision based lane keeping assistance control triggered by a driver inattention monitor. In Proceedings of the International Conference on Systems Man and Cybernetics (SMC), Istanbul, Turkey, 10–13 October 2010.

43. Thiffault, P.; Bergeron, J. Monotony of road environment and driver fatigue: A simulator study. Accid. Anal. Prev. 2003, 35, 381–391.

44. Vural, E. Video-based detection of driver fatigue. Ph.D. Thesis, Sabanci University, Istanbul, Turkey, 2009.

45. Khushaba, R.N.; Kodagoda, S.; Lal, S.; Dissanayake, G. Intelligent driver drowsiness detection system using Uncorrelated Fuzzy Locality Preserving Analysis. In Proceedings of the Intelligent Robots and Systems (IROS) Conference, San Francisco, CA, USA, 25–30 September 2011.

46. Stanford Sleepiness Scale. Available online: http://www.stanford.edu/~dement/sss.html (accessed on 4 May 2020).

47. Shahid, A.;Wilkinson, K.; Marcu, S.; Colin, M.; Shapiro, M. Karolinska Sleepiness Scale (KSS) in STOP, THAT and One Hundred Other Sleep Scales, 1st ed.; Springer: New York, NY, USA, 2012; pp. 209–210.

48. Awais, M., Badruddin, N., & Drieberg, M. (2017). A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability. Sensors (Basel, Switzerland), 17(9), 1991.

49. Yu, X. Real-Time Nonintrusive Detection of Driver Drowsiness; CTS 09-15 Technical Report; University of Minnesota: Minneapolis, MN, USA, May 2009.

50. Lin, C.T.; Wu, R.C.; Liang, S.F.; Chao, W.H.; Chen, Y.J.; Jung, T.P. EEG-based drowsiness estimation for safety driving using independent component analysis. IEEE Trans. Biomed. Circuit Syst. 2005, 52, 2726–2738.

51. T. Danisman, I. M. Bilasco, C. Djeraba and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," 2010 International Conference on Machine and Web Intelligence, Algiers, 2010, pp. 230-233.

52. M. Y. Hossain and F. P. George, "IOT Based Real-Time Drowsy Driving Detection System for the Prevention of Road Accidents," 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, 2018, pp. 190-195.

53. Ghoddoosian, R., Galib, M., & Athitsos, V. (2019). A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection arXiv e-prints, arXiv:1904.07312.

54. Zhang, Z.; Zhang, J. A new real-time eye tracking based on nonlinear unscented Kalman filter for monitoring driver fatigue. J. Contr. Theor. Appl. 2010, 8, 181–188.

55. Shen, W.; Sun, H.; Cheng, E.; Zhu, Q.; Li, Q. Effective driver fatigue monitoring through pupil detection and yawing analysis in low light level environments. Int. J. Digit. Technol. Appl. 2012,6, 372–383.

56. Alioua, N., Amine, A., & Rziza, M. (2014). Driver's fatigue detection based on yawning extraction.International Journal of Vehicular Technology.10.1155/2014/678786.

57. Anitha, C., Venkatesha, M., & Adiga, B. (2016). A Two-Fold Expert System for Yawning DetectionProcedia Computer Science, 92, 63-71.

58. Liu, W., Qian, J., Yao, Z., Jiao, X., & Pan, J. (2019). Convolutional Two-Stream Network Using Multi-Facial Feature Fusion for Driver Fatigue Detection Future Internet, 11, 115.

59. Owais, S. (2017, Aug. 10). Eye Blink Detection Algorithms Retrieved May 29,2020 from: https://hackaday.io/project/27552-blinktotext/log/68360-eye-blink-detection-algorithms.

60. Picot, A.; Charbonnier, S.; Caplier, A. On-line detection of drowsiness using brain and visual information, Systems, Man and Cybernetics, Part A: Systems and Humans. IEEE Trans. Syst. Hum.2012, 42, 764–775.

61. Elsenbruch, S.; Harnish, M.J.; Orr, W.C. Heart rate variability during waking and sleep in healthy males and females. Sleep 1999, 22,1067–1071.

62. Murugan, S., Selvaraj, J. & Sahayadhas, A. Detection and analysis: driver state with electrocardiogram (ECG). Phys Eng Sci Med (2020).

63. Barua, S. (2019). Multivariate Data Analytics to Identify Driver's Sleepiness, Cognitive load, and Stress. (Doctoral dissertation, Mälardalen University).

64. Patel, M.; Lal, S.K.L.; Kavanagh, D.; Rossiter, P. Applying neural network analysis on heart rate variability data to assess driver fatigue. Exp. Syst. Appl. 2011, 38, 7235–7242.

65. Khushaba, R.N.; Kodagoda, S.; Lal, S.; Dissanayake, G. Driver drowsiness classification using fuzzy wavelet-packet-based featureextraction algorithm. IEEE Trans. Biomed. Eng. 2011, 58, 121–131.

66. Akin, M.; Kurt, M.; Sezgin, N.; Bayram, M. Estimating vigilance level by using EEG and EMG signals. Neural Comput. Appl. 2008,17, 227–236.

67. Hu, S.; Zheng, G. Driver drowsiness detection with eyelid related parameters by support vector machine. Exp. Syst. Appl. 2009, 36,7651–7658.

68. Kurt, M.B.; Sezgin, N.; Akin, M.; Kirbas, G.; Bayram, M. The ANN based computing of drowsy level. Exp. Syst. Appl. 2009, 36, 2534–2542.

69. Lin, C.T.; Chang, C.J.; Lin, B.S.; Hung, S.H.; Chao, C.F.; Wang, I.J. A real-time wireless brain–computer interface system for drowsiness detection. IEEE Trans. Biomed. Circuit Syst. 2010, 4, 214–222

70. Guosheng, Y.; Yingzi, L.; Prabir, B. A driver fatigue recognition model based on information fusion and dynamic Bayesian network. Inform. Sci. 2010, 180, 1942–1954.

71. Lee, B.G.; Jung, S.J.; Chung, W.Y. Real-time physiological and vision monitoring of vehicle driver for non-intrusive drowsiness detection. Commun. IET 2011, 5, 2461–2469.

72. Cheng, B.; Zhang, W.; Lin, Y.; Feng, R.; Zhang, X. Driver drowsiness detection based on multisource information. Hum. Factors Ergon. Manuf. Serv. Indust. 2012, 22, 450–467.

73. L. Oliveira, J. S. Cardoso, A. Lourenço and C. Ahlström, "Driver drowsiness detection: a comparison between intrusive and nonintrusive signal acquisition methods," 2018 7th European Workshop on Visual Information Processing (EUVIP), Tampere, 2018, pp. 1-6.

74. Hayat,R.H.(Monday, 23 November 2015). SumVoices: Problems Without End in Algeria's internet. http://blog.sumrando.com/2015/11/sumvoices-problems-without-end-in-algerias-internet.html.

75. Keomany, Jean & Marcel, Sébastien. (2006). Active Shape Models Using Local Binary Patterns.

76. Wu, Y., & Ji, Q. (2018). Facial Landmark Detection: A Literature Survey. International Journal of Computer Vision, 127(2), 115–142.

77. Soni, B., Verma, G., Gao, X., & Borgohain, S. (2020). MACHINE LEARNING, IMAGE PROCESSING, NETWORK SECURITY AND DATA SCIENCES. Springer.p175.

78. M. Omidyeganeh, A. Javadtalab, & S. Shirmohammadi (2011). Intelligent driver drowsiness detection through fusion of yawning and eye closure. In 2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings (pp. 1-6).

79. Ashlesha Singh, Chandrakant Chandewar, and Pranav Pattarkine.2018. Driver Drowsiness Alert System with Effective Feature Extraction. INTERNATIONAL JOURNAL FOR RESEARCH IN EMERGING SCIENCE AND TECHNOLOGY.

80. Pranali Awasekar, M. Ravi, Shivani Doke, & Zaheed Shaikh (2018). Driver Fatigue Detection and Alert System using Non-Intrusive Eye and Yawn Detection International Journal of Computer Applications, 180, 1-5.

81. Raspberry Pi Foundation. (n.d.). What is a Raspberry Pi? Retrieved May 20, 2020, from https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/.

82. Python Software Foundation. (n.d.). What is Python? Executive Summary. Retrieved May 20, 2020, from https://www.python.org/doc/essays/blurb/.

83. TensorFlow. (n.d.). Why TensorFlow. Retrieved May 20, 2020, from https://www.tensorflow.org/.

84. Keras. (n.d.). Keras. Retrieved May 20, 2020, from https://keras.io/.

85. OpenCV team. (n.d.). About. Retrieved September 17, 2020, from https://opencv.org/about/

86. V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1867-1874.

87. MRL. (n.d.). MRL Eye Dataset. Retrieved May 20, 2020, from http://mrl.cs.vsb.cz/eyedataset.

88. Brownlee, J. (2016). Master Machine Learning Algorithms: Discover How They Work and Implement Them from Scratch. Jason Brownlee.p23.

89. Ho, D., Liang, E., & Liaw, R. (2019, June 7). 1000x Faster Data Augmentation. Retrieved August 11, 2020, from https://bair.berkeley.edu/blog/2019/06/07/data_aug/

90. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res., 15(1), 1929–1958.

91. Kyung-Ah Kwon, Rebecca J. Shipley, Mohan Edirisinghe, Daniel G. Ezra, Geoff Rose, Serena M. Best, Ruth E. Cameron J R Soc Interface. 2013 Aug 6; 10(85): 20130227. doi: 10.1098/rsif.2013.0227.

92. S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, “YawDD: A Yawning Detection Dataset”, Proc. ACM Multimedia Systems, Singapore, March 19 -21 2014, pp. 24-28.