INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL

Fatigue detection system to aid in remote work

GONÇALO GOMES TEIXEIRA
Outubro de 2022

P.PORTO

Fatigue detection system to aid in remote work

Gonçalo Gomes Teixeira

Student No.: 1170700

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Artificial intelligence

Supervisor: Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Evaluation Committee:

President:

Doutor Carlos Fernando da Silva Ramos, Professor Coordenador Principal do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Members:

Doutora Ana Maria Neves Almeida Baptista Figueiredo, Professora Coordenadora do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Porto, October 25, 2022

Abstract

During the Covid-19 pandemic there was a noticeable surge in the amount of remote workers. In the aftermath of the pandemic working from home still remains a reality for many workers with noticeable impacts on the mental health of people. With the increased stress caused by current situation and the harder time establishing boundaries there was an increase in the overall stress and fatigue in workers, leading to burnouts.

Fatigue detection systems are used in several areas, mainly in the automotive industry as a mean to decrease the number of accidents. This research started by approaching the Artificial Intelligence (AI) area and its domains, followed by a study of the current techniques used in order to predict fatigue. With the main ones utilising eye state, facial landmarks, electrocardiogram or heart rate. After a research into existing Fatigue detection systems was done in order to identify the strengths of solutions currently in the market, whether in the automotive industry or other applications.

This thesis proposes the creation of a system able to detect fatigue in a user as well as warn him when fatigue levels increase. This system incorporates a webcam analysing the users face and performing eye state detection in order to calculate the percentage of the time the eyes are closed (PERCLOS). Heart rate data was also analysed and a model was developed in order to incorporate this data, the percentage of time the eyes are closed, the program the user has open and time of day in order to predict the level of fatigue. By combining these two different techniques this system can be more effective and more accurate in giving predictions of the level of fatigue. The review of literature showed that the conjunction of these two techniques in predicting fatigue is novelty. The developed system also contains integration with smartwatch technology in order to both harness heart rate data as well as communicate with the user via pop up notifications to inform him when fatigue levels get too high.

The conclusion of this work is that eye state detection using Artificial Intelligence can achieve a high accuracy and be a reliable tool in identifying fatigue in an user. The combination of Heart Rate and PERCLOS allows the system to have a higher accuracy as well as not being completely reliant on one sensor. The creation of a fatigue prediction model was hindered by the lack of existent data in order to train a model, a problem that could be fixed with the adoption of the system in a broader scope.

Keywords: Covid-19,Artificial Inteligence, Computer vision, Fatigue detection system

Resumo

Durante a pandemia de Covid-19, houve um aumento notável na quantidade de trabalhadores remotos. No rescaldo da pandemia, trabalhar a partir de casa continua a ser uma realidade para muitos trabalhadores, com impactos visíveis na saúde mental das pessoas. Com o aumento do stresse causado pela situação atual e a dificuldade de estabelecer limites, houve um aumento do stresse geral e da fadiga dos trabalhadores, levando ao esgotamento.

Os sistemas de detecção de fadiga são utilizados em diversas áreas, principalmente na indústria automobilística como forma de diminuir o número de acidentes. Este estudo começou por abordar a área de Inteligência Artificial (IA) e os seus domínios, seguida de um estudo das técnicas atuais utilizadas para prever a fadiga. Com os principais utilizando o estado dos olhos, pontos de referência faciais, eletrocardiograma ou frequência cardíaca. Depois foi feita uma pesquisa sobre os sistemas de detecção de fadiga existentes de forma a identificar os pontos fortes das soluções actualmente no mercado, quer seja na indústria automóvel ou outras aplicações.

Esta dissertação propõe a criação de um sistema capaz de detectar fadiga num utilizador, bem como alertar quando os níveis de fadiga aumentam. Este sistema incorpora uma webcam que analisa a face do utilizador e realiza a detecção do estado dos olhos para calcular a percentagem de tempo em que os olhos estão fechados (PERCLOS). Os dados de frequência cardíaca também foram analisados e um modelo foi desenvolvido para incorporar estes dados, a percentagem de tempo que os olhos ficam fechados, o programa que o utilizador tem aberto e a hora do dia para prever o nível de fadiga. Ao combinar essas duas técnicas diferentes, este sistema pode ser mais eficaz e mais preciso em fornecer previsões do nível de fadiga. A revisão da literatura mostrou que a conjunção dessas duas técnicas na previsão da fadiga é novidade. O sistema desenvolvido também contém integração com a tecnologia smartwatch para aproveitar os dados da frequência cardíaca e comunicar com o utilizador por meio de notificações pop-up para informá-lo quando os níveis de fadiga se encontrarem altos.

A conclusão deste trabalho é que a detecção do estado ocular usando Inteligência Artificial pode alcançar uma alta precisão e ser uma ferramenta confiável na identificação de fadiga num utilizador. A combinação da frequência cardíaca e PERCLOS permite que o sistema tenha maior precisão, além de não depender completamente de um unico sensor. A criação de um modelo de previsão de fadiga foi dificultada pela falta de dados existentes para treinar um modelo, problema que poderia ser colmatado com a adoção do sistema numa população maior.

Palavras-chave: Covid-19,Artificial Inteligence, Computer vision, Fatigue detection system

Contents

List of Figures

List of Tables

List of Algorithms

List of Source Code

List of Acronyms

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
CNN	Convolution Neural Network.
ML	Machine Learning.
PERCLOS	Percentage of eyelid closure over the pupil over time.

Chapter 1

Introduction

In this chapter the context and motivation of this thesis will be discussed. These are especially important given that this work is written during the fallout of the Covid-19 pandemic and it's effects. Then the objectives are explored as well as the methodology for this research and a brief summary of the contents of this thesis.

1.1 Context and motivation

In 2020 the world was affected by a global pandemic that brought the need to social distance and in some cases work remotely from home. Thus, this pandemic caused a colossal surge in the amount of people working from home. Research points to the fact that as much as 71% of American workers worked remotely during the year of 2020. A majority of countries also promoted remote work in cases where it was possible [1]. This increase in the number of workers performing their job remotely is here to stay as inquires of remote workers suggest that this trend was not just a fad but a permanent change in our society.

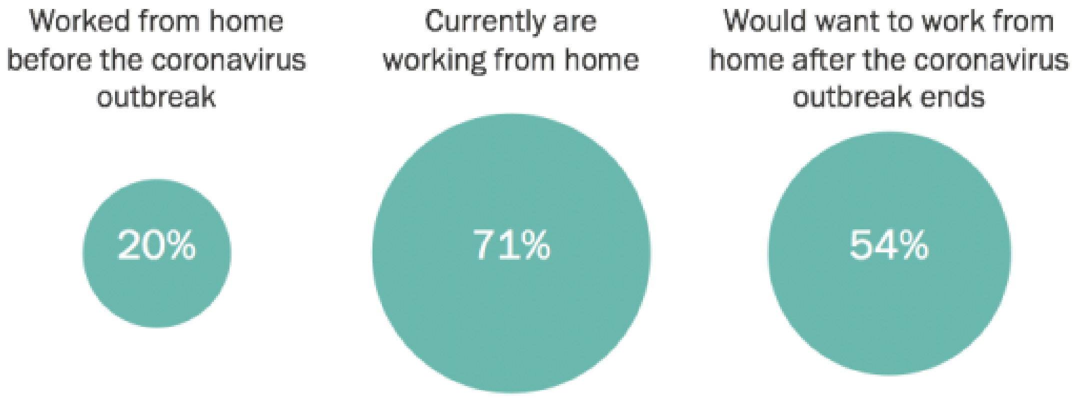

Figure 1.1: Survey of U.S. adults conducted Oct. 2020. "How the Coronavirus Outbreak has - and Hasn't - changed the way Americans work"

It is of note in Figure 1.1 that 54% of the workers that worked remotely during the pandemic, if possible, will choose to continue to work from home after the pandemic is handled and it is once again safe to return to the office. This percentage is not the same across all fields as technological companies adopted remote work much better and with even a wider scope with up to 28% of job postings from tech companies in the United States boasting the ability to work from home as a perk [2]. These facts reflect the permanence of remote work with both it's upsides and downsides.

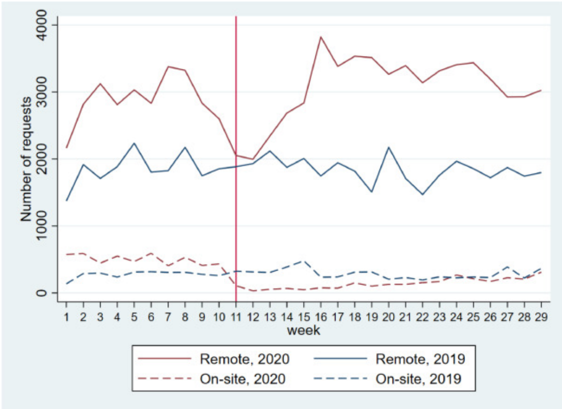

Figure 1.2: Increase in the amount of search in remote work

Source: [3]

This data is also supported by studies into the European work force [3]. In Figure 1.2 it is also of note that the amount of workers utilizing a remote working system. In this figure an increase in the number from the years of 2019 to 2020 shows the change in the current system.

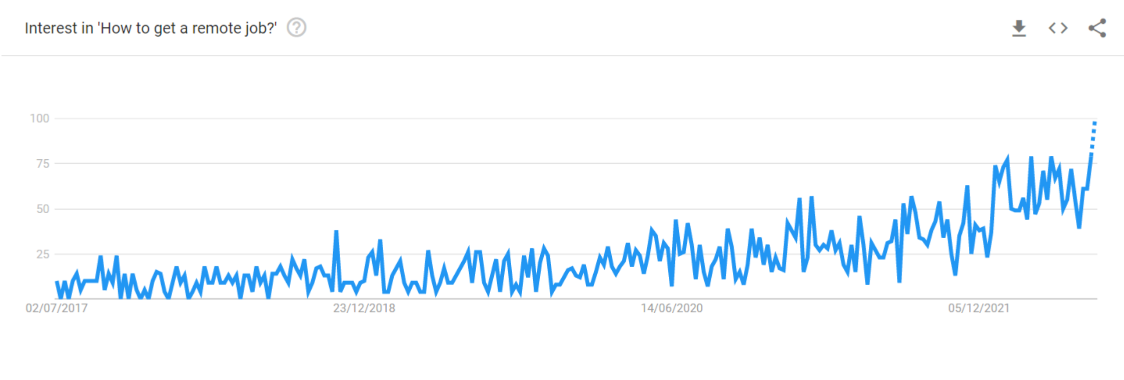

Figure 1.3: Google trend search of interest in getting a remote job

Google trend search is a tool created by Google that analyses every Google search and allow the users to search for terms and know how many people have searched for a certain topic over time. Google search trends where used here to show the evolution of the interest of the general population as there was an increase in the number of searches. In figure 1.3 it is of note the evolution in the number of people interested in learning more on how to get a remote job. This serves to prove that the surge of workers performing their jobs from home is still very much active and increasing, demonstrating the adaptation of the job market to embrace remote work even further.

One of the downsides of remote work is the risk of burnouts. Burnout is defined as a three pronged fork: emotional exhaustion, reduced personal accomplishment and depersonalization [4]. Emotional exhaustion, which is the feeling of being depleted of one's emotional resources, is regarded as the basic individual stress component of the syndrome. Reduced personal accomplishment indicates a tendency to evaluate oneself negatively with regard to one's competence and productivity and a lowered sense of self-efficacy. Depersonalization refers to negative, cynical or excessively detached responses to other people at work[4]. Thus burnout can be defined as a psychological syndrome that is the result of long-term, job-specific, physical and emotional exhaustion from interpersonal stress that results in detachment, cynicism, reduced feelings of efficacy and accomplishment and may have significant impacts on job performance and satisfaction.

There is a direct relationship between working remotely and and increased number of stress and fatigue, with the existence of chronic workplace stress being a well documented fact [5]. A survey made of professionals working remotely [5] sought out to answer multiple questions, amongst them how the perceived stress levels of workers varied before and after the covid-19 and what causes where in the origin of this increase in the stress levels in the working population. The survey was able to answer both of these questions pointing to a noticeable increase in the levels of stress amongst the workers interviewed. There was the interesting caveat that the increase in the stress levels of workers in part-time jobs was noticeably lower than that of those working full time jobs. The factors identified in this survey for the increase in the amount of perceived stress were mainly:

- maintaining appropriate levels of communication with my team/colleagues (21.36%);
- managing technology/communication tools (19.20%);
- managing my time/avoiding distractions (18.42%);

With the advent of new and faster hardware, Artificial Intelligence (AI) has earned a new breath and is experiencing a surge in popularity [6]. The main goal of AI is to empower computers to perform intellectual tasks such as decision making, problem solving, perception and understanding human communication [6]. Thus AI excels in the automatizing of tasks as well as solving problems complex problems.

AI has also recently gained a more prominent role when it comes to building knowledge from complex data [7] via the utilization of techniques such as deep learning. This new technique has been showing superior results in comparison to other machine learning techniques widely used until then. In this technique the computer learns what is naturally done by humans, making it possible to learn from the direct classification of reading and images, text or sound. Deep learning has exceeded expectations in terms of results and often exceeds human performance [8].

This conjugation of factors gave way to the idea to use deep learning to create a fatigue detection system that can aid workers manage fatigue and decrease perceived fatigue by informing them about their levels of fatigue and insight over external factors that have an impact on how the worker feels fatigue. This fatigue detection system can thus be a contributing factor in avoiding excessive fatigue and in helping workers maintain a healthy work life balance. It can even help workers avoid and prevent burnouts. This ability makes it important especially as the amount of remote workers continues to grow.

1.2 Objectives

The main objective of this thesis is to create a Fatigue Detection System that can observe and identify fatigue in the worker. The goal is to create a Fatigue detection system by combining two existent methods by combining analysis of the heart rate of the user and the percentage of time the eye is closed(PERCLOS). By combining two techniques it is possible to obtain a higher overall accuracy in the detection of fatigue. Hence the goal of this system is to incorporate two different techniques and study the overall performance of the system.

In order to create a tool with such abilities AI must be incorporated into the system, giving it the capacity of analysing complex data and learning how to identify the level of fatigue.

The produced tool must be able to:

- Harness all required data to predict level of fatigue;
- Predict level of fatigue in the user;
- Warn the user when fatigue levels get over a certain threshold.

This system needs to be able to calculate and process information related to the percentage of the time the eye is open (PERCLOS) as well as heart rate data. With this information it is possible to predict the level of fatigue of the user at a given time. If a value of fatigue above the desired maximum threshold is detected the system needs to be able to warn the user of the excessive level of fatigue.

In order to fulfill these objectives a research question was formulated. The question is as follows: Will it be possible to create a hybrid fatigue detection system that combines more than one fatigue detection technique and warn an user when fatigue levels increase?

1.3 Investigation Methodology

Prior to developing any solution, careful and thorough research is required to analyze and select the appropriate development tools, methods and research strategies to use to develop the solution. In this paper the research methodology used was PRISMA. PRISMA is an evidence-based minimum set of items for reporting in systematic reviews and meta-analyses. PRISMA primarily focuses on the reporting of reviews evaluating the effects of interventions, but can also be used as a basis for reporting systematic reviews with objectives other than evaluating interventions (e.g. evaluating aetiology, prevalence, diagnosis or prognosis) [9].

In figure 1.4 a flow chart on how the research using PRISMA is done. In the beginning records are identified and then screened(usually via the reading of the abstract) and then exclusion criteria are applied to the records that passed the screening phase.

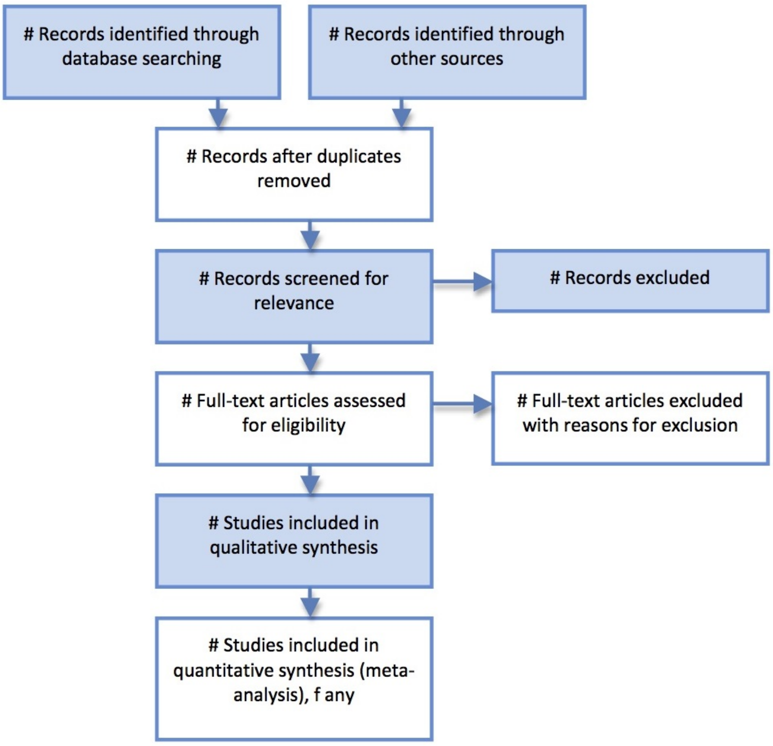

Figure 1.4: PRISMA flowchart

Source: [10]

1.4 Contributions

The creation of this fatigue detection system can be a factor in helping users to prevent burnout and improve the user's lives. By warning the user when the detected fatigue level is above a certain threshold the user can take measures to improve his productivity and routine.

This thesis also presents a technological contribution as this combination of two different techniques is something that does not seem to exist as it could not be found in literature. This work is therefor an innovation, with this system capable of obtaining better results and creating a more robust system that is not as dependant on one sole technique or sensor.

1.5 Structure of the dissertation

To better understand this dissertation its structure needs to be defined. This thesis has 5 chapters.

The present chapter, Chapter 1, approaches the theme and motivation as well as an introduction to the research methodology used.

Chapter 2 approaches the state of the art starting with the development of the PRISMA method, followed by the research into Artificial Intelligence and Fatigue detection systems using several approaches.

Chapter 3 contains a high level explanation of the proposed system. It contains a description of the modules and functionalities required in order for the system to work as well as detail the flows and interactions between them and each modules responsibility.

Chapter 4 is a more detailed look into the the modules presented in Chapter 3. This in depth look analyses all the decisions and steps takes as well as an analyses of the results of the models developed.

Chapter 5 contains the conclusions takes from the development of this thesis.

Chapter 2

State of the Art

In this section the research for this project will be exposed and the associated methodology will be described , as well as the resulting information.

2.1 Research Methodology

The purpose of this research is to extract and analyze documents that are related to the automation issues referred to in the introduction. This systematic review was performed in accordance with the Preferred Reporting Items for Systematic Reviews and Meta-Analyses. As defined in PRISMA, we indicate the protocol used by the team:

- Define the Research Questions;
- Define the Information Sources. They are electronic repositories to be used in the research;
- Identify the Research Terms. These are the basis for the search query that will be used, without changes, in the different Information Sources;
- Inclusion and Exclusion criteria, along with quality assessment criteria are described. Finally, a description of the data extraction process is presented in order to explain how the selected articles were continuously filtered until the final copies.

2.1.1 Research Questions

Initially we started with the identification of the main purpose of this study in order to identify the main research questions and subjects of research in this paper. This consisted of analysing the following topics:

- The state of the art of Artificial Intelligence.
- The state of fatigue detection.
- The state of fatigue detection systems.

In table 2.1 this topics were divided in research questions. RQ1 arises from the need to incorporate intelligent behavior into the system. This is necessary in order to provide the end user with as much accurate information as possible and to improve productivity in the greater amount possible. RQ2 portraits to the current techniques utilizes in order to detect fatigue and the state of the art of fatigue detection. RQ3 portraits to the state of the art of fatigue detection systems, as well as to which commercially available solutions exist.

Table 2.1: Research Questions

ID	Research Question
RQ1	What approaches using Artificial Intelligence (AI) exist?
RQ2	Which approaches exist to detect fatigue in users?
RQ3	Which fatigue detection systems exist to detect fatigue?

2.1.2 Information sources

The electronic databases chosen for this study are identified in table 2.2. The last survey was carried out on these on the 20th of June, 2022.

Table 2.2: Electronic Databases

ID	Database
DB1	B-on
DB2	Web of Science (WOS)
DB3	Google Scholar

2.1.3 Search

The search terms used in this review in response to the three Research Questions are identified in table 2.3

Table 2.3: Keywords used to select search terms

Research questions	Keywords
RQ1	(TI AI or TI artificial intelligence or TI Machine Learning or TI Deep Learning)
RQ2	(TI fatigue detection OR TI fatigue detection system AND AB camera AND SU fatigue detection) NOT TX (eeg or electroencephalogram or electroencephalography or brain waves or erp or event related potentials or muscle fatigue or ecg or EMG or brainwaves or waveform or muscular visual fatigue or fatigue damage)
RQ3	(TI fatigue detection system)

The search terms were applied to all parameters provided by default in each of the two electronic databases. We cannot say that these terms were, for example, applied only to the Title, Abstract and Keywords parameters. We were able to affirm that they were applied to the indicated parameters, as it is common to the two electronic databases, however each one presents other parameters as a standard, in addition to these, and different between the electronic databases.

It was not possible to get results with the search performed with the operator "AND". With this research we intended to obtain the intersection of the three domains. Thus, we chose to use the "OR" operator, joining the results of those domains. The final query can be found in the table 2.4.

Table 2.4: Final query used

Final query
(TI AI or TI artificial intelligence or TI Machine Learning or TI Deep Learning) OR (TI fatigue detection OR TI fatigue detection system AND AB camera AND SU fatigue detection) NOT TX (eeg or electroencephalogram or electroencephalography or brain waves or erp or event related potentials or muscle fatigue or ecg or EMG or brainwaves or waveform or muscular visual fatigue or fatigue damage) OR (TI fatigue detection system)

2.1.4 Quality Assessment: Inclusion and Exclusion Criteria

In order to determine whether or not a document should be included in this review, inclusion and exclusion criteria were defined. These are identified in table 2.5 and table 2.6.

Table 2.5: Inclusion criteria

ID	**Criteria**
IC1	Document type: articles, conference paper, proceedings papers, review articles
IC2	Document addresses topics that fall within the defined domains
IC3	Artificial Intelligence Techniques

Table 2.6: Exclusion criteria

ID	**Criteria**
EC1	Before 2017
EC2	The document is not written in English
EC3	Public data set without reference location
EC4	Document not accessible because it is paid or because it is not possible to access it via the educational institution
EC5	Abstract does not refer to results

Some of these criteria were applied immediately when searching on those engines as they have common filters. Among these criteria are: IC1, EC1 and EC2.

2.1.5 Data Extraction

In Figure 2.1 the flowchart with the steps and quality criteria applied can be seen.

The screening phase was made up of two main phases, the screening of the title and the documents that passed this phase where then subject to a screening of the abstract hence any result from the databases that did not contain both a subject and abstract were rejected before the screening step.

The results of the research are shown in the following sections with the selected articles being used as support for the research gathered.

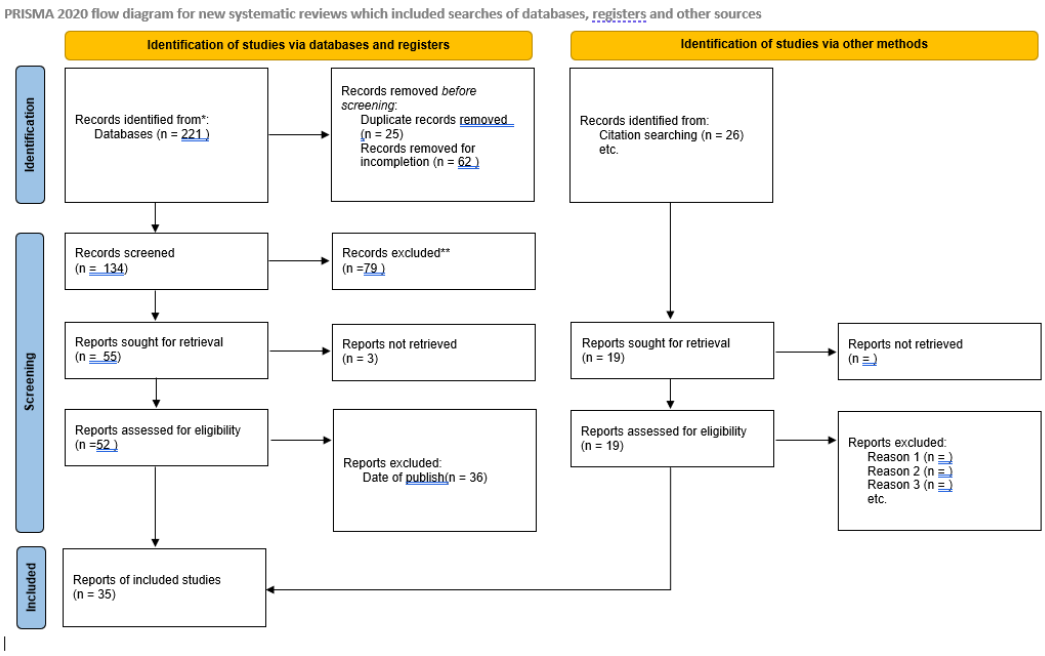

Figure 2.1: Decision Process in PRISMA

Source: [11].

2.2 Result of RQ1- Artificial Intelligence

Artificial Intelligence (AI) deals with algorithms capable of learning, perceiving, and solving problems that would otherwise require human intelligence. Often Artificial Intelligence does so by mimicking human behaviour or learning from sample data curated by a data scientist in order to train a model to solve a problem. AI is a field of study with several paths that are going to be explored in the following sections. Some of these paths are [12]:

- Fuzzy logic;
- Knowledge-based systems (also refered to as expert systems);
- Genetic algorithms;
- Machine Learning;

2.2.1 Fuzzy logic

Real-world issue solving necessitates juggling a variety of ambiguous variables. Different sorts of uncertainty, including randomness, fuzziness, indistinguishability, and incompleteness, can be discovered depending on the variable nature of the uncertainty.

Different approaches, including fuzzy logic, have been offered in the AI area to deal with this ambiguity. In essence, fuzzy logic enables the processing of inaccurate data from the actual world. Fuzzy logic makes it possible to express information from the actual world in a way that is both more accurate and straightforward. In comparison to other model types, fuzzy logic-based models often need fewer rules and variables. The foundation of fuzzy set theory is the notion that things are frequently permitted to be a question of degree in the real world [13].

In more technical words, the fundamental feature of fuzzy logic is that, in contrast to traditional logical systems, it seeks to reflect the imprecise forms of reasoning that are crucial to the amazing capacity of humans to make reasoned judgments in the face of uncertainty and imprecision. This capability is dependent on our capacity to deduce a rough response to a query from a body of knowledge that is imperfect, partial, or not entirely trustworthy. For instance: Usually it takes about half an hour to commute home from work. Exactly how long will the trip take today?[14];

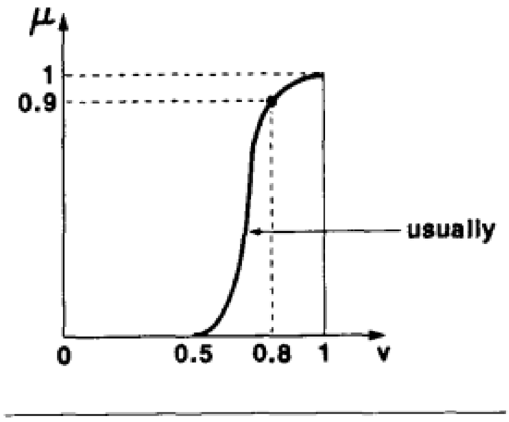

Figure 1. Representation of "usually" as a fuzzy proportion.

Figure 2.2: Representation of "usually" as a fuzzy proportion.

Source: [14].

In this statement, "generally" functions as a hazy component of the structure depicted in Figure 2.2. The majority of what is typically thought of as common sense information may be seen as a collection of dispositions, which is why dispositional logic is important. Therefore, the primary focus of dispositional logic is on the creation of inference rules based on common sense[14].

2.2.2 Knowledge-based systems

Domain knowledge is represented by a collection of rules in Expert Systems, while the current situation is provided by the set of facts kept in the database. It is the responsibility of an inference engine to match the rule with the fact. The firing rule may alter the collection of facts and introduce new ones [15].

An expert system is a computer program that[16]:

- reasons with symbolic as well as mathematical knowledge;
- uses heuristic (possible) as well as algorithmic (certain) methods;
- performs as well as specialists in its problem area;
- makes what it knows and the reasons for its answers understandable;

- retains flexibility.

An expert system is, by definition, a tool that aids human decision making. It's used for knowledge-intensive jobs that need skill. As a result, wherever we find people performing tasks like diagnosing a system, constructing a structure, or educating a pupil, we have also found a place for technology. There are several established expert systems covering a wide range of areas. Table 2.7 depicts the primary application areas that evolved naturally—the breadth of the applications is astounding[17].

Problem Type	Description
Control	Governing system behavior to meet specifications
Design	Configuring objects under constraint
Diagnosis	Inferring system malfunctions from observables
Instruction	Diagnosing, debugging, and repairing student behavior
Interpretation	Inferring a situation description from data
Monitoring	Comparing observations to expectations
Planning	Designing actions
Prediction	Inferring the likely consequences of given situations
Prescription	Recommending solutions to system malfunctions
Selection	identifying the best choice from a list of possibilities
Simulation	Modeling the interaction between system components

Table 2.7: Fields of usage of expert systems

Source: [17].

2.2.3 Genetic algorithms

The traditional Genetic Algorithm (GA) is based on a set of candidate solutions that provide a solution to the optimization issue at hand. A solution is a prospective candidate for the optimization problem's optimum. Its representation is critical since it decides which genetic operators are used. Representations are often lists of values and are based on sets of symbols. They are termed vectors if they are continuous, and bit strings if they are made up of bits. In the case of combinatorial issues, the solutions frequently consist of symbols from a list. In the context of the traveling salesman dilemma, one example is traveling salesman problem [18].

Genetic operators generate new solutions in the chosen representation and enable navigation in solution space. The coding of the answer as a representation, which is subject to evolution, is known as genotype or chromosome.

Algorithm 2.1 Genetic algorithm
Based on model from: [18]

```
procedure Genetic Algorithm
    Initializepopulation
    while Terminus condition is not met do
        while All individuals are not processed do
            Crossover
            Mutation
            Phenotypemapping
            Fitnesscomputation
        end while
        Selectionofparentalpopulation
    end while
end procedure
```

GA was inspired by Darwinian evolutionary theory, in which the survival of fitter creatures and their genes was simulated. The GA algorithm is a population-based algorithm [19]. Each solution represents a chromosome, and each parameter a gene. GA uses a fitness (objective) function to assess the fitness of each member in the population. To improve unsatisfactory solutions, the best solutions are picked at random using a selection mechanism (e.g., a roulette wheel or the mutation step used in Algorithm 2.1). Because the probability is related to the fitness, this operator is more likely to select the best solutions (objective value). The possibility of selecting inferior solutions also enhances the likelihood of avoiding local optima. This indicates that if good solutions become stuck in a local solution, they can be extracted using other solutions.

Because the GA method is stochastic, its dependability may be called into doubt. The technique of keeping the best answers in each generation and applying them to enhance subsequent solutions is what makes this algorithm dependable and capable of estimating the global optimum for a particular issue. As a result, the entire population improves generation after generation. Individual crossover results in the use of the 'area' between the specified two parent solutions. Mutation helps this method as well. This operator modifies the genes on the chromosomes at random, preserving the variety of the population and increasing GA's inquisitive behavior. The mutation operator, like nature, may produce a significantly superior solution and lead to the globalization of other alternatives [19].

2.2.4 Machine Learning

Machine Learning (ML) is a part of AI and in itself several techniques are encompassed such as Convolution Neural Network (CNN) and deep learning. ML is often described as categories of algorithms that allow the computer to make accurate and specific predictions. This type of AI focuses on the development of algorithms that receive data input and, through its analysis, predict an output. Some of the use cases of machine learning can be seen on image 2.3.

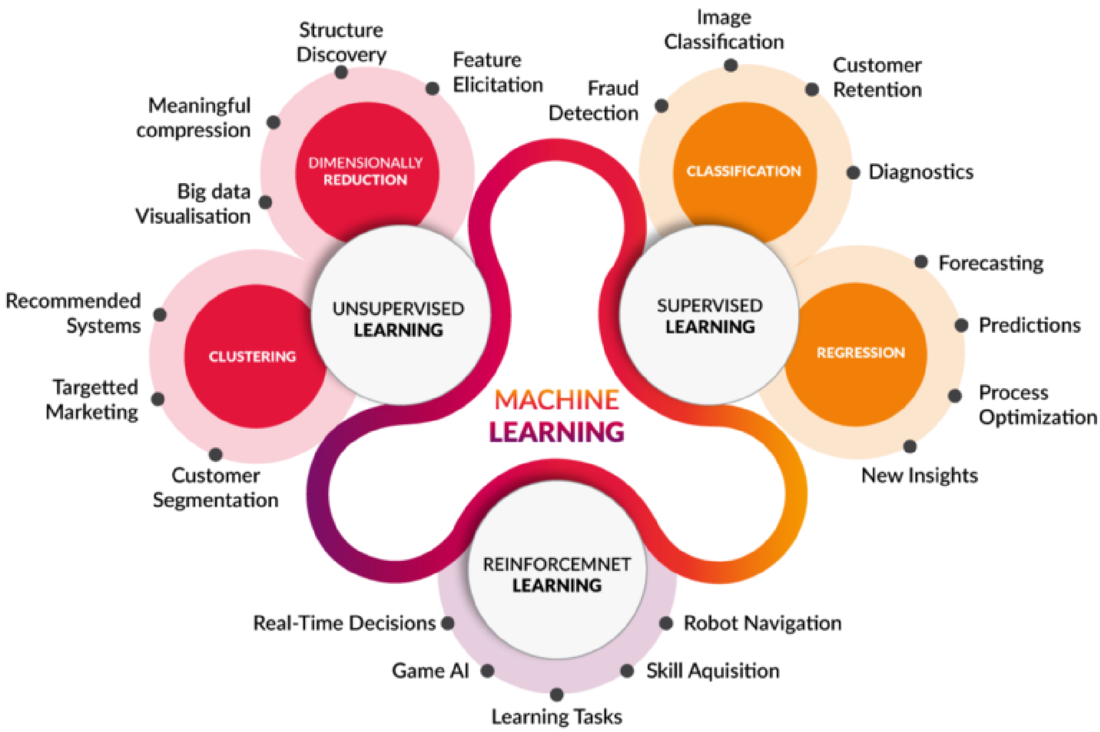

Figure 2.3: Branches of machine learning

Source: [20].

ML can be divided into different types of learning algorithms [21]:

- supervised learning;
- unsupervised learning;
- reinforcement learning;
- deep learning.

2.2.4.1 Supervised Machine Learning

Supervised ML is the construction of algorithms that produce patterns an hypotheses by using existent information (often labeled training data) and trends in order to predict the result for a given instance in the future. Supervised ML classification algorithms aim at categorizing data from the aforementioned prior information available to the model. Classification is a typical and very frequently carried out problem in data science. Several techniques exist in order to solve these problems such as Logic-based techniques, Instance-based techniques, stochastic techniques [22].

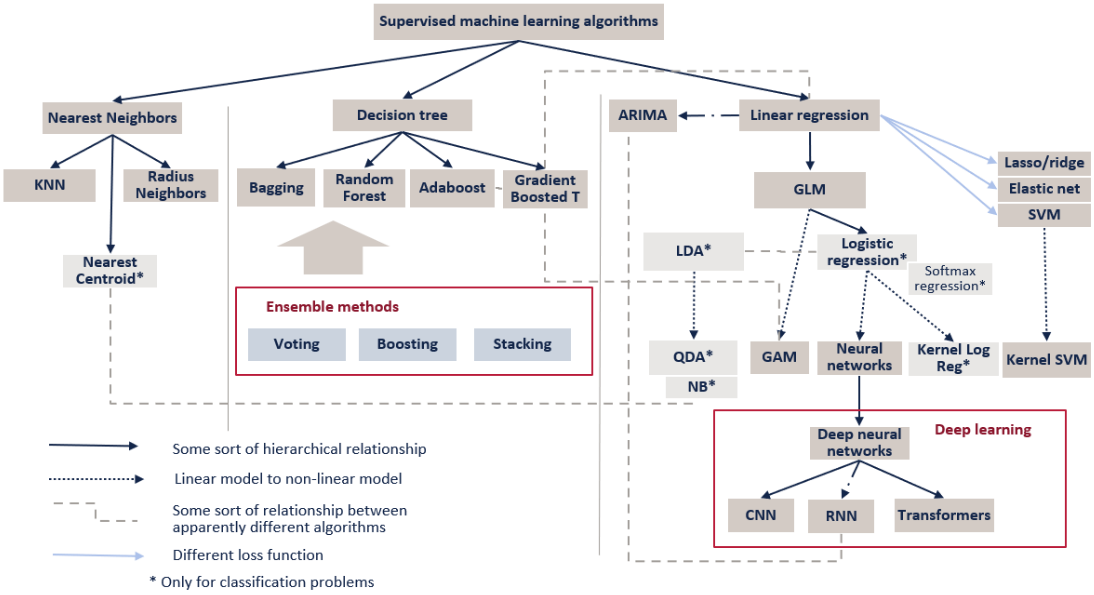

Figure 2.4: Types of Supervised ML

Source: [23]

In figure 2.4 it is possible to observe all of the different algorithms that exist using supervised learning as well as all the main concepts they are used for: nearest neighbors (clustering), decision trees and regression.

The main problems supervised learning faces surface mainly from the quality of the dataset as in classification problems with insufficient instances near the border can lead to an over fitting of the model, this problem also exists with different classes as they need a balanced amount of instances and they in themselves must be good representatives of the class they are representing. In cases with a big amount of information flowing into the system there also arises the problem of having to label, often by hand, all instances. Furthermore it also requires a lot of computational time [24].

2.2.4.2 Unsupervised Machine Learning

Unsupervised ML on the other hand involves pattern recognition without the need of labeled past data. Therefore this kind of ML uses pattern recognition without the need of a target attribute. This means all variables are used as inputs and that because of this approach it is especially use full for clustering and for discovering associations between the attributes[25]. One usage for such an algorithm can be, as a matter of example, creating the labeling the data so that it can be later feed into a supervised ML algorithm. That is, unsupervised clustering algorithms identify inherent groupings within the unlabeled data and subsequently assign label to each data value [26]. On the other hand, unsupervised association mining algorithms tend to identify rules that accurately represent relationships between attributes.

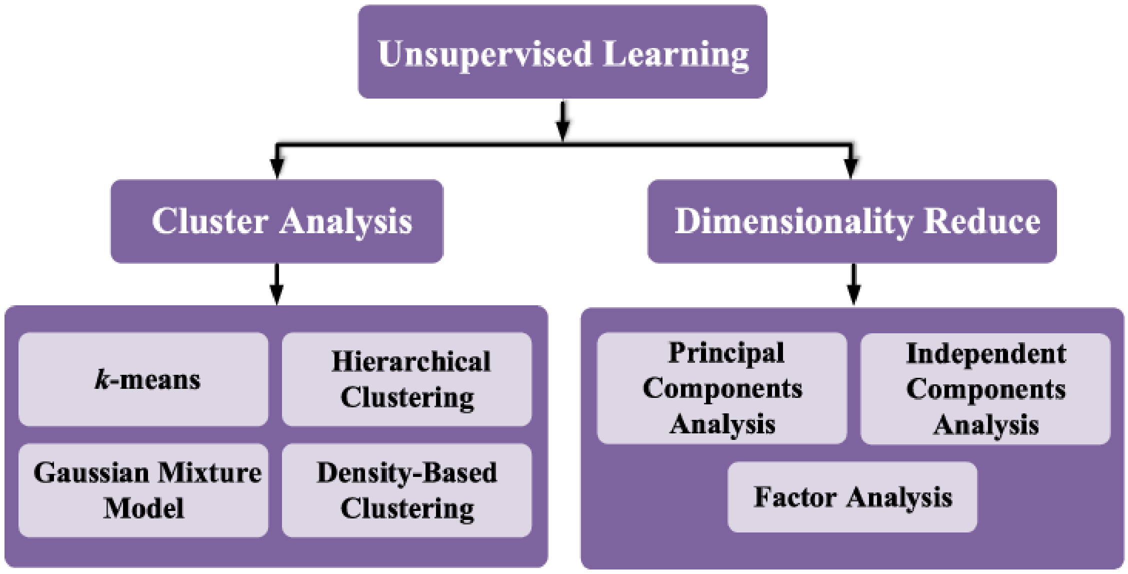

Figure 2.5: Unsupervised Machine Learning Algorithms

Source: [27].

In figure 2.5 it is possible to see the main types of unsupervised ML and the main algorithms that use this type of learning.

The main problems with unsupervised ML is that it is hard to get a high enough accuracy as the input data is not labeled by a human. Furthermore the classes these algorithms generate can not map well to informational or domain classes that exist in the real world. It also requires an analysis of the classes and labels as users need to interpret the classes. Properties can also change over time making it possible two instances have different class information.

2.2.4.3 Reinforcement learning

Considering the example of a game of chess, the number of combinations of all the pieces is extremely high so it is not fathomable to create such a complete set of previous data. In this case the agent can learn a transition model for its own moves and can perhaps learn to predict the opponent's moves, but without some feedback about what is good and what is bad, the agent will have no grounds for deciding which move to make [21]. In order for the agent to know if an action is good or bad it is needed to supply the agent write feedback when he does something that puts it closer to winning, or further away. Reinforcement learning algorithms work exactly by providing feedback on how a change in the world positively or negatively affects it, giving the user rewards or reinforcement respectfully.

In complex domains reinforcement learning is often the most feasible approach to train an agent to interact with the world and perform complex tasks such as playing games [21]. In this case it is very hard for a human to provide accurate and consistent evaluations of large numbers of positions, which would be needed to train an evaluation function directly from examples. Instead, the program can be told when it has won or lost, and it can use this information to learn an evaluation function that gives reasonably accurate estimates of the probability of winning from any given position. Similarly, it is extremely difficult to program an agent to fly a helicopter; yet given appropriate negative rewards for crashing, wobbling, or deviating from a set course, an agent can learn to fly by itself.

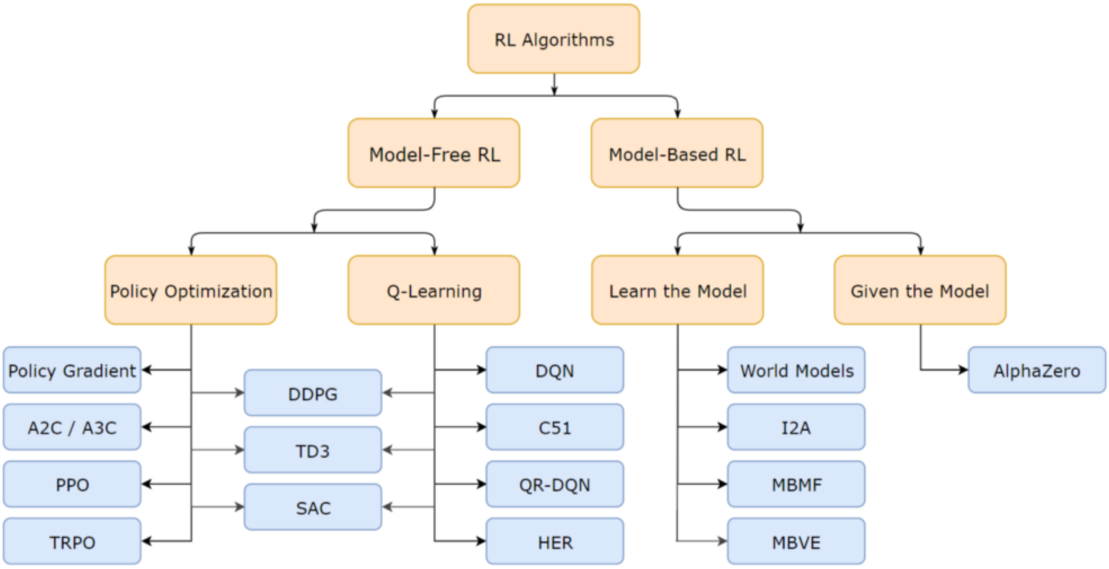

Figure 2.6: Reinforcement Machine Learning Algorithms

Source: [28].

In figure 2.6 it is possible to observe all the different types of algorithms used in reinforcement learning.

2.2.4.4 Neural Networks

Neural networks offer a variety of strong new ways for handling pattern recognition, data analysis, and control challenges. They have a number of distinguishing characteristics, including fast processing rates and the capacity to learn the answer to a problem from a series of instances. The bulk of real neural network applications presently employ one of two fundamental network models [29].

Artificial Neural Network

An Artificial Neural Network (ANN) is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives signals then processes them and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times [30]. A sample of the architecture can be seen in Figure 2.7.

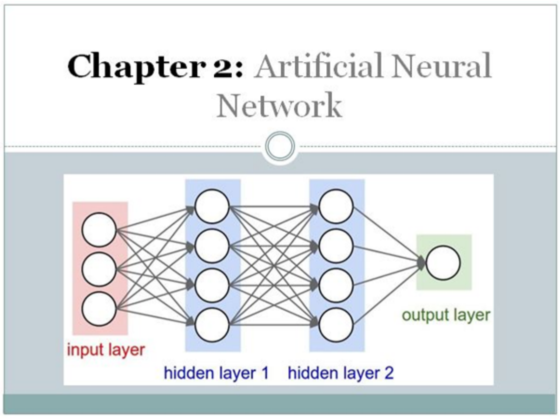

Figure 2.7: Architecture of an ANN

Source: [31].

Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process [32].

Learning is the adaptation of the network to better handle a task by considering sample observations. Even after learning, the error rate typically does not reach 0. If after learning, the error rate is too high, the network typically must be redesigned. Practically this is done by defining a cost function that is evaluated periodically during learning. As long as its output continues to decline, learning continues. The cost is frequently defined as a statistic whose value can only be approximated. The outputs are actually numbers, so when the error is low, the difference between the output (almost certainly a cat) and the correct answer (cat) is small [33].

The learning rate defines the size of the corrective steps that the model takes to adjust for errors in each observation [34]. A high learning rate shortens the training time, but with lower ultimate accuracy, while a lower learning rate takes longer, but with the potential for greater accuracy. Optimizations such as Quickprop are primarily aimed at speeding up error minimization, while other improvements mainly try to increase reliability. In order to avoid oscillation inside the network such as alternating connection weights, and to improve the rate of convergence, refinements use an adaptive learning rate that increases or decreases as appropriate [35].

While it is possible to define a cost function *ad hoc*, frequently the choice is determined by the function's desirable properties (such as convexity) or because it arises from the model (e.g. in a probabilistic model the model's posterior probability can be used as an inverse cost).

Back-propagation is a method used to adjust the connection weights to compensate for each error found during learning. The error amount is effectively divided among the connections.

Technically, backprop calculates the gradient (the derivative) of the cost function associated with a given state with respect to the weights [36].

Deep Learning

Deep learning is a subset of ML. It is a neural network with a large number of layers and parameters. Most deep learning methods use neural network architectures. Therefore it is also referred to as deep neural networks. Artificial Neural Network (ANN) are algorithms that mimic the inner workings of the human brain. Inspired by the sophisticated functionality of human brains where hundreds of billions of interconnected neurons process information in parallel, researchers have successfully tried demonstrating certain levels of intelligence on silicon. Examples include language translation and pattern recognition software [37]. In Figure 2.8 the architecture of a ANN is shown. It is of note that the there are several layers and parameters.

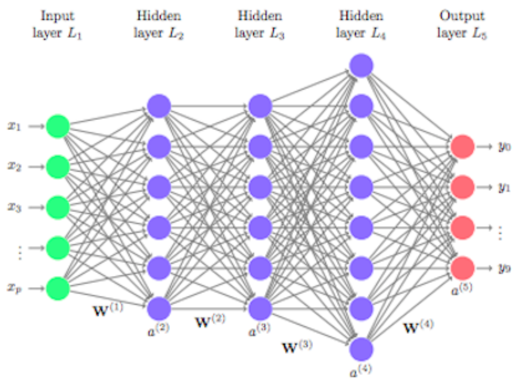

Figure 2.8: Artificial Neural Network Architecture
Source: [31]

Source: [31].

Deep Learning is a learning method that incorporates layers of neural networks in order to create knowledge from data. Deep Learning is particularly used when you want to extract patterns from unstructured data sets, as it simulates the functioning of the human brain to allow computers to deal with the same types of problems and abstractions [38]. Deep learning is widely used to create image, sound and computer vision recognition software [8].

The usage of deep learning is now wide spread and it is used currently used in several areas such as [39]:

- Virtual assistants;
- Translations;
- Vision for driver less delivery trucks, drones and autonomous cars;
- Chatbots and service bots;
- Image colorization;
- Facial recognition;

- Medicine and pharmaceuticals;
- Personalised shopping and entertainment.

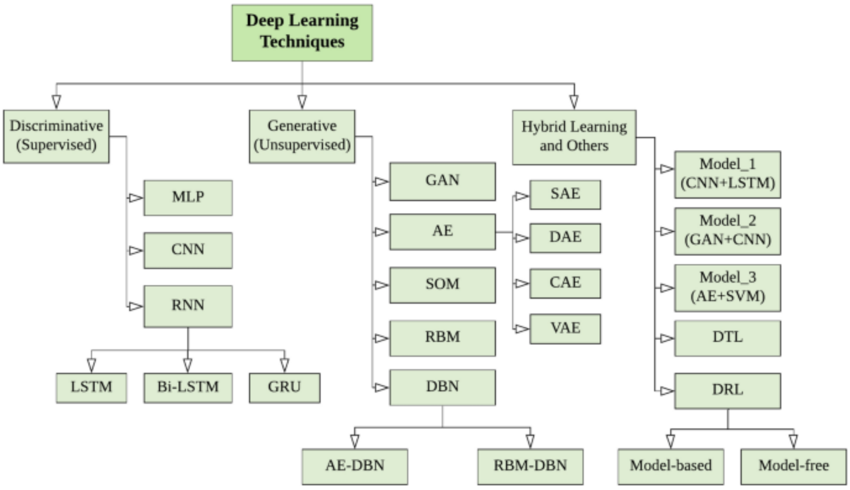

Figure 2.9: Architecture comparison of deep learning algorithms

Source: [40].

In figure 2.9 we can see the different techniques and algorithms that fall inside the specter of machine learning and the main branches of deep learning.

Some of the most used deep learning algorithms are Recurrent Neural Networks(RNN), Recursive Neural Networks(RvNN), Convolutional Neural Network (CNN) [41].

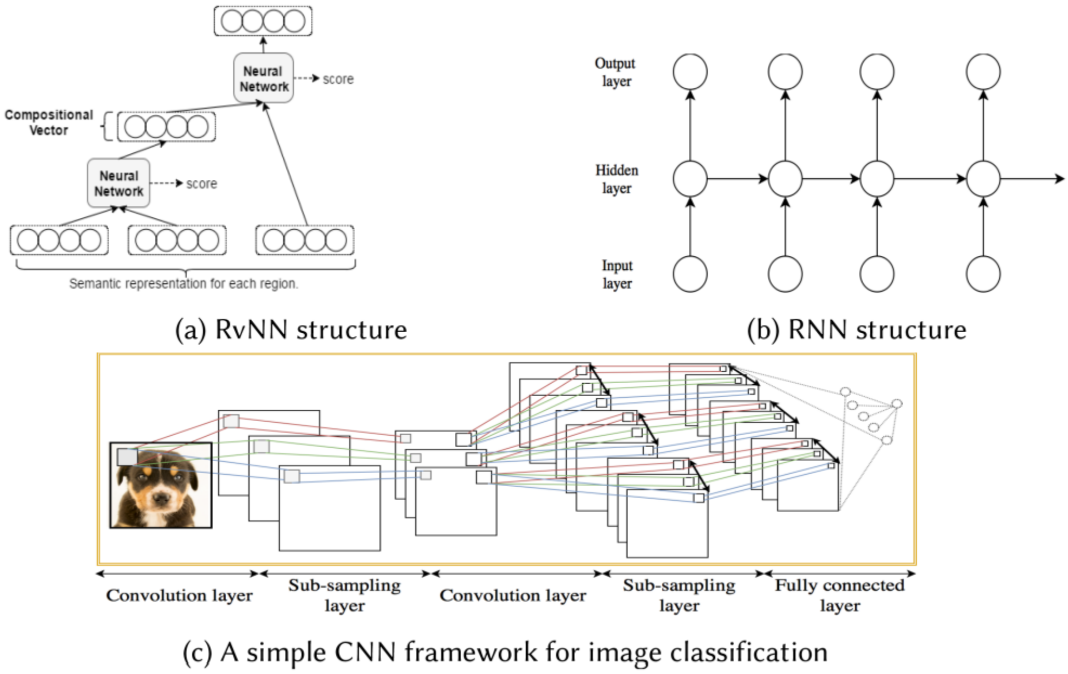

Figure 2.10: Architecture comparison of deep learning algorithms

Source: [41].

Figure 2.10 contains a comparison of the architecture of the three types of algorithms mentioned before hand. RvNN can produce hierarchical predictions as well as categorize outputs using compositional vectors. The creation of a RvNN was primarily motivated by Recursive Autoassociative Memory (RAAM), an architecture designed to analyze objects with arbitrary shapes, such as trees or graphs. The concept was to take a variable-size recursive data structure and build a fixed-width distributed representation. To train the network, the Backpropagation Through Structure (BTS) learning technique was used [42]. BTS takes a similar approach to the normal backpropagation method and can additionally support a tree-like structure. Autoassociation is used to teach the network to duplicate the pattern of the input layer at the output layer.

RNN is another frequently used and popular deep learning algorithm, particularly in NLP and voice processing [43]. Unlike standard neural networks, RNN makes use of the network's sequential information. This characteristic is critical in a wide range of applications where the embedded structure in the data sequence provides important information. For example, knowing the context is required to understand a word in a phrase. One major difficulty with RNNs is their sensitivity to disappearing and exploding gradients [44]. In other words, because of the multiplications of many tiny or large derivatives during training, the gradients may decay or increase exponentially. This sensitivity decreases with time, implying that the network forgets the original inputs as new ones enter. As a result, Long Short-Term Memory (LSTM) is used to address this issue by supplying memory blocks in its recurrent connections.

CNN is another prominent and commonly used deep learning algorithm [45]. It has been widely used in a variety of applications, including natural language processing [46] , voice processing [47] , and computer vision[48] , to mention a few. Its topology, like that of classic neural networks, is inspired by neurons in animal and human brains. It replicates the visual cortex of a cat's brain, which has a complicated series of cells. CNN has three major benefits: parameter sharing, sparse interactions, and equivalent representations. Local connections and shared weights in the network are used instead of standard fully connected networks to fully leverage the two dimensional structure of an input data (e.g., picture signal). This technique results in a network with fewer parameters, making it quicker and easier to train.

2.3 Result of RQ2- Fatigue detection

Fatigue is associated with change in the psychobiological state of a person, caused by relentless periods of arduous activity. It is one of the most common problems occurring amongst patients, in which an individual experiences resistance against an activity, mood swings and feeling lethargic[49]. It is a symptom that is mainly linked with illness, stress, aging and depression. Also, it is reported that one of the major symptoms in multiple sclerosis and Parkinson's disease, post-stroke, and Poliomyelitis is fatigue. Fatigue is mainly categorized as physical and mental fatigue. Physical fatigue is an ephemeral condition in which an individual is unable to maintain optimal physical performance, whereas mental fatigue is a psychobiological state induced by a prolonged period of demanding cognitive. In mental fatigue, an individual can experience a temporary decrease in cognitive performance which can manifest sleepiness, lack of energy, attention and sometimes an acute decline in cognitive performance might be observed as well. [49]

Fatigue is mainly categorized as active fatigue and passive fatigue. Active fatigue is a state caused by continual involvement in an arduous activity. Some characteristics like an increase in feeling of tiredness, an aversion towards a task or a decrease in mental and athletic performance can be observed. People who work long hours in grueling tasks experience this kind of fatigue.

A remote worker suffering from fatigue will not be as effective in performing the normal tasks associated with his everyday job, having a major effect in productivity especially in the pandemic and post pandemic situation, with a noticeable increase in the number of remote workers [50].

Currently fatigue detection techniques are being greatly improved because of their recent adoption into the central systems in the automotive industry. This industry is developing fatigue detection systems in cars as a counter measure to cut down on traffic accidents caused by drivers with excessive levels of fatigue. Fatigued drivers also cause in average more damages to property and a higher fatality rate then accidents caused by drivers in a normal state [51]. For this reason companies like BMW are creating fatigue detection systems and alertness detection systems and informing users when they are fatigued[52].

Currently several techniques exist to detect fatigue levels [53]:

- Fatigue detection using yawning as parameter;
- Fatigue detection using eye state;
- Fatigue detection using facial behavior;
- Fatigue detection using electrocardiogram signal.

2.3.1 Fatigue detection using yawning

Fan [54] suggested a method for tracking face movement with the help of a camera. During yawning, geometric features of the face changes so, yawning can be detected using these geometric features. This proposed technique was evaluated on 400 images and 20 videos consisting of 4512 faces. This method not only detects faces in instantaneously but also provides high accuracy. Gabor features were able to detect it with an accuracy of 96% whereas, the Geometric features were able to detect yawning with an accuracy of 69.5%. Hence the results of this experiment illustrate that Gabor coefficients are more efficient than Geometric features.

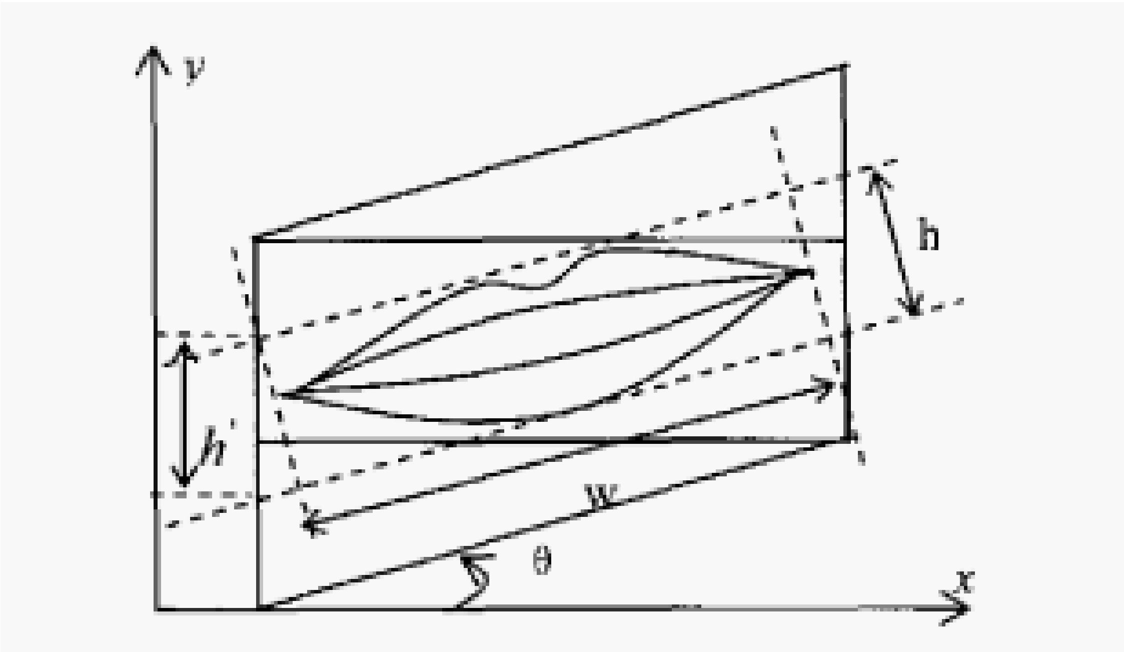

Figure 2.11: Mouth openness model

Source: [55].

Another method for yawning detection is calculating the degree of mouth openness. According the mouth model in figure 2.11, degree of mouth openness(DOO), is defined as follows:

$$DOO = \frac{h}{w} = \frac{h' * coS(\theta)}{W}$$

Where w is the width of mouth bounding rectangle, which is the distance between the two lip corners, and h is the height of mouth bounding rectangle, which is measured by the distance between two lines running through the upper and lower lip boundaries.

Knapic [56] also accomplished good results in yawn detection getting around 90% accuracy by using a thermal camera and applying successive masks to detect firstly the facial area and eye corners an then applied to detect yawning. These results are interesting because the usage of thermal cameras makes the system independent of the lighting of the room the user is in.

2.3.2 Fatigue detection using eye state

Wang [57] proposed a method for the detection of state of eye. There are mainly two challenges that are faced while detecting an eye's state: eye's position and light illumination. The eye-detection system has two main tasks, the first is open eye detection and the second is closed eye state detection. There are many methods developed in the past that were developed to detect eyes with geometrical features. So, in this paper, the author's main purpose was to detect eye state. Violae Jones method was used to detect eyes from the video frames captured by the webcam. AdaBoost algorithm was used to select the feature set and to train classifier. It ensembles weak classifiers and forms a strong classifier. Finally, to find the state of an eye a traditional binary image classifier was used. The above suggested method used 1500 open eye (positive)and 700 closed eyes (negative) with different illuminations for training. Table 2.8 illustrates the detection results of ensemble method.

Table 2.8: Table describing result of experiment

Source: [57].

Condition	**Accuracy overall**	**Accuracy Open eyes**	**Accuracy Closed eyes**
Indoor	99.17	97.34	97.11
Outdoor	89.51	96.11	98.5
Driving	81.86	93.3	81.25

Zhang [58] also realized an experiment into the detection of eye state in users with a focus on applying eye state detection to calculate percentage of eye lid closure over time(PERCLOS), a proven factor in determining fatigue in drivers.

He also went further with this information and calculated blinking frequency comparing it with standard values asserting that slower blinking frequency correlates to a higher degree of fatigue in the subject. When the subject is awake, the eye blink frequency is about 15 to 30 times per minute, about 0.25 to 0.3 seconds between blinks. When the blink frequency is less than 7 times per minute, the subject may be fatigued.

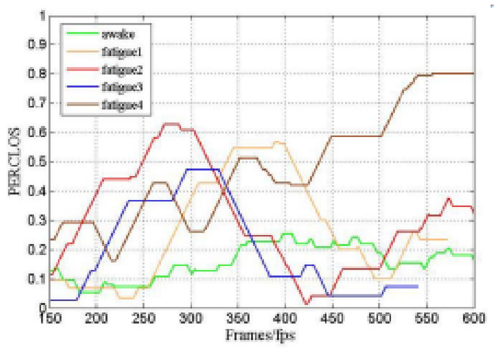

Figure 2.12: Percentage of eyelid closure over the pupil over time

Source: [58].

The experimental results show that the fatigue detection method can effectively detect the fatigue. However, when wearing glasses, the detection of eyes state is disturbed and the obtained accuracy was lower.

2.3.3 Fatigue detection using facial behavior

Dey [59] suggested a novel and economical method for real-time detection of eye state, yawning and head movement. The article suggested a pre-trained model which consists of Histogram-Oriented Gradient method (HOG) and SVM. The model assessed Eye Aspect Ratio (EAR), Nose Length ratio and Mouth Opening ratio. To detect the features in real-time, a camera was designed to take the video frame and a total of 68 data points were extracted from every frame. Along with this, a simultaneous model was used to evaluate the eye aspect ratio to detect drowsiness, nose length ratio for detecting head bending, and mouth opening ratio to detect yawning. These values are then compared with the threshold values from the original model. If these values are not in the specified range, it would be considered as an outlier thereby detecting fatigue. The accuracy for the detection of facial behavior with Functional Linear Discriminant Analysis (FLDA) and SVM were 93.3% and 96.4% respectively.

Irtija [60] also aimed to detect fatigue using facial expressions, more concretely the usage of facial landmarks. The resulting point cloud can be seen on figure 2.13. The main challenge of the method was to detect the subtle changes in features in a fatigued face in which respect the method achieves a satisfactory lever of success. The author used Cohn-Kanade method dataset and the Psychological Image Collection data set. He also used the concept of landmark detection for feature extraction and then used a SVM to detect fatigue from faces. Finally, the author achieved an 83% accuracy.

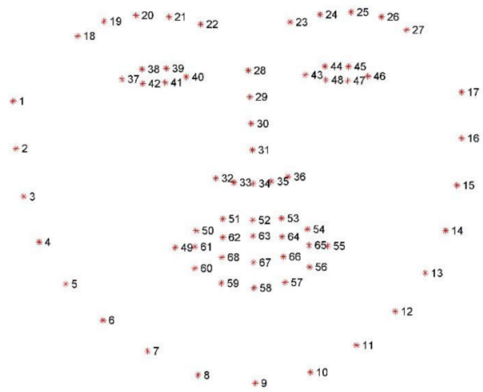

Figure 2.13: Facial landmark points

Source: [60].

2.3.4 Fatigue detection using heart rate

Heart rate variability is also a main factor to provide an insight into stress state, drowsiness or emotional reactions. There is a direct correlation in between the variation in heart rate and levels of fatigue in drivers.[61] used heart rate variability as a parameter to calculate fatigue, [62] monitored the heart rate of construction workers to search for a link between fatigue and a lower heart rate. KNN algorithms were used for classification. To see the impact of different values of K on the output of algorithms, the value of K started to increase from one. It was concluded that the accuracy of algorithm is increased as the value of K is enhanced. Based on the achieved results, it is conceived that monitoring of heart rate signals is an effective tool to assess physical fatigue in manufacturing and construction sites since there is a direct relationship between fatigue and heart rate features.

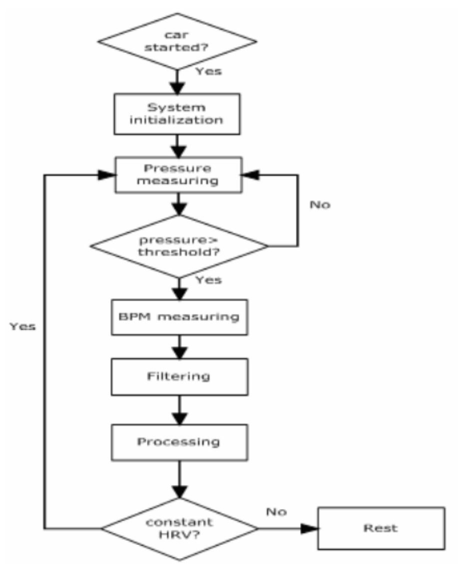

Figure 2.14: System Diagram

Source: [61].

In figure 2.14 it is possible to observe the flow of the software used by [61] and how the direct correlation between the heart rate variability and the level of fatigue was used to give feedback to a user about his level of perceived fatigue.

2.4 Result of RQ3- Fatigue detection systems

In this section some existing fatigue detection systems will be exposed as well as their contributions and shortcomings. The systems chosen were picked based on several criteria, namely the overall popularity(systems with over all more users or included in platforms with several users). It is of note that currently most of the major car manufactures are working on some kind of fatigue or drowsiness detection system some of which can be be seen in Table 2.9.

Current Technologies	Manufacturing Company	Monitoring Device	Detection parameters	Warning system	Detection category	Important features	Challenges
Driver Alert Control	Ford	Camera	Lane position	Audio vibration.	Vehicle based measure	1. A reverse steering is applied to direct the vehicle back into the lane	1. Apart from drowsiness, road side obstacle and rash driving are the other two main reasons of lane deviation
Driver monitoring system	Toyota	Charge coupled camera(CCD)	Eye tracking and head motion	Audio	Behavioural measure	1. Advanced Obstacle Detection (AOD) System pushes the brake automatically by tightening the seat belt during the chance of forward collision	1. Not feasible when the driver wears sunglasses or contact lenses 2. Nodding off has been considered as the final stage of drowsiness when thedriver falls asleep at the steering wheel 3. AOD system lessens forward collisions, but noprevention has been offered if the driver in the rear vehicle falls asleep
Attention Assist	Mercedes-Benz	Sensor on the steering column	Steering wheel movement and speed	Audio-visual	Vehicle based measure	1. Individual driving profile is created during the first few minutes of drive to be used as reference2. Driver's behaviour,road surface, weatherand period elapsed behind the wheel have been taken into account to check whether theerrors are due to drowsiness or not	1. Apart from drowsiness other factors such as side winds, road bumps and signal indicator may cause steering wheel movements
Driver Fatigue Detection	Volkswagen	Sensor on the steering wheel	Steering wheel movements	Audio-visual	Vehicle based measure		1. Different driving styles and road surfaces are the primary challenges that inhibit the implementation of this system
Driver Alert Control	Volvo	Camera and sensor for steering wheel movement	The distance between the road line marking and the car	Audio-visual	Vehicle based measure	1. Lane departure warning system prevents single vehicle running off the road crashes as well as head on collisions	1. Good lighting for the visibility of lane marking is required 2. Driving behaviour may not influenced by fatigue for professional drivers

Table 2.9: Commercial uses of fatigue detection systems

Source: [63]

The systems included in this revision of literature are:

- Don't sleep and drive – Volkswagen's fatigue detection technology [64];
- Saab's Driver Attention Warning System [65];
- Fatigue detection system for aircraft pilots [66];
- Fatigue detection system using heart rate [67];
- Fatigue detection system using ECG [67].

The first two systems included are developed by car manufacturers and are a feature of the companies ecosystem and operating system, thus these features are used by a vast array of users. This vast number of users makes these systems a popular choice. The third option contains a fatigue detection system currently used in aircrafts.

2.4.1 Volkswagen's fatigue detection technology

Human error was recognized as a contributing element in many public road accidents by Volkswagen(VW) researchers [64]. Human mistake is explained by flaws in perception, information interpretation, decision-making, information recall, and direct action. However, general physical and cognitive factors such as attention and weariness are crucial because they influence other cognitive processes. When drivers are fatigued, they fail to take the necessary precautions to avoid an accident.

The Volkswagen Driver State Monitoring System employs a non-intrusive observational technique, measuring blink rate, head movement (nodding), and eye pupil location. The device employs two dashboard-mounted cameras aimed at the driver's face, and a detection algorithm evaluates the data to estimate the likelihood of a micro-sleep or abnormal driving distraction [64]. If the system detects any of the three tiredness indications, it will inform the driver with auditory and visual alarms [68]. The VW type system is a useful and efficient strategy for treating tiredness in consumer automobiles on medium to long road journeys.

2.4.2 Saab's Driver Attention Warning System

The Driver Attention Warning System from Saab is a project that aims to combat the two most prevalent causes of road accidents: 1) driver tiredness and 2) distraction. The device makes use of two small infrared (IR) cameras that are aimed at the driver's eyes. It also makes use of the SmartEye [69] software to obtain precise eyelid, gaze, and head position data. The frequency of the driver's eye blinking is monitored in their algorithm. A pattern of long-duration eyelid closures signals the possible beginning of sleepiness [65]. For sleepiness detection, a three-level warning interface was created. This issue begins with a chime sound and text message, then progresses to a spoken message, and lastly, a harsher warning tone audio message is provided repeatedly until the driver pushes the reset button.

2.4.3 Fatigue detection system for aircraft pilots using Optalert

Several gadgets that have been designed to inform operators of the beginning of weariness are now on the market. These devices are intended to detect indicators of exhaustion such as changes in the pattern of head and eye movements, facial characteristics, and skin conductance [70]. Optalert is a product that monitors the velocity and length of eyelid closure and reopening while blinking. This personal safety gadget uses visual and audible signals to warn drivers if their tiredness exceeds certain limits. The technology has been proved to be successful in alerting vehicle drivers when they are sleepy in the road transport business. As a result, it has the potential to reduce traffic accidents when weariness may be a role. The Royal Australian Air Force (RAAF) Institute of Aviation Medicine (AVMED) has been collaborating with Optalert to develop variations of the Optalert device for prospective use in aviation [70]. This system was chosen for further examination because it appeared to have the potential to be developed into a gadget for use in aviation.

Systems such as Optalert, which offer an objective assessment of tiredness/drowsiness, are seen to have the potential to be integrated in aircrew fatigue risk management systems. The system might contain warning/alerting measures and be integrated with aircraft systems in a mature configuration. A self-powered device might deliver visual or audible alerts independent of aircraft systems, providing real-time input to a pilot as an intermediate alternative. As a consequence of this research, it was determined that more research into the usage of Optalert in airplanes was required.

2.4.4 Fatigue detection system using heart rate using SmartCap

Smart Cap. The first production version of this system was distributed throughout an entire Australian mine site (with roughly 200 miners) in late 2011. EEG sensors are inserted beneath either a baseball hat with a sweatband or a headband, which is connected to an electrical processing card. The data is wirelessly relayed to a tiny dashboard-mounted monitor, and it may also be transferred to a central dispatcher for remote monitoring. The Oxford sleep resistance test was used to calculate threshold scores (OSLER). Smart Cap triggers an alert when it reaches a threshold of four, which is comparable to four consecutive OSLER lapses and indicates micro-sleeps. If the cap is attached but not worn for a prolonged length of time, an alert will sound [67].

Monash University researchers' preliminary (unpublished)[67] validation results indicate high sensitivity (.95) and specificity (.82) for identifying four consecutive lapses on the OSLER test [67]. More study is being conducted to see whether the Smart Cap can detect lower levels of weariness (Kevin Greenwood, personal communication).

2.4.5 Fatigue detection system using ECG using B-Easy

EEG sensors are incorporated in a headband-like device, and data is relayed through radio waves through a tiny unit affixed to the back of the head. EEG signals are categorised into four stages of awareness (sleep onset, relaxed wakefulness, low engagement, and high engagement) based on cognitive and maintenance of wakefulness tests (MWT) performed on awake and sleep-deprived people [67]. B-Alert levels significantly decrease with prolonged wakefulness and are positively correlated with PVT lapses, subjective sleepiness, objective sleepiness on memory test performance, driving simulator performance, human scored EEG activity, and observed sleepiness (e.g., facial expressions, head nodding) in sleep-deprived healthy subjects, according to device developers' validation studies [67].

2.4.6 Comparison of Fatigue Detection Systems

The aforementioned systems use several approaches in order to try and be more accurate in the prediction of fatigue, with the usage of one primary mean of information intake (for example both Saab's system an Volkswagen use a camera as a primary information intake).

These systems contain a camera and use it in order to calculate different metrics such as eye gaze and head position, this makes the system dependant on one technology unlike the solution proposed in this thesis. A comparison of the studied Fatigue Detection Systems can be seen on Table 2.10.

Table 2.10: Comparison of the studied systems

	Application	Sensor	Commercially available
Volkswagen	Car	Camera	No
Saab	Car	Camera	No
Optalert	Car, Plane	Camera Galvanic Skin Response(skin conductance)	Yes
SmartCap	Mining equipment	Heart Rate sensor	Yes
B-Easy	Generic	ECG	Yes

2.5 Conclusions

In this section the state of the art of AI was explored as well as several areas of study that compose it. The usage of fatigue detection systems was also explored as it is a technology that is widespread in the automotive industry nowadays, with the assistance of a small camera in the dashboard aimed at the driver many companies like BMW can give drivers information about their fatigue level [52]. Being a prevalent technology in this day this problem as been suffering iteration over iteration and several techniques have appeared over time.

Given the current relevance of fatigue detection systems, mainly included in the software of vehicles, this is a topic that has been developed and studied and even though several approaches were taken in the extent of creating a fatigue detection system, as far as the research shows the system developed in this paper innovates by using more than one technique to predict the level of fatigue in the user. The systems analysed either used input from a camera or input from other sources(EEC or ECG for example), whereas the system proposed in this thesis analyses both the camera and the heart rate of the user thus outputting a more accurate estimate of the real level of fatigue.

Chapter 3

Proposed Solution

In this chapter it will be explored the proposed solution and architecture of all the necessary components of the solution as well as the interactions the system has with the user and the desired output of the system.

3.1 Description of the solution

The proposed solution to resolve this issue is an application that can make use of a laptop webcam and a heart rate monitor in order to give the most accurate prediction for the perceived level of fatigue in an user. This choice was made as this solution aims to help remote workers and as such the solution should be incorporated into everyday objects the user already possesses and not by adding more complexity to the home office of the user.

This solution also took into consideration the application the user currently has open in his computer in order to establish correlations and provide the user with enough insights to be able to better his workflow and productivity throughout the day.

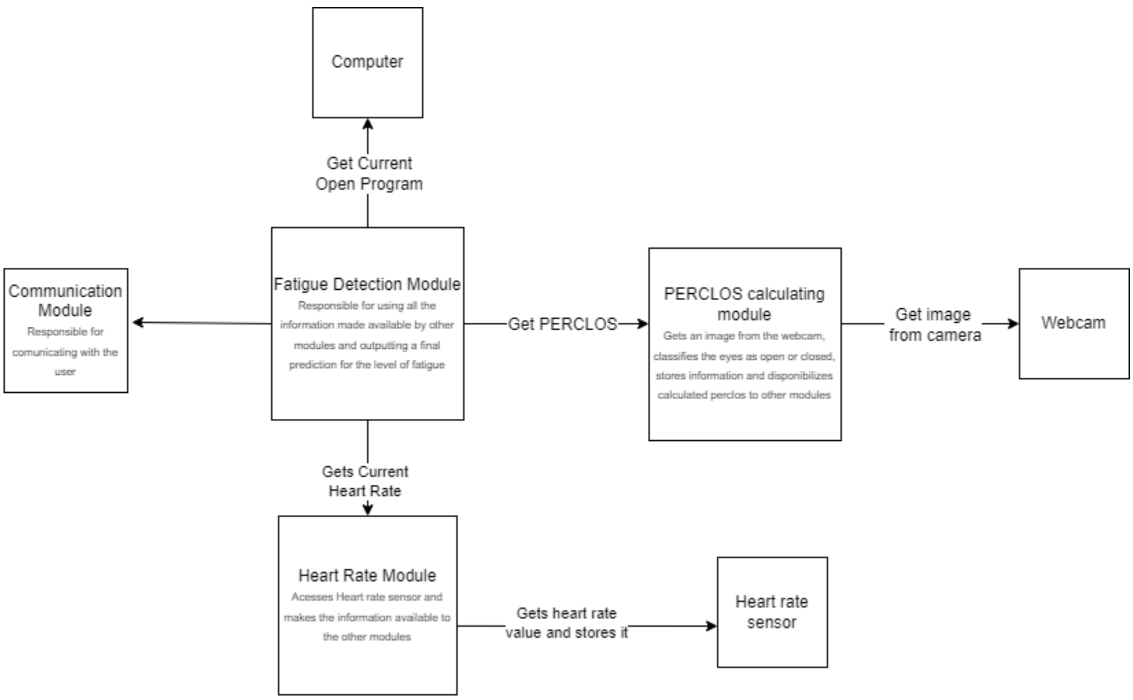

Figure 3.1: Container diagram of proposed solution

In figure 3.1 a diagram of the solution can be seen. In it we can see the four main components as well as the interactions between them. It is of note that the proposed system is made of four main components:

- PERCLOS calculating system;
- Heart rate monitoring system;
- Communication System;
- Fatigue prediction system.

These modules were separated in order to increase modularity and allow each module to have only one responsibility. The PERCLOS calculating system is responsible for accessing the camera of the user as well as perform all calculations necessary to calculate PERCLOS, the heart rate monitoring system retains the responsibility of accessing the heart rate of the user and of making such information available to other modules, the communication system is responsible for handling all the flow of information from the system to the user and the fatigue prediction system is responsible for calculating the final value of fatigue by consuming information from the other modules.

3.1.1 PERCLOS calculating system

In Section 2.3.2 a fatigue detection system using eye state was studied. The goal of this system was to create a model able to detect eye state via image processing [57]. With the classification of the eye as open or shut it is possible to calculate the Percentage of eyelid closure over the pupil over time (PERCLOS) of the user. This value has been observed as a good metric while calculating fatigue [57] and does not require any extra equipment as the camera present on a laptop can be used for the eye state classification.

Utilizing the webcam present on the computer an Artificial Intelligence model classifies the eyes of the user in one of two states, open or closed. This one of the responsibilities of this module, the development of a model capable of analysing the image and classifying the eye as open or close. Alongside the classification a timestamp of the classification made is also saved. With the classification and this timestamp it is possible to calculate the PERCLOS of the user as the percentage of the time the eye is closed for a certain time interval can be calculated by dividing the number of images classified as closed by the total amount of images(open + closed).

This module must be able to do all the processing necessary to calculate the PERCLOS of a person, thus it will also be responsible for all the image processing done to the image in order do feed the image to the model as well as provide a visual interface that can be used during testing of the application. This visual interface is to be constituted of a feed of the original image of the camera with the eyes highlighted(inside a square to demarcate boundaries) and text with the prediction made by the model. The interface must also have a label that shows the current PERCLOS.

Figure 3.2 represents a mock up of the proposed user interface of the PERCLOS calculating system featuring all the requirements presented beforehand. With this user interface it is possible to test the calculation of the PERCLOS as well as perform real usage tests of the model responsible for the categorization of the eyes as open or closed, essential to the calculation of PERCLOS.

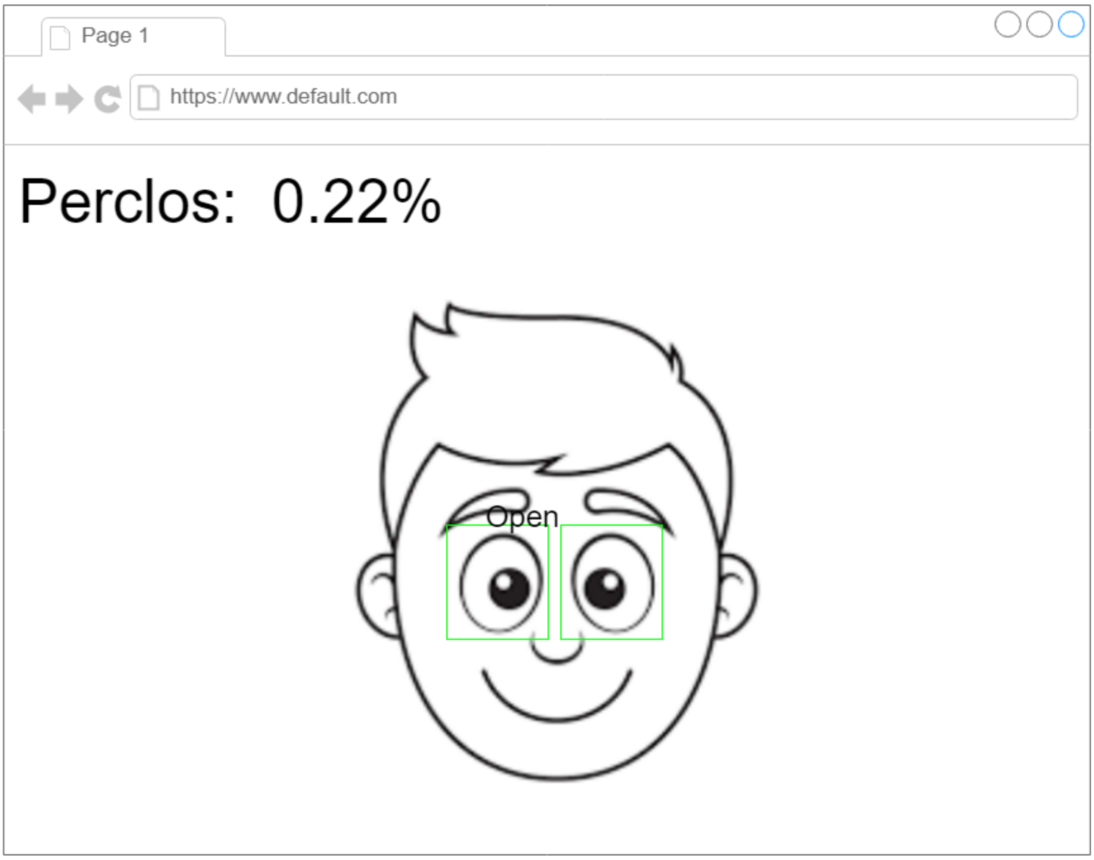

Figure 3.2: User interface mock up of proposed solution

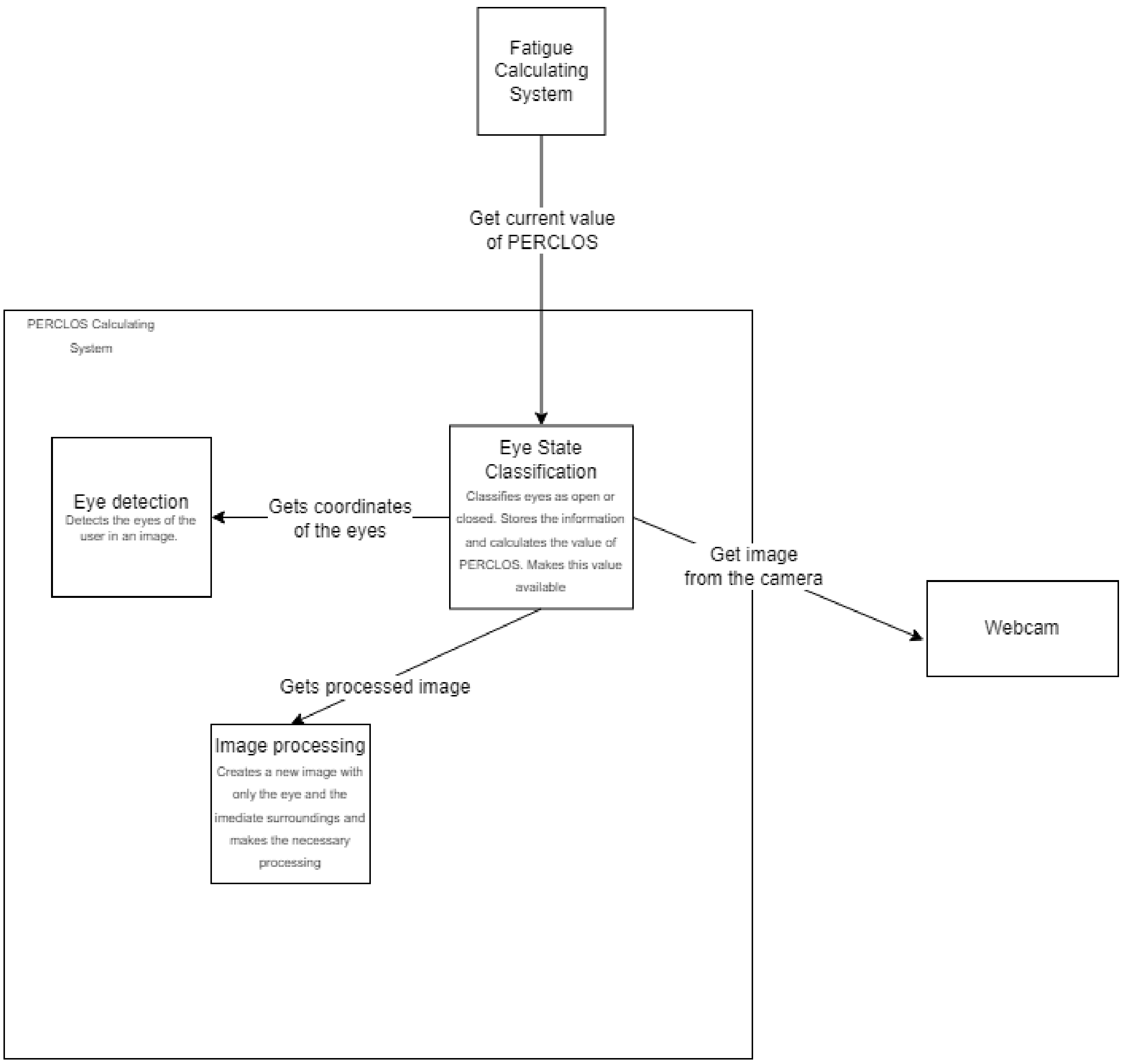

Figure 3.3: Component diagram of the perclos calculating system

In figure 3.3 it is possible to observe the components inside the PERCLOS calculating module and the interactions between them. As it is possible to see the solution is composed of two main parts, the model responsible for the detection and classification of the eye state and the one responsible for getting the image from the camera buffer, doing all the pre processing required for the machine learning method to be successful. The pre processing the images were subject to will be further developed in the upcoming chapter.

3.1.2 Heart rate monitoring system

In Section2.3.4 the viability of the usage of an heart rate monitor as an indicator was studied. [61] used a modified steering wheel in order to measure the heart rate of drivers through their hands, a solution that was not easily applied to this situation as this proposed solution had the requirement of being able to be easily applied to the remote working environment. The proposed solution proposes the usage of a smartwatch, a piece of wearable technology often used for tracking exercise.

The usage of this kind of wearable device is becoming ever more widespread as we can see in Figure 3.4. The existence of such a widespread availability of smartwatches presents itself as a solution as to what kind of device to use in order to collect heart rate information from an user whilst disrupting the workers workspace as little as possible, using a device that in all odds, he already possesses.The inclusion of an heart rate monitor on a smart watch in today's market is practically a given as this sensor is included in practically all the smart watches in the market.

There have also been studies into the accuracy of sensors in smartwatches revealing that there is not a significant difference between the accuracy of standalone sensors used in the medical field and those present in smartwatches [71].

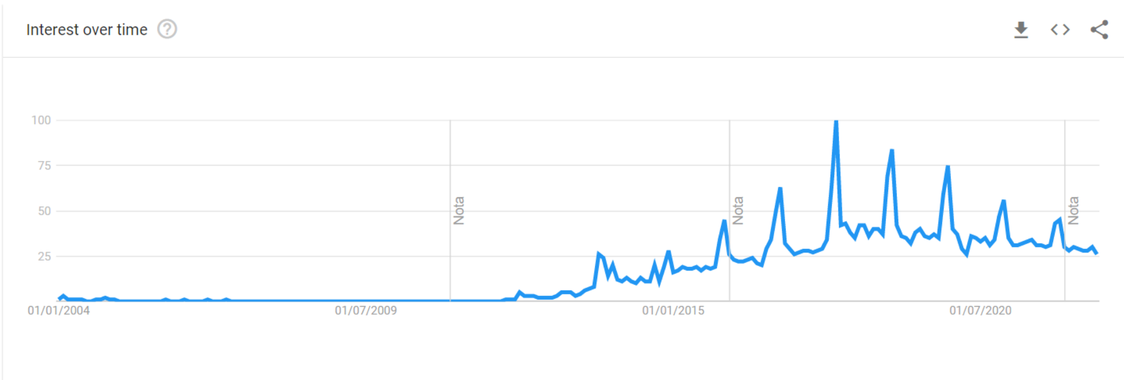

Figure 3.4: Google trend search of smartwatches

This component in the overall architecture is responsible for measuring the heart rate of the user but it also needs to accomplish another task in order to assure all other components work properly, this component also needs provide the data to the other components so that the main module can get the current heart rate of the user and use it in order to predict fatigue.

Figure 3.5 shows the inner workings of the heart rate monitoring module. This module shows the existence of a custom layer between the system and the smartwatch as a specific app must be developed in order to both show the user notifications and access the sensors in order to return the current heart rate of the user. This app has the responsibility of encapsulating the smartwatch software and communicating with the main application via requests.

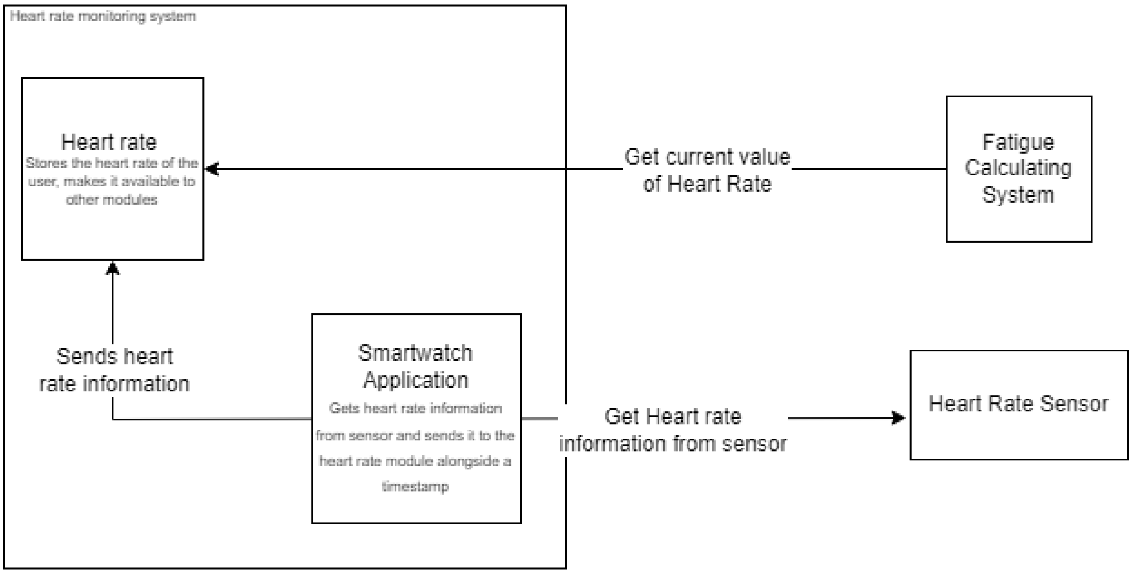

Figure 3.5: Component diagram for the heart rate monitoring module

3.1.3 Communication system

The necessity of a communication system pertains mainly with the necessity of communicating with the user and to provide tips and relevant information. This communication is a necessary for the system to be complete as well as the way the system provides feedback and tips to the user. To this effect this module was designed to send pop up notifications when the system detected a high level of fatigue.

The goal of this type of communication is to warn the user once a certain threshold of fatigue as been hit so that he can take appropriate action. This kind of message needs a mean of communication that is fast and the user should get it as soon as it is delivered. This kind of notification fits nicely with a pop up notification. Since in the previous section the usage of a smartwatch was discussed this presents itself as an opportunity to use this technology again. This notifications can be implemented in the smartwatch to allow users to get information and tips as soon as possible so that they can take action as soon as possible.

As was previously mentioned this notification appears on the smartwatch of the user. In order for this functionality to work a layer is necessary between the smartwatch operating system and our application. With this purpose a custom smartwatch application was developed in order for the user to receive the information.

In Figure 3.6 a component diagram of this module can be seen as well as the interactions between the smartwatch application and the system.

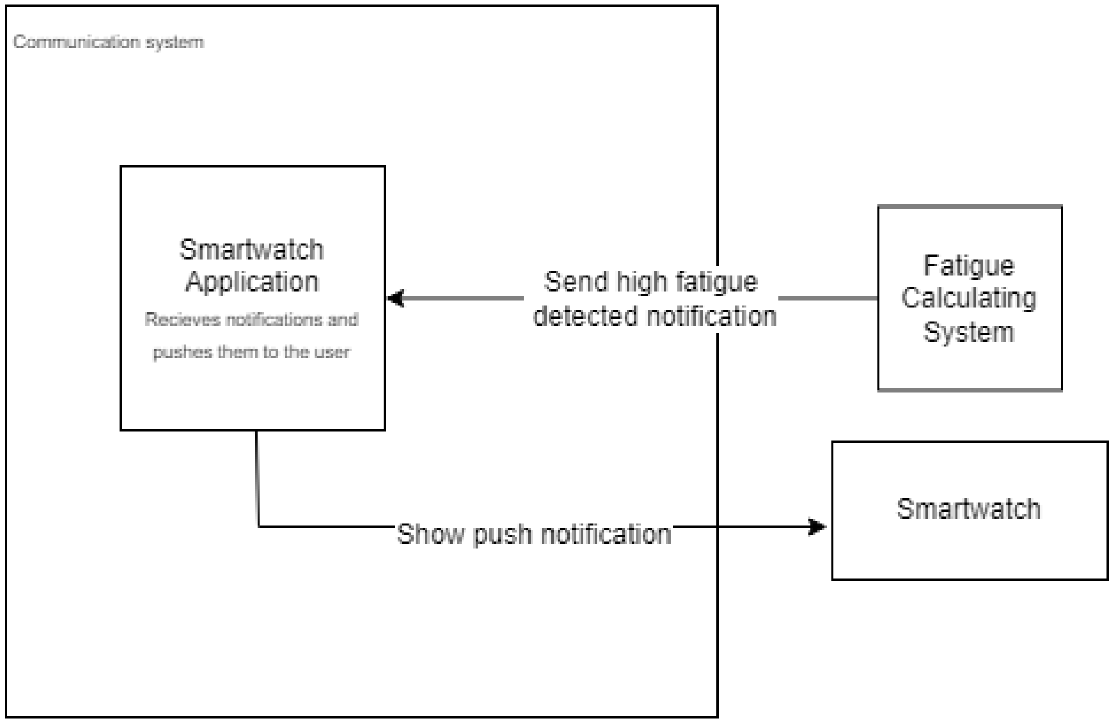

Figure 3.6: Component diagram for the communication module

3.1.4 Fatigue detection system

This module encapsulates the main functionality of the system: predicting the perceived level of fatigue in the user. This module is responsible for predicting the level of fatigue by using information gathered by other modules. This module utilizes a model trained to predict fatigue by analysing the heart rate of the user, the PERCLOS calculated beforehand , the application the user current has open and the timestamp in which this occurs.

In order to provide predictions of the level of fatigue in an user a model is to be train to utilize PERCLOS, the heart rate, timestamp and the program the user has open to output as accurate a prediction as possible for the users perceived level of fatigue.

This module is also responsible for generating the two types of communications discussed beforehand. Once the user surpasses a certain level of fatigue a notification is dispatched to the communication module and on a pre determined schedule a newsletter is generated with the insights this module was able to create. These insights are created with the intention of educating the user, therefore this insights are mainly correlations between activities or time of the day and levels of fatigue.

3.2 Conclusion

The system developed in this solution is composed of four modules. The PERCLOS module is responsible for using the webcam to get an image and calculating the value of perclos. The Heart Rate Module reads the heart rate of the user and makes this information available to the rest of the system. The communication Module is responsible for warning the user when fatigue levels get two high. Finally the fatigue detection system is responsible for giving a final prediction for the level of fatigue of the user.

The system outlined in this chapter allows the usage of two different fatigue detection techniques, using the Percentage of eyelid closure over the pupil over time as well as the heart rate of the user to give a prediction of the level of perceived fatigue. This chapter also outlined in a high level the necessary functionalities of the modules and the necessary AI models. In the next chapter these modules will be further explored.

Chapter 4

Development of the solution

In this chapter the developments required for the system as a whole to function will be explored. This chapter will do a deep dive on each of the modules specified in the previous chapter with a special focus on the machine learning methods and the steps taken to ensure these were as reliable and accurate as possible, whilst also focusing on the developments needed for the remainder of the system such as the development of the communication module.

4.1 PERCLOS calculating module

The PERCLOS calculating module is constituted mainly of 3 parts all working towards the same goal, getting an image of the user from the camera and calculating the final value of the PERCLOS:

- Eye detection;
- Image processing;
- Eye state classification.

In Figure 4.1 the representation of a generic use case of this module is portrayed. As can be seen each component in this module has a responsibility: the eye detection module analyses the image and returns the position of the eyes, the image processing module performs the necessary transformations and the Eye State classification is responsible for getting the image from the webcam and by utilizing the other modules classify the eyes as open or closed.

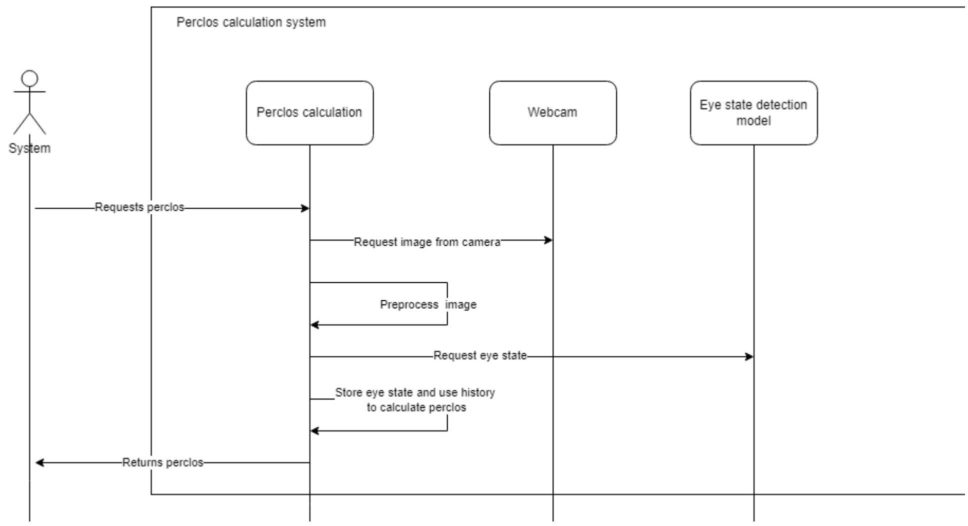

Figure 4.1: System sequence diagram of PERCLOS calculating module

In Figure 4.2 the flowchart of calculating PERCLOS can be seen. After getting the image the eyes are detected and the image is processed. Afterwards the eye state is classified and the final value PERCLOS calculated. The eye detection is responsibility of the Eye detection component and the image processing part of the Image Processing component, the rest of the actions are part of the Eye state classification module. In the following sections the flow of an image in the system is going to be explored, starting with an image being captured and the eye detection component detecting the positions of the eyes. The following sections will also be exploring the same image going through the system.

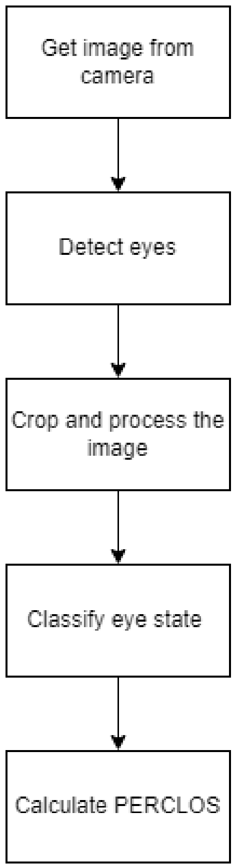

Figure 4.2: Flowchart of the PERCLOS module

4.1.1 Eye detection

As previously stated this module is responsible for identifying the positions of the eyes from the image captured by the webcam. The example being used will be part of the same flow of information previously discussed.

In order to feed the eye state classification there was a need to find in the image where said eyes are. In order to accomplish this result a library called OpenCV (Open source computer vision) was used. OpenCV is an open source library for image and video analysis [72] , originally introduced more than decade ago by Intel. Since then, a number of programmers have contributed to the most recent library developments. Nowadays the library has more than 2500 optimized algorithms. It is extensively used around the world, having more than 2.5M downloads and upwards of 40K people in the user group [73]. Regardless of whether one is a novice programmer or a professional software developer, unaware of OpenCV, the main library content should be interesting for the graduate students and researchers in image processing and computer vision areas.

Module	Description
Core	Core data structures, data types and memory management
Imgproc	Image filtering, geometric image transpowmations, structure, and shape analysis
Highgui	GUI, reading and writing images and video
Video	Motion analysis and object tracking in video
Calib3d	Camera calibration and 3? reconstruction from multible views
Features2d	Feature extraction. description and matching
Obldetect	Object detection using cascade and histogram of gradient classifiers
ML	Statistical Models and classification algorithms for use in computer vision applications
Flann	Fast Library for approximate nearest neighbors - fast seaches in high dimensional spaces
GPU	Parallelization of selected algorithms for fast execution on GPU
Stitching	Warping, blending and bundle adjustments for image stitching
NonFree	Implementations of algorithms that are patented in some contries

Table 4.1: OpenCV modules

Source: [74].

OpenCV was chosen because of the vast array of functionality it has as can be seen in Figure 4.1. The adaptability of this library allows for the usage of both classifiers and image processing tools, both necessities in this thesis.

As stated OpenCV has a vast array of algorithms and is vastly used, one of such algorithms is an object classifier that can be configured to identify eyes by using a cascade classifier as can be seen in source code bellow.

```
gray_fr = cv2.cvtColor(fr, cv2.COLOR_BGR2RGB)
eyeCascade = cv2.CascadeClassifier(cv2.data.haarcascades + '
    haarcascade_eye.xml')
eyes = eyeCascade.detectMultiScale(gray_fr, 1.1, 4)
```

Listing 4.1: OpenCV eye detection

Another big advantage of choosing this library is the existence of methods that allow to alter the image on the fly without much hassle. This presents itself as a big advantage in this case as there was a desire to also create an interface to test the software as well as graphically show the detections and after the integration with the model responsible for eye state classification and PERCLOS calculation as described on Figure 3.2.

In order to fulfill the interface requirements advanced in section 3.1.1 the OpenCV library was used to identify the coordinates of eyes in the image and draw boxes around the eyes. The result of this process can be seen on Figure 4.3.

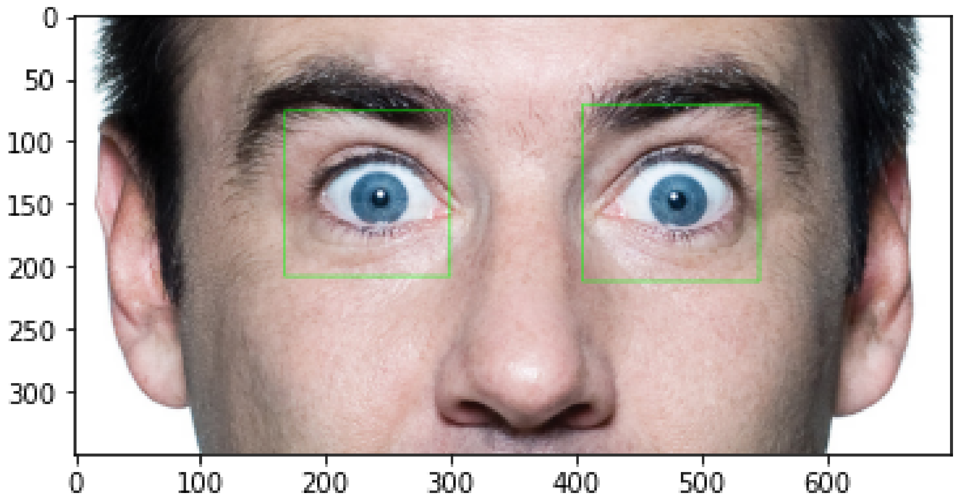

Figure 4.3: Eye detection using OpenCV

4.1.2 Image Processing

As previously stated this module is responsible for doing the necessary preprocessing of images and cropping the image so that the eye is the only thing in it. The images used in this section will be the sequence of transformations of Figure 4.3.

In order to feed the eye state classification model clear images of eyes it is necessary to do some preprocessing to the images. The first step to achieving this goal is to isolate the eye which was accomplished using OpenCV. By using the aforementioned Cascade Classifier OpenCV can find and return the coordinates of the eyes as well as create a new image from this coordinates. In Figure 4.4 an example of the result image can be seen(transformation of the image in Figure 4.3), it is of note only the eye and a small subset of its surroundings can be seen in the image.

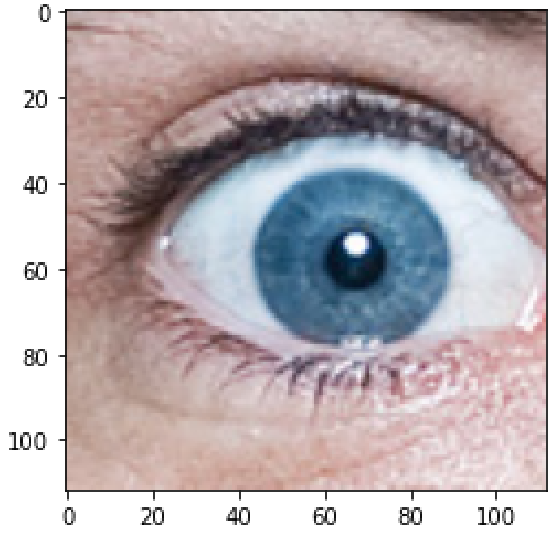

Figure 4.4: Eye detection using OpenCV: single eye

After being submitted to the crop the new image is also processed in order to transform it into a black and white image. This process was also done via the OpenCV library.The necessary code can be seen in Listing 4.2.

```
img_array = cv2.imread(file, cv2.IMREAD_GRAYSCALE)
backtorgb = cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
new_array = cv2.resize(backtorgb, (img_size,img_size))
```

Listing 4.2: OpenCV image transformation

Figure 4.5 shows the result of applying the snippet of code in Listing 4.2 to Figure 4.4.

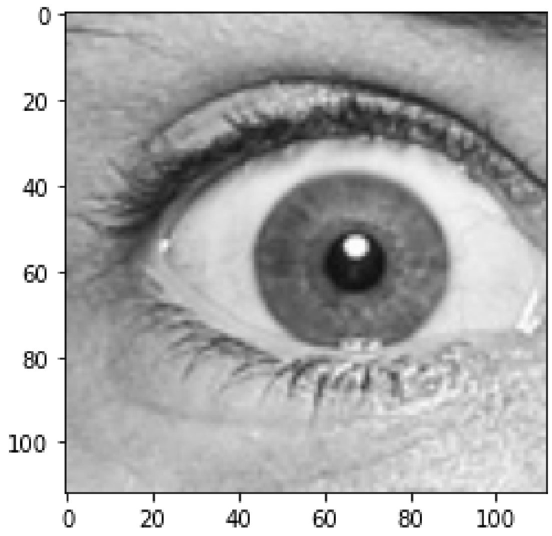

Figure 4.5: Eye detection using OpenCV: single eye after preprocessing

4.1.3 Eye state classification

As previously stated this module is responsible for getting using the other components to get an image of the eye and classify the eye as open or close as well as calculating PERCLOS. In order to do this this module gets an image from the camera, calls the Eye Detection module and with the information it provides calls the Image processing module. By doing this this module has access to the image of the eye cropped and ready to use.

After this it was necessary to create a model capable of classifying the eye state of the user.

4.1.3.1 Dataset

In order to detect fatigue one of the proposed methods was using the eye state of the user in order to calculate blinking times and detect fatigue from there as someone who is tired will blink in a slower rhythm often indicates a higher degree of fatigue.

The chosen data set was MRL Eye Dataset [75]. This data set is constituted by 84,898 images of pupils and has been cited in several papers. 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. In order to simplify the comparison of algorithms, the images are divided into several categories, which also makes them suitable for training and testing classifiers.

This dataset contains images of both men and woman and has annotations regarding the gender, whether the user was wearing glasses, the eye state, the presence of reflections, the light condition and the resolution of the camera used to take the photo.

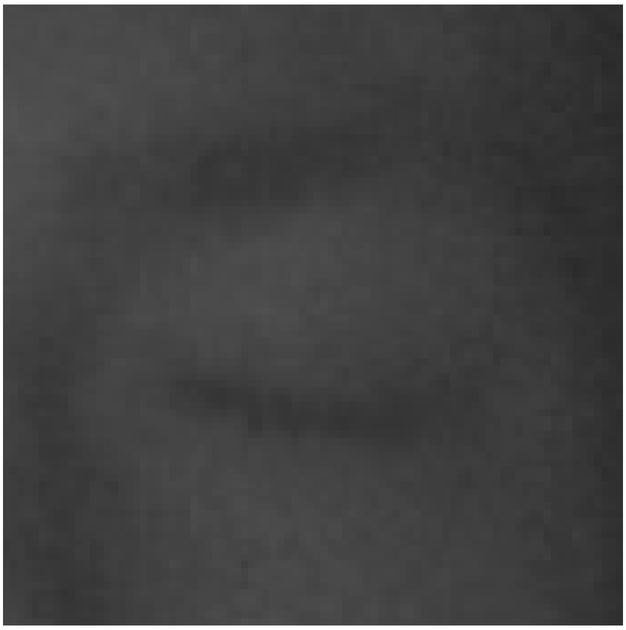

Figure 4.6: Example image from dataset

In Figure 4.6 we can see an example of one image in the dataset.

In the dataset, we annotated the following properties (the properties are indicated in the following order):

- Subject ID; in the dataset, we collected the data of 37 different persons (33 men and 4 women);
- Image ID; the dataset consists of 84,898 images;
- Gender [0 - man, 1 - woman]; the dataset contains the information about gender for each image (man, woman);
- Glasses [0 - no, 1 - yes]; the information if the eye image contains glasses is also provided for each image (with and without the glasses);
- Eye state [0 - closed, 1 - open]; this property contains the information about two eye states (open, close);
- Reflections [0 - none, 1 - small, 2 - big]; we annotated three reflection states based on the size of reflections (none, small, and big reflections);
- Lighting conditions [0 - bad, 1 - good]; each image has two states (bad, good) based on the amount of light during capturing the videos;
- Sensor Id; id of the camera that took the photo.

4.1.3.2 Model

The goal of this model was only one, to receive an image of an eye and classify it as open or close. To achieve this, transfer learning was utilized. Transfer learning is a technique used to improve a learner from one domain by transferring information from a related domain. We can draw from real-world non-technical experiences to understand why transfer learning is possible [76]. This technique, much like deep learning is modeled by nature, more specifically humans, as one person is able to take information from a previously learned task and use it in a beneficial way to learn a related task for example it is easier for a person that can ride a bike to learn how to ride a motorcycle than one that does not.

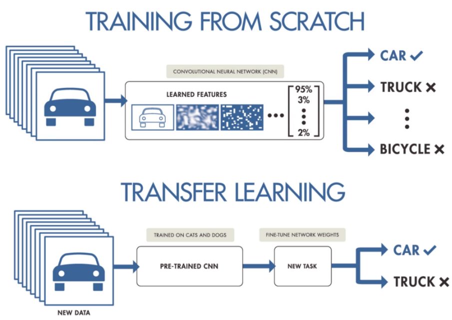

Figure 4.7: Example flow of transfer learning

Source: [77].

In figure 4.7 we can see the example of the flow used by transfer learning. In the first flow a new model is trained and learns the features of the images as well as certain patterns. In the second a pre trained model of a similar subject (both models did object classification) was used and there was only a need to fine tune the weights of the model in order to get it to classify another class.

As the goal for this model was to realize object classification on an image the base model adapted also had to be based around the idea of object classification. The model chosen is called MobileNet. MobileNet is a general architecture and can be used for multiple use cases. Depending on the use case, it can use different input layer size and different width factors. This allows different width models to reduce the number of multiply-adds and thereby reduce inference cost on mobile devices [78].

This model was used as its architecture was designed to be fast, a characteristic desired as in order to calculate perclos the pooling rate of the camera should be as high as computationally possible in order for the sampling of images analysed by the model and used to calculate the percentage of the time the eye is closed. It is also of note that this model was pre trained in image classification using data from ImageNet.

ImageNet is a project aimed at labeling and categorizing images into almost 22,000 categories based on a defined set of words and phrases. At the time of writing, there are over 14 million images in the ImageNet project, making ImageNet a standard for images classification. A yearly contest is run with millions of training images in 1000 categories. The models used in the ImageNet classification competitions are measured against each other for performance. Therefore it provides a "standard" measure for how good a model is for image classification. So many often used transfer learning model models use the ImageNet weights.

As the dataset was structured in folders for each user it was necessary to access the content of each folder and get the file path of each file in order to shuffle the order of the files before training as can be seen on Listing 4.3.

```
files = []
Datadirectory = './input/'

for subject in os.listdir(Datadirectory):
      path = os.path.join(subject, category)

      for img in os.listdir(path):
        files.append(os.path.join(path,img))
```

Listing 4.3: Loading of dataset

Due to RAM memory constraints it is not possible to load all 84898 images into memory at once to train the model and as such it is necessary to train the model in batches reading only a certain number of images from storage for iteration and training the model. The implementation can be seen in Listing 4.4.

```python
batch = 10000
epoch_num = 20
i = 0
while batch * i < len(files) and i < 1:
    print("Started batch - " + str(i))
    training_data_batch = []
    end = batch * (i+1) if batch * (i+1) < len(files) else len(files)-1
    for file in files[batch*i : end]:
            img_array = cv2.imread(file , cv2.IMREAD_GRAYSCALE)
            backtorgb = cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
            new_array = cv2.resize(backtorgb, (img_size,img_size))
            training_data_batch.append([new_array,float( file.split('_')
    [4])])
    X = []
    y = []
    for features, label in training_data_batch:
        X.append(features)
        y.append(label)
    X = np.array(X).reshape(-1, img_size, img_size, 3)
    X = X/255.0
    Y = np.array(y)
    print('started the model')
    history.append(new_model.fit(X, Y, epochs=epoch_num,
    validation_split=0.1))
    i+=1
```

Listing 4.4: Loading of dataset

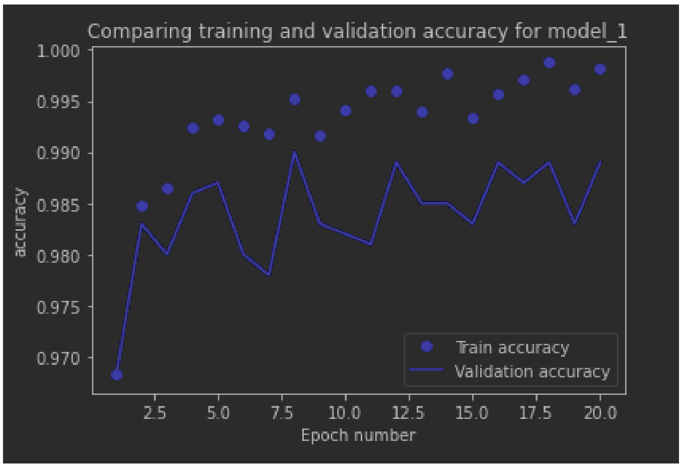

Figure 4.8: Graph with the comparison of accuracy in training and validation data

In Figure 4.8 it is possible to see graphs related to the metrics gathered from training. In this Figure the comparison of accuracy in training and validation data can be seen. As it can be seen the final values for accuracy obtained were 98,8%, higher than the result obtained by [57].

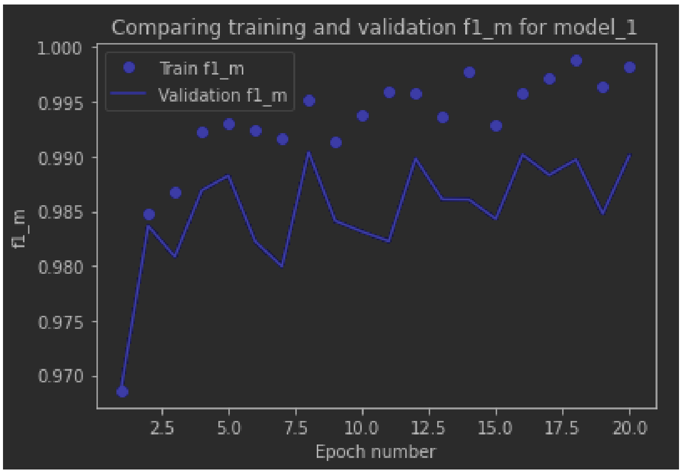

Figure 4.9: Graph with the comparison of f1 in training and validation data

In Figure 4.9 the F1 score of the model is shown. F1 is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive. Precision is also known as positive predictive value, and recall is also known as sensitivity in diagnostic binary classification [79]. As the value obtained was close to one, this shows the model had good performance.

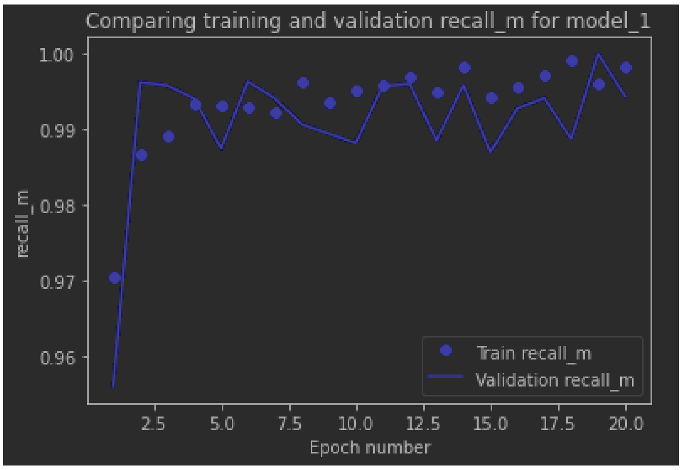

Figure 4.10: Graph with the comparison of recall in training and validation data

In Figures 4.10 it is possible to see the evolution of the recall. Recall (also known as sensitivity) is the fraction of positives that were classified as positive, in this case the percentage of open eyes that were classified as such.

4.2 Heart rate monitoring module

As previously said the heart rate monitoring module solution is based on getting the heart rate of the user by using the heart rate sensor on his smartwatch and making it available to other modules. In order to access the information on the sensor as well as being able to provide it to the other modules when requested. To accomplish this a smartwatch application had to be developed to interface with the operating system of the watch as well as the other modules previously explained in this paper. The sample use case of this module can be seen in Figure 4.11

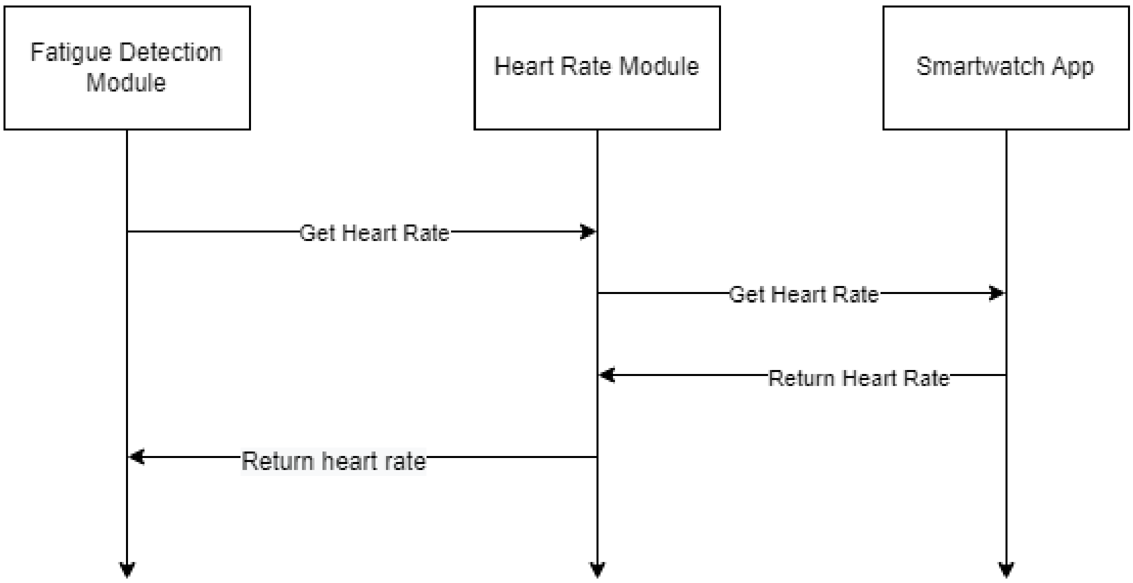

Figure 4.11: Heart rate monitoring system sequence diagram

This application was developed in Flutter. Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. Some of the target platforms are:

- Mobile, both Android and IOS;
- Web App, development of web pages;
- Desktop, development of native Windows, macOS, and Linux apps.

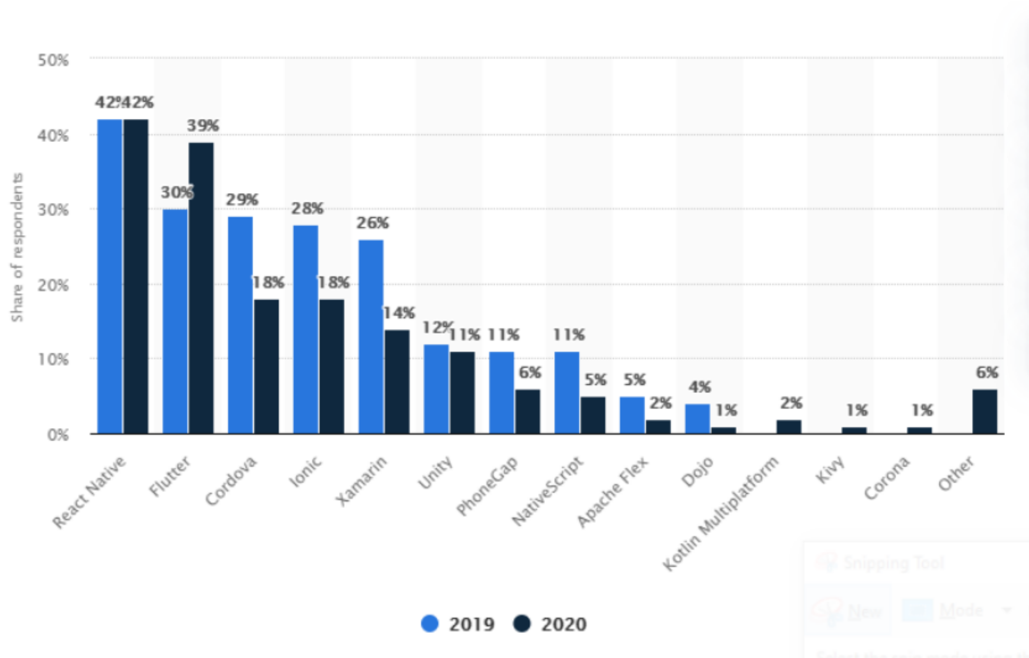

Figure 4.12: Relevance of flutter in the market

Source: [80]

Flutter as several advantages over the competition, firstly being multi platform allows the application to be written in a flutter codebase and executed on several devices in this case the main advantage is being able to, with a single codebase create both an Android and IOS application. Secondly is the increasing popularity and community surrounding Flutter as can be seen on Figure 4.12.

The application was created in order to read the heart rate of the user from the sensor every 10 seconds and store the last 2 minutes of heart rate measurements. The application also allows the user to stop the measurements as well as see the values that are being used. In order to build this application a flutter for WearOs was used as a basis, as targeting WearOs in Flutter requires some configuration. With this done the only steps necessary were to access the sensors and dispatch this information to the other modules. In order to read information from a sensor it is necessary to subscribe to a stream as can be seen in Listing 4.2. This code allows the application to read information from the sensor, annotate it and dispatch it to a central server.

```
Future<void> _startHeartRate() async {
  if (_heartRateSubscription != null) return;
  if (_heartRateAvailable) {
    final stream = await SensorManager().sensorUpdates(
      sensorId: 21, //Id of heart rate sensor
      interval: Sensors.SENSOR_DELAY_NORMAL,
    );
  _accelSubscription = stream.listen((sensorEvent) async {
          Map<String, String> data = {
              'time': DateTime.now().toString(),
              'value': sensorEvent.data[0].toString()
          };
          var body = json.encode(data);

          if (sensorEvent.data != 0.0) {
              Response r = await post(
              Uri.http('APPLICATIONURL', '/api/add_value/'),
              headers: {'Content-Type': 'application/json'},
              body: body,
          );
          print(r.statusCode);
  }
      print(sensorEvent.data);
      setState(() {
        _accelData = sensorEvent.data;
      });
    });
  }
}
```

Listing 4.5: Flutter Code to read information from a sensor and write it in a main server

4.3 Communications module

The goal of this module is to handle all the communications made by the system. This module receives a message from the Fatigue Detection Module that fatigue is above a certain threshold and sends a pop up notification to the user. The representation of this use case can be seen in Figure 4.13.

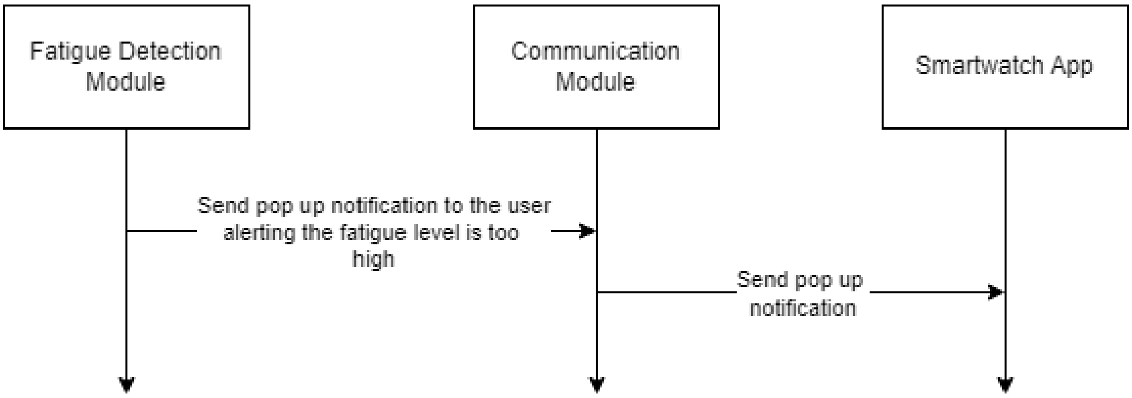

Figure 4.13: Communication module system sequence diagram

In the previously chapter the implementation of pop up notifications was explored, with the conclusion of the implementation of the pop up notifications inside the smartwatch application used to harness the heart rate data. This implementation was done via the Firebase solution popularized by Google. Firebase is a gathering of hosting services for any type of application, one of these services being a notification service. This notification service works just as a Queue, by usage of a publisher subscriber methodology.

The implementation of the Firebase notification service in flutter allow the application to launch pop up notifications when a message is put into a queue in Firebase. For this purpose the official package for integrating flutter and Firebase was used. Simply by following the steps in the quick start guide we can get an application up and running that connects to the Firebase Messaging service and generates pop up notifications in the users device.

This allows the application to show the user any kind of information the application wants by creating a pop up notification with the information. When the system detects a fatigue level over a certain threshold it writes to the queue a message warning the user as can be seen in Figure 4.14.

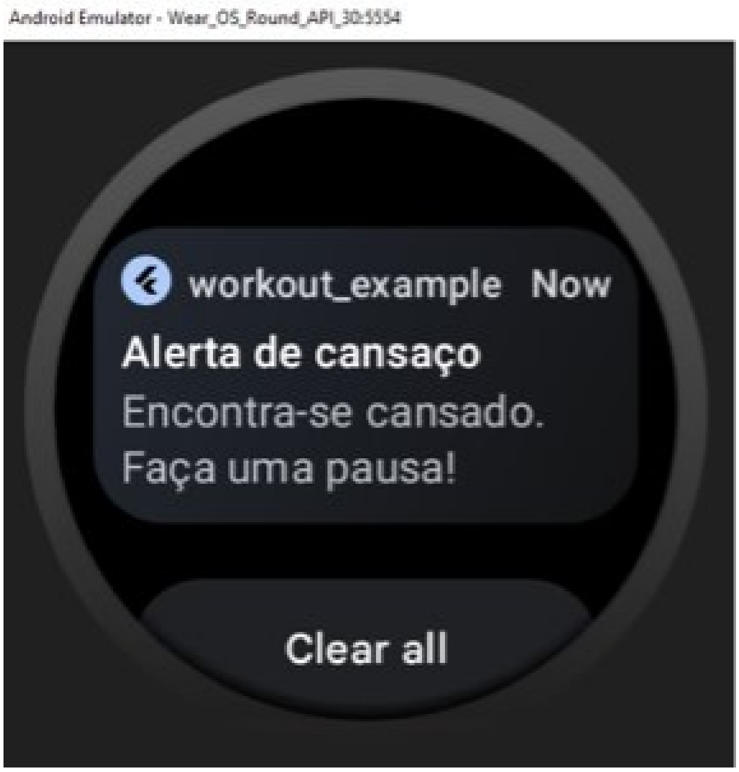

Figure 4.14: Interface of the flutter application running on Android watch

In Listing 4.6 it is possible to see the necessary steps to create a push notification on the Firebase Notification service.

```python
import firebase_admin
from firebase_admin import credentials, messaging

cred = credentials.Certificate("./credentials.json") #File generated by
    Firebase that can be downloaded in the portal
firebase_admin.initialize_app(cred)

topic = "News"
message = messaging.Message(
    notification=messaging.Notification(
    title="Fatigue Level High!",
    body="You should take a break!"
  ),topic="News" )
messaging.send(message)
```

Listing 4.6: Notification example

4.4 Fatigue detection module

In Figure 4.15 the flow of information in this module can be seen. As can be seen this module consumes other modules to get the necessary information to accurately predict the fatigue level. This approach combines two techniques, PERCLOS and heart rate information as well as the time of day and the program that is open.

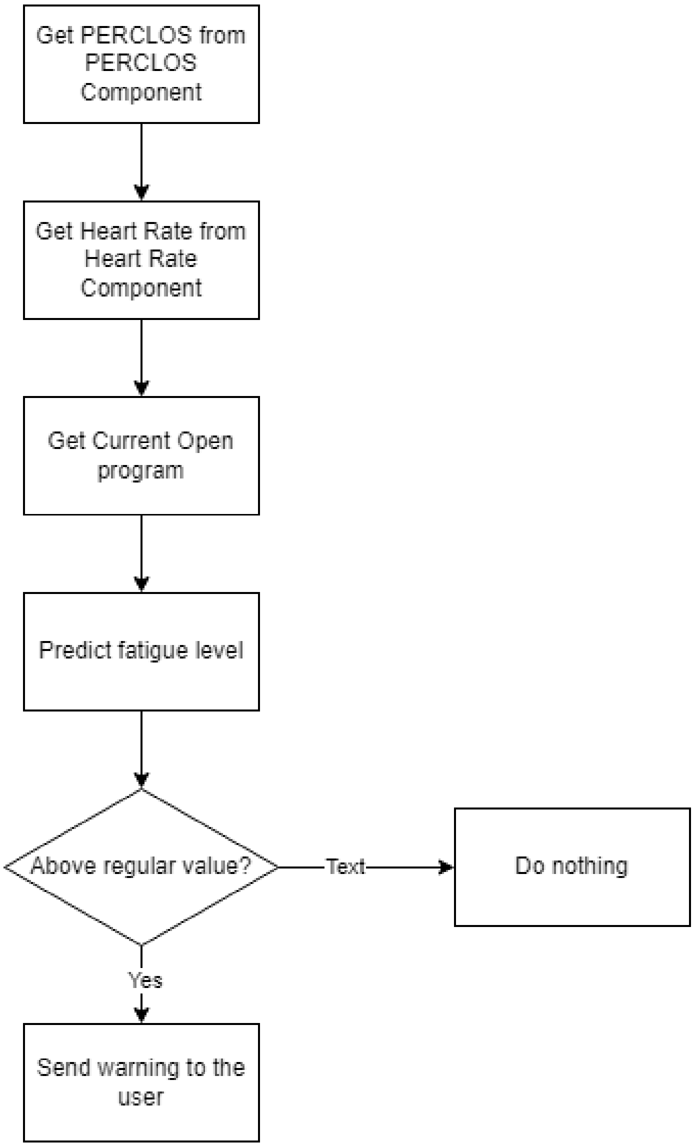

Figure 4.15: flowchart of Fatigue Detection module

The fatigue detection module is the main core of the application as it consumes all other modules in order to give a prediction for the level of perceived fatigue of the user. The main goal of this module is to use the heart rate, PERCLOS, timestamp and application the user has open in his computer in order to provide the final prediction. This module consumes the previous modules thus obtaining the PERCLOS and heart rate from them.

Since the goal of this module was the prediction of a value given a certain time series the chosen method was the usage of an LSTM.

The LSTM network has had a significant influence on language modeling, speech-to-text transcription, machine translation,prediction of time series and other applications. Inspired by the outstanding benchmarks described in the literature, some academic and industry readers seek to study more about the Long Short-Term Memory network (henceforth, "the LSTM network") in order to assess its relevance to their own research or practical use-case [81].

This module makes use of deep learning in order to predict the value from the rest of the parameters. Since the task consisted mainly of using past values to predict future ones the model used was an LSTM. The name of LSTM refers to the analogy that a standard RNN has both "long-term memory" and "short-term memory". The connection weights and biases in the network change once per episode of training, analogous to how physiological changes in synaptic strengths store long-term memories; the activation patterns in the network change once per time-step, analogous to how the moment-to-moment change in electric firing patterns in the brain store short-term memories. The LSTM architecture aims to provide a short-term memory for RNN that can last thousands of time steps, thus "long short-term memory". LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series [82].

This decision was taken as the relevance of LSTM's in predicting values in a time series is undeniable [82] [83]. Since LSTM networks have a memory and use past data(also known as time steps) the usage of this network is appropriate for this problem as the goal is to use past data of fatigue in order to provide as accurate a prediction as possible for the current situation.

4.4.1 Dataset

The dataset utilized was constructed and harnessed for this paper. The resulting dataset was constructed by gathering all the other parameters (timestamp, PERCLOS, open application and heart rate) and prompt the user to rate his fatigue on a level of one to ten. This was done by 8 people for the span of a week. And a similar process is applied when the system is applied to a new user, thus improving the accuracy of the model and growing the dataset over time.

This being said the dataset used to train the model being discussed had 720 lines and would grow each time it was applied to another user.

Before training the dataset was subject to one feature extraction that was to simplify the time stamp into day of the week and time of day as these factors have a higher impact on fatigue according to the data gathered. For example Mondays always showed an increase in the median fatigue level, especially in the morning.

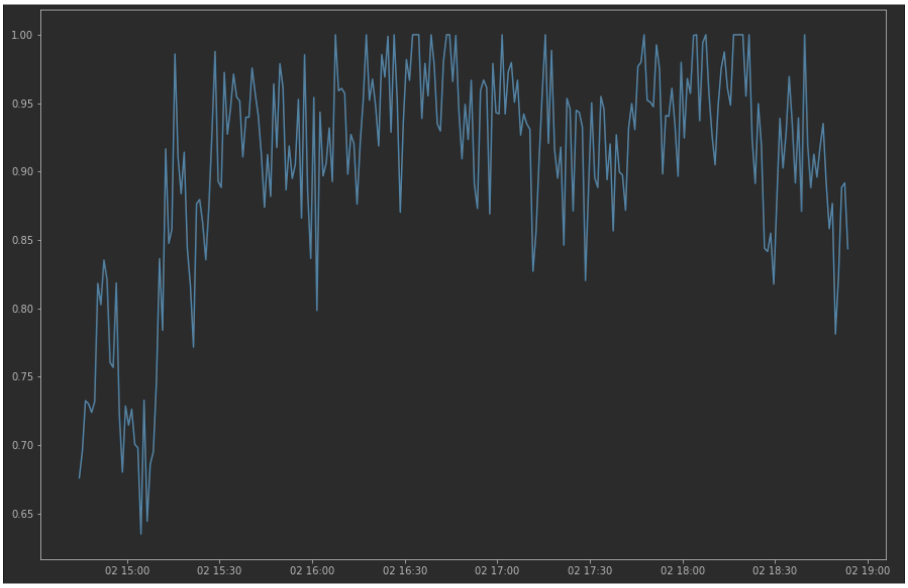

Figure 4.16: Graph of fatigue evolution throughout a day

Figure 4.16 shows the evolution of the fatigue of an user throughout a day. As can be seen the fatigue level varies a lot making the effort of predicting the level of fatigue harder. It can also be seen that there is a very strong correlation between the perceived level of fatigue and the value of PERCLOS. The correlation plot can be seen in Figure 4.17.

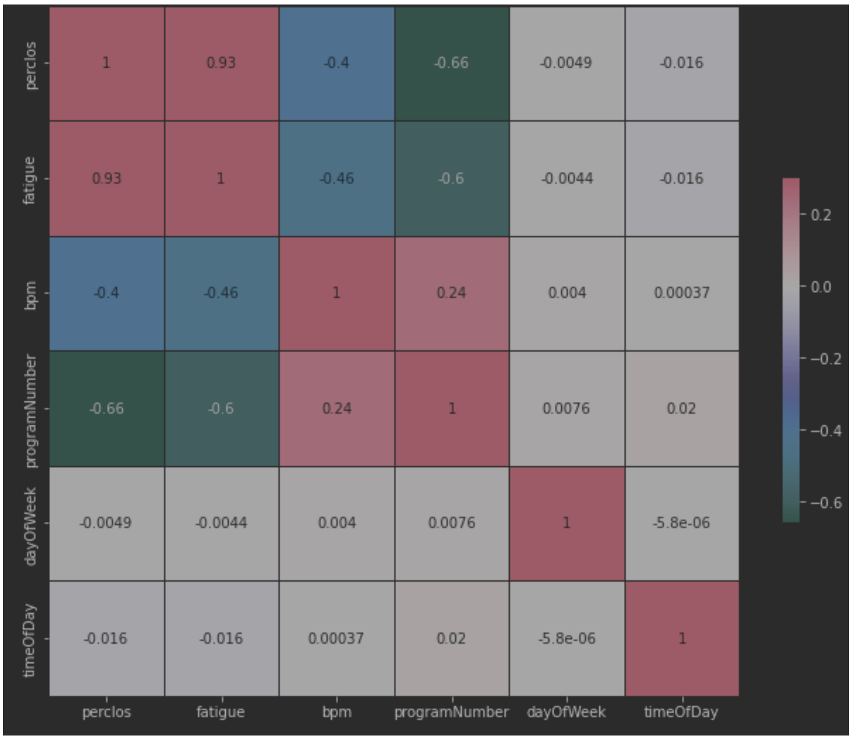

Figure 4.17: Graph of correlation of the variables of the dataset

The correlation can be further seen in Figure 4.18 with a graph of perclos and fatigue.

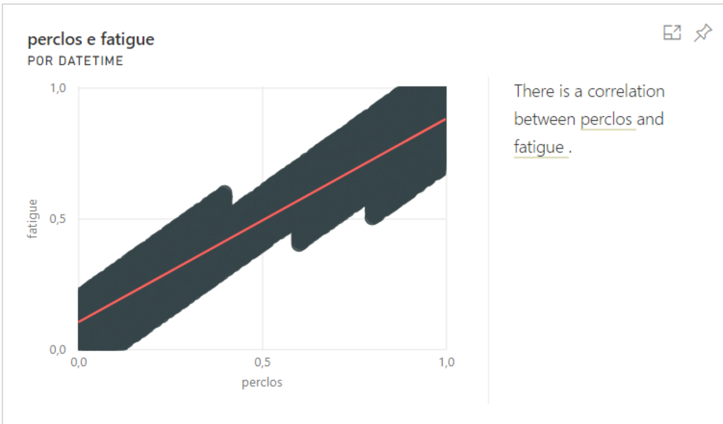

Figure 4.18: Graph of correlation of PERCLOS and fatigue

In Table 4.2 we can see the values and for the metrics associated with fatigue such as mean, quarters and standard deviation.

Metric	Fatigue
mean	0.521648
standard deviation	0.243788
1st quarter	0.339200
3rd quarter	0.703253

Table 4.2: Table with exploratory analysis of the fatigue level

With a further exploration of the data it is also possible to find more interesting correlations such as some programs having a higher impact in PERCLOS and therefore fatigue as can be seen in Figure 4.19.

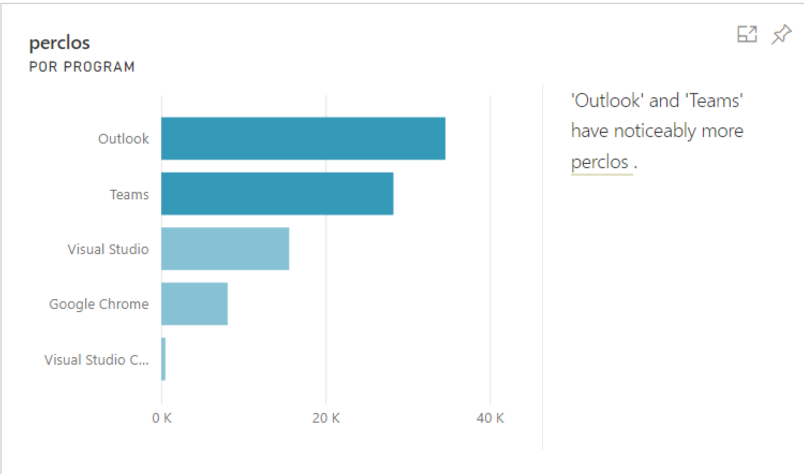

Figure 4.19: Correlation between program and PERCLOS

4.4.2 Model

The model used for this task was an LSTM given the fact that LSTM's excels in the prediction of future values given a historic of the values. The values were encoded and the model trained. Table 4.3 contains a summary of the configurations tested.

Table 4.3: Configuration tests

Model	RMSE
LSTM-50 units Dropout- 0.2 LSTM-50 units Dropout- 0.2 LSTM-50 units Dropout- 0.3	0.1177
LSTM- 100 units Dropout- 0.1 LSTM - 100 units Dropout-0.1 LSTM - 100 units Dropout-0.1	0.1070
LSTM - 100 units Dropout- 0.01 LSTM - 100 units Dropout- 0.01 LSTM- 100 units Dropout- 0.01	0.1065
LSTM- 200 units Dropout- 0.01 LSTM- 200 units Dropout- 0.01 LSTM- 200 units Dropout- 0.01	0.1011
LSTM- 200 units Dropout- 0.1 LSTM- 200 units Dropout- 0.1 LSTM- 200 units Dropout- 0.1	0.1052

In Figure 4.20 it is possible to see the winning configuration used.

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_25 (LSTM)              (None, 60, 200)           164800

 dropout_25 (Dropout)        (None, 60, 200)           0

 lstm_26 (LSTM)              (None, 60, 200)           320800

 dropout_26 (Dropout)        (None, 60, 200)           0

 lstm_27 (LSTM)              (None, 60, 200)           320800

 dropout_27 (Dropout)        (None, 60, 200)           0

 lstm_28 (LSTM)              (None, 200)               320800

 dropout_28 (Dropout)        (None, 200)               0

 dense_6 (Dense)             (None, 1)                 201

=================================================================
Total params: 1,127,401
Trainable params: 1,127,401
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.20: Summary of the model used

Listing 4.7 contains the necessary transformations to the dataset. It was a necessary to encode the program names as well as create new columns from the existing date time. The created columns were the day of the week and the time of day.

```
df['datetime'] = pd.to_datetime(df['datetime'])
possibleApplications = ["Outlook", "Teams", "Visual Studio Code", "Visual Studio", "Google Chrome"]
df['programNumber'] = df['program'].map({'Outlook':0, 'Teams':1 , "Visual Studio Code":2 , "Visual Studio":3 , "Google Chrome":4})

def calc_new_col(row):
    return row["datetime"].weekday()

def calc_new_col2(row):
    now  = row["datetime"]
    return (now - now.replace(hour=0, minute=0, second=0, microsecond=0)).total_seconds()

df["dayOfWeek"] = df.apply(calc_new_col, axis=1)
df["timeOfDay"] = df.apply(calc_new_col2, axis=1)

df = df.set_index('datetime')
df
```

Listing 4.7: Feature extraction from dataset

With the information encoded and the new columns created it was necessary to normalize the data. A Min-Max scaler was used to encode the information between 0 and 1. Listing 4.8 contains the necessary code to perform this task.

```
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
```

Listing 4.8: Encoding features from dataset

The results obtained for the best configuration of an layers. The results obtained were as follows in Figure 4.21, where the Root Mean Squared Error can be observed.

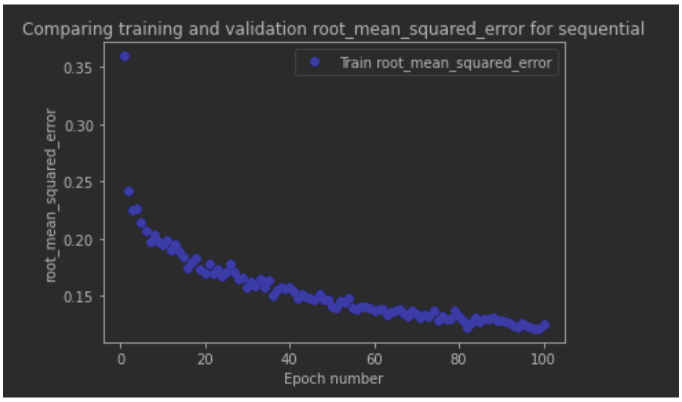

Figure 4.21: Graph with the root mean square error of the model

In Figure 4.22 a graph with the real values of fatigue and the values predicted by the model can be seen. This graph serves to show the proximity of the values in relation to the real ones.

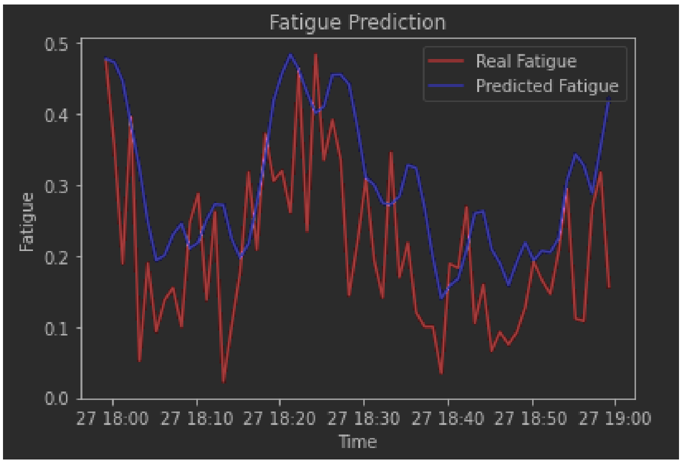

Figure 4.22: Graph with the comparison between real and predicted values

4.5 Research question

The research question for this thesis was "Will it be possible to create a hybrid fatigue detection system that combines more than one fatigue detection technique and warn an user when fatigue levels increase?"

As was seen throughout this Chapter the creation of an hybrid system capable of using several fatigue detection techniques is possible. In this Chapter the details of a system capable of combining the calculation of the percentage of eye lid closure over time and the Heart rate information of the user. This program is also capable of sending a notification to the user's smartwatch when fatigue levels are detected to be to high. With these two factors a fatigue detection system combining two different techniques was created. The ability to warn the user was added by the pop up notification on the user's watch.

4.6 Conclusions

This chapter explored the system developed for this thesis. In this chapter all the steps necessary to create a fatigue detection system was explored. This fatigue detection system combines two existing techniques in order to predict fatigue, PERCLOS and Heart Rate. This approach thus required the capacity to calculate PERCLOS and together with the heart rate and the other used factors and a model capable of combining all these factors in order to predict the level of fatigue.

In this Chapter the inner works of the modules were explored. The PERCLOS Calculating Module required the development of an eye detection component and an image transformation component. They were necessary in order to aid the main component, the PERCLOS calculating module by taking an image and returning an image the AI model developed for this module can use in order to classify the eye state. It is of note the results obtained by the AI model. The results obtained quoted around 99.4% accuracy. In the Heart rate module it was necessary to create a smartwatch application in order to harness the data from the sensor inside the watch. The Communication Module has the responsibility of being triggered when fatigue levels are high and sending a notification to the users smartwatch. In order to send the notification the existing application developed for the Heart Rate Module was modified in order to integrate with Google's Firebase Notification center in order to show notifications created on Firebase on the device. The Fatigue detection Module- This module is responsible for using the other modules in order to obtain a prediction for the value of fatigue, an AI model had to be created and the results analysed. These results do a good job of predicting reality as can be seen on Figure 4.22.

Chapter 5

Conclusions and Future work

In this chapter the work done previously will be analysed and the conclusions will be presented. This chapter will also discuss future work.

The fatigue detection system developed in this thesis utilizes both the calculation of the percentage of the time that the eye is open and the utilization of the heart rate of the user, two techniques discussed on Chapter 2. By combining these two techniques the system can become more robust and analyse more data in order to improve the accuracy of the prediction given. As analysed in Chapter 2 no other system was found that was able to combine both PERCLOS data and Heart rate data in order to improve the quality of the prediction given.

5.1 Achieved work

The main goal of this section is to analyse the results obtained in the development of the system as well as analyse some of the main constraints of the system.

In Chapter 2 the state of the art for the technologies that current fatigue detection systems use were studied. These technologies consisted of using the eye state of the user, calculating eye curvature via facial landmarks, using EEC technology and using heart rate information in order do provide an estimate for the level of fatigue in an user. This information was used in order to assist in the choice of which method to use in our system. In this Chapter existing fatigue detection systems were also researched. Currently a lot of fatigue detection systems are being developed by car manufacturers and used with the goal of reducing the number of accidents but there are also other commercially available solutions being implemented with focuses outside of the automotive industry. In the review of literature no system was found that combined two techniques in order to predict fatigue, and this is where this system innovates. By combining the calculation of the percentage of the time the eye is closed and the heart rate of the user this system can provide more accurate predictions towards the level of fatigue as well as being more reliable as the system is not reliable on a sole sensor.

In Chapter 3 an outline of the proposed solution was presented with the main responsibilities of each model was explored. This solution is made up of of four modules, one to handle communication one to handle the heart rate information , one for eye state classification and one to output the final prediction of the fatigue. The two latter modules required the development of Artificial Intelligence modules in order to fulfill the requirement that where set.

This architecture was developed in order to accommodate the functional requirements necessary to create a fatigue detection system that combines PERCLOS and heart rate to predict the level of fatigue. The modules were design in order to separate responsibilities and the main module, Fatigue Detection Module, utilizes the other modules to gather the information required in order to work; it utilizes the PERCLOS calculating system to get the value of PERCLOS and the heart rate module in order to get the heart rate of the user. When fatigue is above a certain threshold this module calls the Communication module to warn the user.

In Chapter 4 the modules from Chapter 3 were further developed. In the heart rate module the smartwatch application developed was explained as well as both Artificial Intelligence models that were developed for this system. This Chapter contained all work that was done in order to create all the models and supporting infrastructure of the system. In this Chapter the modules that were talked about in the previous paragraph. The inner works of the the following modules were explored:

- PERCLOS Calculating Module- In this module the developments of an eye detection component and an image transformation component were necessary in order to aid the main component, the PERCLOS calculating module by taking an image and returning an image the AI model developed for this module can use in order to classify the eye state. It is of note the results obtained by the AI model. The results obtained quoted around 99.4% accuracy which is a value higher than that obtained in the literature as can be seen in Table 2.7 even under ideal circumstances. This value was also obtained training with images with a variety of subjects, angles, reflections and people wearing glasses; this makes this model more robust than one that does not take into account these factors. There also was proven to be a strong correlation between the calculated for PERCLOS and the final value of fatigue in the user;
- Heart rate module- In this module it was necessary to create a smartwatch application in order to harness the data from the sensor inside the watch. In order to do this several steps had to be taken to configure the app to be able to access the sensor as well as to send them to an API so that the other modules can access the information on command;
- Communication Module- This module has the responsibility of being triggered when fatigue levels are high and sending a notification to the users smartwatch. In order to send the notification the existing application developed for the Heart Rate Module was modified in order to integrate with Google's Firebase Notification center in order to show notifications created on Firebase on the device;
- Fatigue detection Module- This module is responsible for using the other modules in order to obtain a prediction for the value of fatigue, an AI model had to be created and the results analysed. These results do a good job of predicting reality as can be seen on Figure 4.22. Since there was not a real dataset with the information required to train this model one had to be constructed just for this purpose. This meant the amount of data used in training this model is not ideal, but this problem would fade as more and more people used the system and went through the calibration period. It is of note that expanding the subjects in these tests is still necessary and something that needs to be included in the future work.

5.2 Research question

The research question for this thesis was "Will it be possible to create a hybrid fatigue detection system that combines more than one fatigue detection technique and warn an user when fatigue levels increase?"

As demonstrated throughout this thesis, it is feasible to design a hybrid system capable of employing more than one fatigue detection algorithm. The features of a system capable of integrating the computation of the percentage of eye lid closure over time and the user's heart rate have been covered in this Chapter. When weariness levels are identified to be too high, this software can send a notification to the user's wristwatch. A tiredness detection system integrating two separate methodologies was constructed using these two parameters. The pop-up notice on the user's watch added the option to warn the user.

5.3 Contributions

The fatigue detection system developed in this thesis utilizes both the calculation of the percentage of the time that the eye is open and the utilization of the heart rate of the user, two techniques discussed on Chapter 2. By combining these two techniques the system can become more robust and analyse more data in order to improve the accuracy of the prediction given. As analysed in Chapter 2 no other system was found that was able to combine both PERCLOS data and Heart rate data in order to improve the quality of the prediction given.

The system developed in this thesis has the ability to improve lives of workers. By monitoring the fatigue levels of users this system can warn them when their fatigue levels increase to much. With this information the user can take appropriate breaks and improve their productivity as well as possibly prevent burnouts. This can have a positive effect on the lives of users as a better handle on fatigue improves the mental health and quality of life.

Fatigue Detection systems are becoming more and more popular because of their adoption in the automotive industry and even though the main focus of this system was to help workers prevent burnouts and create better habits the core ideas behind the system(combining PERCLOS and Heart Rate Data) can easily be adopted into the automotive industry. The main concept also presents an idea of the way forward for Fatigue detection systems and a way other systems can become more complex and more adaptable.

5.4 Future Work

This thesis delved into a topic that is the subject of multiple studies because fatigue detection systems are being developed currently to aid drivers. Several companies have already incorporated some kind of fatigue detection system into their car and others are certain to follow and improve upon the existent technology. The work done in this thesis can be adapted to be included in such systems and this can be a future approach to the development of this work.

In order for the system to be more accurate in the prediction a bigger user base would be necessary as the dataset utilised to train the Fatigue Prediction Module was small. The system would benefit from an increase in the number of users to grow the data available in the system, thus improving further the accuracy the system can obtain. By widening the scope of the system and thus increase the number of users using it it would be possible to create a better system.

By adopting a larger reach the existing program can use the new users as a source of data, requesting users fill in information about their level of fatigue. By doing this a bigger dataset can be created and utilized to train the Fatigue Detection module with more data from more people, the data can be a better representation of workers. In the current iteration all the participants belong to the same company and do a similar job, which could lead to a lower accuracy when applied to workers in other professions.

If the interest came to incorporate this system into a car, there would be a need to substitute components such as the heart rate module as there would be the possibility for the manufacturer to incorporate this sensor into the steering wheel and thus render useless the necessity of the smartwatch. The Communications module could also integrate into the car's operating system as it would render it more similar to approaches that exist in the current market.

Bibliography

[1] Yijing Xiao et al. "Impacts ofWorking from Home during COVID-19 Pandemic on Physical and Mental Well-Being of Office Workstation Users". In: *Journal of Occupational and Environmental Medicine* 63 (3 Mar. 2021), pp. 181–190. issn: 15365948. doi: 10.1097/JOM.0000000000002097.

[2] Owen Hughes. *How to get a tech job with no experience*. Mar. 2022. url: https://www.zdnet.com/education/professional-development/how-to-get-a-job-with-no-experience/.

[3] Frank Mueller-Langer and Estrella Gómez-Herrera. "Mobility restrictions and the substitution between on-site and remote work: Empirical evidence from a European online labour market". In: *Information Economics and Policy* 58 (2022), p. 100951. issn: 0167-6245. doi: https://doi.org/10.1016/j.infoecopol.2021.100951. url: https://www.sciencedirect.com/science/article/pii/S0167624521000391.

[4] Dorota Żołnierczyk-Zreda. "An intervention to reduce work-related burnout in teachers". In: *International Journal of Occupational Safety and Ergonomics* 11.4 (2005), pp. 423–430. doi: 10.1080/10803548.2005.11076661.

[5] Sherrill W. Hayes et al. *STRESS AND WORK-RELATED BURNOUT DURING COVID-19*. Nov. 2020.

[6] TowardsDataScience. *Why Is Everyone Talking About Artificial Intelligence?*

[7] Riccardo Miotto et al. "Deep learning for healthcare: review, opportunities and challenges". In: *Briefings in Bioinformatics* 19 (6 Nov. 2018), pp. 1236–1246. issn: 1467-5463. doi: 10.1093/bib/bbx044.

[8] Yann LeCun and Alfredo Canziani. *What Is Deep Learning? 3 things you need to know*.

[9] Prisma. *Prisma*. url: https://www.prisma-statement.org//.

[10] Hameed Haq, REHAN KHAN, and Rahila Yasmeen. "Healthcare needs of the Muslim patient community in the undergraduate medical curriculum – Are we there?" In: *Pakistan Journal of Medical Sciences* 35 (May 2019). doi: 10.12669/pjms.35.3.861.

[11] Prisma. *Prisma Flow Diagram*. url: https://prisma-statement.org/prismastatement/flowdiagram.aspx.

[12] D.T Pham and P.T.N Pham. "Artificial intelligence in engineering". In: *International Journal of Machine Tools and Manufacture* 39 (6 June 1999), pp. 937–949. issn: 08906955. doi: 10.1016/S0890-6955(98)00076-5.

[13] Jesus Serrano-Guerrero, Francisco P. Romero, and Jose A. Olivas. "Fuzzy logic applied to opinion mining: A review". In: *Knowledge-Based Systems* 222 (June 2021), p. 107018. issn: 09507051. doi: 10.1016/j.knosys.2021.107018.

[14] L.A. Zadeh. "Fuzzy logic". In: *Computer* 21.4 (1988), pp. 83–93. doi: 10.1109/2.53.

[15] Rabia Tehseen, Muhammad Shoaib Farooq, and Adnan Abid. "Earthquake Prediction Using Expert Systems: A Systematic Mapping Study". In: *Sustainability* 12 (6 Mar. 2020), p. 2420. issn: 2071-1050. doi: 10.3390/su12062420.

[16] Bruce G Buchanan and Reid Q Smith. *" " FUNDAMENTALS OF EXPERT SYSTEMS*.

[17] J. Durkin. "Expert Systems: A View of the Field". In: *IEEE Intelligent Systems* 11.02 (Mar. 1996), pp. 56–63. issn: 1941-1294. doi: 10.1109/64.491282.

[18] Oliver Kramer. *Genetic Algorithms*. 2017. doi: 10.1007/978-3-319-52156-5_2.

[19] Seyedali Mirjalili. *Genetic Algorithm*. 2019. doi: 10.1007/978-3-319-93025-1_4.

[20] TowardsDataScience. *Coding Deep Learning For Beginners*. url: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d.

[21] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.

[22] Batta Mahesh. "Machine Learning Algorithms-A Review". In: *International Journal of Science and Research* (2018). issn: 2319-7064. doi: 10.21275/ART20203995. url: https://www.researchgate.net/publication/344717762.

[23] Angela Shi. *Overview of Supervised Machine Learning Algorithms*. url: https://towardsdatascience.com/overview-of-supervised-machine-learning-algorithms-a5107d036296.

[24] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. *Supervised Learning*. Springer Berlin Heidelberg, pp. 21–49. doi: 10.1007/978-3-540-75171-7_2.

[25] Michael W Berry Azlinah, Mohamed Bee, and Wah Yap. *Supervised and Unsupervised Learning for Data Science Unsupervised and Semi-Supervised Learning Series Editor: M. Emre Celebi*. url: http://www.springer.com/series/15892.

[26] Stephen Marsland. *Machine Learning - An Algorithmic Perspective.* Chapman and Hall / CRC machine learning and pattern recognition series. CRC Press, 2009, pp. I–XVI, 1–390. isbn: 978-1-4200-6718-7.

[27] Kaifeng Gao et al. "Julia Language in Machine Learning: Algorithms, Applications, and Open Issues". In: *Computer Science Review* (May 2020), pp. 100254–100259. doi: 10.1016/j.cosrev.2020.100254.

[28] Robert Moni. *Reinforcement Learning algorithms — an intuitive overview*. url: https://smartlabai.medium.com/reinforcement-learning-algorithms-an-intuitive-overview-904e2dff5bbc.

[29] Chris M Bishop. "Neural networks and their applications". In: *Review of scientific instruments* 65.6 (1994), pp. 1803–1832.

[30] Anders Krogh. "What are artificial neural networks?" In: *Nature Biotechnology* 26 (2 Feb. 2008), pp. 195–197. issn: 1087-0156. doi: 10.1038/nbt1386.

[31] Amir Ali. *Artificial Neural Network (ANN) with Practical Implementation*. url: https://medium.com/machine-learning-researcher/artificial-neural-network-ann-4481fa33d85a.

[32] A.K. Jain, Jianchang Mao, and K.M. Mohiuddin. "Artificial neural networks: a tutorial". In: *Computer* 29.3 (1996), pp. 31–44. doi: 10.1109/2.485891.

[33] Laith Alzubaidi et al. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8 (1 Dec. 2021), p. 53. issn: 2196-1115. doi: 10.1186/s40537-021-00444-8.

[34] Jiakai Wei. "Forget the Learning Rate, Decay Loss". In: ().

[35] Igiri Chinwe Peace, Anyama Oscar Uzoma, and Silas Abasiama Ita. *Effect of Learning Rate on Artificial Neural Network in Machine Learning*. url: www.ijert.org.

[36] Philip J. Drew and John R.T. Monson. "Artificial neural networks". In: *Surgery* 127 (1 Jan. 2000), pp. 3–11. issn: 00396060. doi: 10.1067/msy.2000.102173.

[37] Sun-Chong Wang. *Artificial Neural Network*. Springer US, 2003, pp. 81–100. doi: 10.1007/978-1-4615-0377-4_5.

[38] John Paul Mueller and Luca Massaron. *Machine Learning For Dummies*. 1st. For Dummies, 2016. isbn: 1119245516.

[39] Pramila P. Shinde and Seema Shah. *A Review of Machine Learning and Deep Learning Applications*. 2018, pp. 1–6. doi: 10.1109/ICCUBEA.2018.8697857.

[40] Iqbal H. Sarker. “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions”. In: *SN Computer Science* 2 (6 Nov. 2021), p. 420. issn: 2662-995X. doi: 10.1007/s42979-021-00815-1.

[41] Samira Pouyanfar et al. “A Survey on Deep Learning”. In: *ACM Computing Surveys* 51 (5 Sept. 2019), pp. 1–36. issn: 0360-0300. doi: 10.1145/3234150.

[42] C. Goller and A. Kuchler. *Learning task-dependent distributed representations by backpropagation through structure*. 1996, 347–352 vol.1. doi: 10.1109/ICNN.1996.548916.

[43] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: (June 2014). url: http://arxiv.org/abs/1406.1078.

[44] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: Proceedings of Machine Learning Research 9 (13–15 May 2010). Ed. by Yee Whye Teh and Mike Titterington, pp. 249–256. url: https://proceedings.mlr.press/v9/glorot10a.html.

[45] Samira Pouyanfar et al. “A Survey on Deep Learning: Algorithms, Techniques, and Applications”. In: *ACM Comput. Surv.* 51.5 (Sept. 2018). issn: 0360-0300. doi: 10.1145/3234150. url: https://doi.org/10.1145/3234150.

[46] Zhiwei Zhao and Youzheng Wu. “Attention-based convolutional neural networks for sentence classification”. In: vol. 08-12-September-2016. International Speech and Communication Association, 2016, pp. 705–709. doi: 10.21437/Interspeech.2016-354.

[47] George E. Dahl et al. “Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1 (2012), pp. 30–42. doi: 10.1109/TASL.2011.2134090.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. issn: 0001-0782. doi: 10.1145/3065386. url: https://doi.org/10.1145/3065386.

[49] Rohit Hooda, Vedant Joshi, and Manan Shah. *A comprehensive review of approaches to detect fatigue using machine learning techniques*. 2021. doi: 10.1016/j.cdtm.2021.07.002.

[50] Ivan D. Brown. “Driver Fatigue”. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 36 (2 June 1994), pp. 298–314. issn: 0018-7208. doi: 10.1177/001872089403600210.

[51] Gulbadan Sikander and Shahzad Anwar. “Driver Fatigue Detection Systems: A Review”. In: *IEEE Transactions on Intelligent Transportation Systems* 20 (6 June 2019), pp. 2339–2352. issn: 15249050. doi: 10.1109/TITS.2018.2868499.

[52] BMW. *STAY SAFE THIS WINTER WITH THE LATEST FEATURES FROM BMW*. 2019. url: https://news.bmwofsilverspring.com/article/stay-safe-this-winter-with-the-latest-features-from-bmw.

[53] Sheng-Yang Shi, Wen-Zhong Tang, and Yan-Yang Wang. “A Review on Fatigue Driving Detection”. In: *ITM Web of Conferences* 12 (Sept. 2017), p. 01019. issn: 2271-2097. doi: 10.1051/itmconf/20171201019.

[54] X. Fan, B.C. Yin, and Y.F. Sun. *Machine Learning and Cybernetics, 2007 International Conference on.* IEEE Xplore, 2007. isbn: 142440973X.

[55] Tiesheng Wang et al. *Workshop VLSI Design & Video Tech.*

[56] Mateusz Knapik and Bogusław Cyganek. "Driver's fatigue recognition based on yawn detection in thermal images". In: *Neurocomputing* 338 (Apr. 2019), pp. 274–292. issn: 18728286. doi: 10.1016/j.neucom.2019.02.014.

[57] H. Wang, L.B. Zhou, and Y. Ying. *Robotics, Automation, and Mechatronics (RAM), 2010 IEEE Conference on : date, 28-30 June 2010.* IEEE, 2010. isbn: 9781424465064.

[58] Fang Zhang et al. "Driver fatigue detection based on eye state recognition". In: Institute of Electrical and Electronics Engineers Inc., Mar. 2017, pp. 105–110. isbn: 9781509049936. doi: 10.1109/CMVIT.2017.25.

[59] Institute of Electrical et al. *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON) : 28th-30th November, FARS Hotel and Resorts, Dhaka, Bangladesh.* isbn: 9781728142944.

[60] Nafis Irtija et al. *Fatigue Detection Using Facial Landmarks*. 2018.

[61] E. Rogado et al. "Driver fatigue detection system". In: IEEE, Feb. 2009, pp. 1105–1110. isbn: 978-1-4244-2678-2. doi: 10.1109/ROBIO.2009.4913155.

[62] MohammadTayarani Darbandy et al. "A new approach to detect the physical fatigue utilizing heart rate signals". In: *Research in Cardiovascular Medicine* 9 (1 2020), p. 23. issn: 2251-9572. doi: 10.4103/rcm.rcm_8_20.

[63] Anuva Chowdhury et al. "Sensor applications and physiological features in drivers' drowsiness detection: A review". In: *IEEE sensors Journal* 18.8 (2018), pp. 3055–3067.

[64] Nick Owen, Horace King, and Matthew Lamb. "Literature Review of Race Driver Fatigue Measurement in Endurance Motorsport". In: *Procedia Engineering* 112 (2015). 'The Impact of Technology on Sport VI' 7th Asia-Pacific Congress on Sports Technology, APCST2015, pp. 344–348. issn: 1877-7058. doi: https://doi.org/10.1016/j.proeng.2015.07.260. url: https://www.sciencedirect.com/science/article/pii/S1877705815015179.

[65] A Nabo. "Driver attention–Dealing with drowsiness and distraction,"Saab Automobile AB, Trollhättan". In: (2009).

[66] John A. Caldwell, Kecia K. Hall, and Bradley S. Erickson. "EEG Data Collected From Helicopter Pilots in Flight Are Sufficiently Sensitive to Detect Increased Fatigue From Sleep Deprivation". In: *The International Journal of Aviation Psychology* 12.1 (2002), pp. 19–32. doi: 10.1207/S15327108IJAP1201_3. eprint: https://doi.org/10.1207/S15327108IJAP1201_3. url: https://doi.org/10.1207/S15327108IJAP1201_3.

[67] Drew Dawson, Amelia K Searle, and Jessica L Paterson. "Look before you (s) leep: evaluating the use of fatigue detection technologies within a fatigue risk management system for the road transport industry". In: *Sleep medicine reviews* 18.2 (2014), pp. 141–152.

[68] T Von Jan et al. "Don't sleep and drive–VW's fatigue detection technology". In: *Proceedings of 19th International Conference on Enhanced Safety of Vehicles, Washington, DC*. Vol. 168. Citeseer. 2005.

[69] SmartEye. *SmartEye*. url: https://smarteye.se/.

[70] MA Corbett. "Science & Technology Watch: A Drowsiness Detection System for Pilots: Optalert®". In: *Aviation, space, and environmental medicine* 80.2 (2009), pp. 149–149.

[71] Fatemeh Sarhaddi et al. "A comprehensive accuracy assessment of Samsung smartwatch heart rate and heart rate variability". In: (). doi: 10.1101/2022.04.29.22274461. url: https://doi.org/10.1101/2022.04.29.22274461.

[72] OpenCV. *OpenCV*. url: https://opencv.org/about/.
[73] Ivan Culjak et al. "A brief introduction to OpenCV". In: *2012 Proceedings of the 35th International Convention MIPRO*. 2012, pp. 1725–1730.
[74] S. Brahmbhatt. *Practical OpenCV*. Practical OpenCV. Apress, 2013. isbn: 9781430260806. url: https://books.google.pt/books?id=%5C_5sQAwAAQBAJ.
[75] MRL. *MRL Eye Dataset*. Jan. 2022.
[76] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big Data* 3 (1 Dec. 2016), p. 9. issn: 2196-1115. doi: 10.1186/s40537-016-0043-6.
[77] Robotic Automation Expert (RAX). *Introducing Transfer Learning as Your Next Engine to Drive Future Innovations*. url: https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567.
[78] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (Apr. 2017).
[79] Abdel Aziz Taha and Allan Hanbury. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool". In: *BMC Medical Imaging* 15 (1 Dec. 2015), p. 29. issn: 1471-2342. doi: 10.1186/s12880-015-0068-x.
[80] statista. *Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021*. url: https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/.
[81] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. issn: 0167-2789. doi: https://doi.org/10.1016/j.physd.2019.132306. url: https://www.sciencedirect.com/science/article/pii/S0167278919305974.
[82] Fazle Karim et al. "Multivariate LSTM-FCNs for time series classification". In: *Neural Networks* 116 (2019), pp. 237–245. issn: 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.04.014. url: https://www.sciencedirect.com/science/article/pii/S0893608019301200.
[83] Jian Cao, Zhi Li, and Jian Li. "Financial time series forecasting model based on CEEMDAN and LSTM". In: *Physica A: Statistical Mechanics and its Applications* 519 (2019), pp. 127–139. issn: 0378-4371. doi: https://doi.org/10.1016/j.physa.2018.11.061. url: https://www.sciencedirect.com/science/article/pii/S0378437118314985.