TED UNIVERSITY

CMPE 491

Senior Design Project I

SAFE AWAKE

-DROWSINESS DETECTION & ALERTING SYSTEM-

High-Level Design Report

07.01.2024

SUPERVISOR:
Yücel ÇİMTAY

JURY MEMBERS:
Gökçe Nur YILMAZ
Venera ADANOVA

Team Members:

Başak DEĞERLİ	basak.degerli@tedu.edu.tr
Özge AYDIN	ozge.aydin18@tedu.edu.tr
Büşra TOMBAK	busra.tombak@tedu.edu.tr
Kemal GEVGİLİ	kemal.gevgili@tedu.edu.tr

PROJECT URL: https://safeawake.github.io/

Table of Contents

1. Introduction

The SafeAwake system is a driver drowsiness and yawn detection and alerting system developed to combat the issue of drowsy driving. It is designed to continuously monitor drivers for signs of drowsiness, such as yawning, using a combination of advanced sensor technology and complex algorithms. The system is intended for deployment across a wide variety of vehicles, including trucks, buses, taxis, and personal automobiles, with the overall goal of raising public safety standards. In fact, we aim to design a system that not only raises public safety standards but is also cost-effective and flexible enough to operate in a variety of environments.

In subsequent sections of this high-level design report, we will examine our system's architecture, explain design goals, and go into the details of each crucial component. Through this document, we do not only outline the architecture of our system but also demonstrate our dedication to developing innovative solutions to provide a safer and more secure driving environment.

1.1 Purpose of the system

The SafeAwake system serves several purposes. Its main goal is to address the significant problem of drowsy driving, which is a major cause of traffic accidents. The system uses state-of-the-art technology to identify early indicators of driver fatigue, such as image processing and machine learning. SafeAwake aims to reduce the risk of drowsiness-related accidents by monitoring drivers' facial expressions and eyelid movements in real-time and sending timely alerts to them. The system's target audience includes long-distance truck drivers, intercity bus drivers, and other professionals engaged in extended periods of driving, acknowledging their heightened susceptibility to drowsy driving incidents.

1.2 Design goals

Functionality:

SafeAwake is engineered to deliver robust functionality, precisely aligned with the program's core objectives. The system excels in interpreting facial expressions and eye movements with utmost precision SafeAwake guarantees reliable detection of symptoms of drowsiness by utilizing advanced machine learning and image processing algorithms in Python. This allows for timely alerts through visual notifications and auditory alarms. The system must detect signs of drowsiness with a minimum of 95% accuracy to minimize false alarms (false positives or false negatives).

Versatility and Cost Effectiveness:

SafeAwake was designed with practical factors in mind, catering to the specific requirements of its wide range of users, which includes drivers of personal vehicles, trucks, intercity buses, and taxis. The Raspberry Pi 4 Model B, which provides an affordable yet reliable hardware solution, is a sensible alternative with its other affordable components such as Raspberry Pi Camera, Raspberry Pi 4 Micro HDMI, power supply and MicroSD Card 32 GB. This calculated move guarantees both broad accessibility and high tech at the same time.

Accuracy:

SafeAwake's design places a high priority on accuracy. The system uses complex Python machine learning and image processing techniques to achieve accuracy in both functional and non-functional aspects. By improving the accuracy of facial expression and eye movement analysis, this decision reduces false alarms and lays the groundwork for trustworthy drowsy detection. System produces the alarm with a minimum of 95% accuracy (up to 5% error) when it recognizes symptoms of drowsiness.

Completeness:

The system architecture provides a comprehensive solution for drowsiness detection and prevention by addressing all defined user criteria. SafeAwake has all the necessary components for a comprehensive and successful drowsiness detection system, including real-time monitoring of drivers' conditions and prompt notifications via visual and auditory signals. Systems completeness comes with the camera capturing high-quality images with a minimum resolution of 720p along with accurate analysis of facial expressions and eye movements to determine the driver's drowsiness and sounds an audible alarm with a minimum volume of 60 decibels when signs of drowsiness are detected.

Enhanced Road Safety:

The primary design goal is to harness the capabilities of the SafeAwake system to significantly improve road safety. The integration of advanced image processing algorithms, implemented on the Raspberry Pi 4 Model B, ensures real-time analysis of drivers' facial expressions and eyelid movements. The software, developed predominantly in Python, facilitates efficient and accurate detection of drowsiness, enabling timely alerts to prevent potential accidents.

Efficiency:

SafeAwake's architecture prioritizes efficiency to ensure maximum resource utilization without sacrificing performance. The system is designed to run smoothly on the Raspberry Pi 4 Model B while balancing resource use and responsiveness. High throughput and low latency are essential for timely alerts in dynamic driving settings, and this efficiency helps to achieve both. The system should also use the vehicle resources such as vehicle battery efficiently.

Reliability:

The efficiency of SafeAwake's architecture is given first priority, guaranteeing maximum resource use without sacrificing speed. The system is designed to operate smoothly on the Raspberry Pi 4 Model B, striking a careful balance between responsiveness and resource usage. Efficient system design achieves high throughput and low latency, which are essential for timely notifications in dynamic driving circumstances. The system is reliable in various lighting scenarios, on different types of roads, and with drivers of varying physical attributes.

Optimization:

The system's design strikes the ideal balance between time and space considerations. Each component's code chunks are carefully optimized to guarantee smooth operation on the Raspberry Pi 4. Pretrained model should provide high accuracy results on the embedded hardware. By ensuring that resources are used wisely, this optimization enhances the system's overall responsiveness and performance.

Scalability (Flexibility):

SafeAwake promotes scalability and flexibility in its architecture, anticipating changing user needs and technical improvements. Future improvements and modifications can be accommodated by the modular architecture, which guarantees that the system will stay relevant and flexible in response to shifting user needs and technical environments. In the field of drowsiness monitoring systems, SafeAwake seeks to be a versatile and long-lasting solution.

Maintainability:

SafeAwake is designed with maintainability in mind, recognizing the importance of easy updates and modifications throughout its lifecycle. Code maintenance is made easier by the use of Python, which enables easy upgrades and modifications to meet changing needs or cover new problems. Libraries that are used in the development of the algorithm allows long-lasting support.

Compatibility:

SafeAwake's architectural design places a high priority on compatibility, realizing how important it is for systems and platforms to work together seamlessly. Integration of Raspberry Pi Model 4 and related Python algorithms used by machine learning and image processing are well-suited for usage of various vehicles and run seamlessly without any inconvenience. Because of the system's adaptability, it can be integrated with many kinds of vehicles, operational scenarios, and potential future integrations.

Security:

When designing SafeAwake, security was given top priority. Because the data being processed is vital, the system is designed to guard against unwanted access and secure sensitive information. There are safeguards in place to preserve the confidentiality and integrity of data gathered while operations are underway.

1.3 Definitions, acronyms, and abbreviations

This part includes a brief list of Definitions, Acronyms, and Abbreviations to help readers navigate specific terms in our high-level design report. By providing brief explanations, we ensure that both technical and non-technical people understand the terms clearly. This section acts as a reference guide, allowing readers to easily locate and comprehend key terms, abbreviations, and acronyms used throughout the document.

- **EAR (Eye Aspect Ratio):** Quantifies the ratio of the eye's width to height, a crucial metric for understanding eye-related movements and expressions in the context of drowsiness detection.

- **MAR (Mouth Aspect Ratio):** Assesses the ratio of the mouth's width to height, a significant parameter for analyzing facial expressions, especially yawning.

- **MTCNN (Multi-Task Cascaded Convolutional Neural Networks):** An advanced neural network architecture extensively used for tasks like object detection and facial recognition, contributing to SafeAwake's ability to identify and analyze facial features.

1.4 Overview

The SafeAwake system integrates cutting-edge image processing and machine learning technologies to monitor drivers' states, encompassing wakefulness, drowsiness, and yawning. The SafeAwake is a modular system that includes a camera module, a Raspberry Pi 4 Model B for real-time processing, an alarm system with visual and auditory notifications, and a sustainable power source powered by the car's battery. This guarantees scalability, compliance, and dependability, resulting in a strong solution that lowers the risk of accidents caused by drowsy drivers. In the following segments of this high-level design report, we will explore the details of each component, examining design strategies, factors to consider, and possible challenges.

2. Current software architecture (if any)

Since we do not have an established system, in this part we will mention the supply of materials required for our project.

Materials Supplied:

- Raspberry Pi 4 Model B.
- Raspberry Pi Camera and the necessary cables for the camera settings.
- MicroSD Card 32 GB.
- Raspberry Pi Power Supply.
- Raspberry Pi 4 Micro HDMI.

Picture 1: Raspberry Pi 4 Model B with Pi Camera

3. Proposed software architecture

3.1 Overview

The software architecture section generally contains detailed functional, software and hardware information that we will use in our system. First of all, we have information about the subsystems that will be included in our system and how they will be located in which part of our system. It also includes information about the software and hardware features we will use for our system and how we will store our data. In addition to all this, this section examines in detail the security and access controls of our system and the boundary conditions it may encounter.

3.2 Subsystem decomposition

Our project consists of 6 different subsystems such as User Interface, Object Detection, Facial Feature Extraction, Machine Learning, Image Processing and Database. These subsystems work in communication with each other.

The user interface is a subsystem that acts as a bridge to other subsystems. When the vehicle starts, the system becomes operational with the message “Welcome to SafeAwake”. The system starts the operation of the “object detection” and “facial feature extraction” subsystems when the “The camera will start...” message appears on the user interface. “Object detection” and “facial feature extraction” subsystems must perform face tracking operations in real time. Since the driver's face will be monitored and analyzed throughout the driving, the object detection subsystem comes into play at this stage. At the same time, the driver's mouth and eye-opening ratio will be monitored, so the “facial feature extraction” subsystem comes into play in this case. In addition, the database subsystem comes into play at this stage, as we permanently keep the photos taken while we are asleep in our system, and the videos taken while driving are kept in our database for 3 days. Therefore, the system will constantly be in communication with the database.

In addition to all this, we use “machine learning” and “image processing” subsystems. We benefit from these subsystems at the stage where we will teach the machine information about whether the driver is drowsy or yawning. While “image processing” processes image data for face and eye tracking, machine learning uses trained models to predict drowsiness levels.

3.3 Hardware/software mapping

Hardware and software mapping for our drowsiness detection system involves a combination of hardware components of Raspberry Pi equipment and some software dependencies for Raspberry Pi. Our system will be implemented on Raspberry Pi 4. In addition, we are planning to use Raspberry Pi cameras to take real-time face movements data. We aim to use the vehicle battery as the power source to provide continuous data input.

For the software part, we are planning to develop our system using Python 3 as the primary programming language. For image processing our software requires OpenCV, for facial extractions we are planning to use Dlib, for deep-learning based drowsiness detection models we are going to use TensorFlow and PyTorch, for convenience functions we have imutils, for numerical operations we are going to use numpy, also to handle command-line arguments our software requirements include argparse, for a powerful face detection system we also plan to use MTCNN. All together, these software requirements allow us to implement our drowsiness detection algorithm. In addition, our system with these hardware and software components can be integrated into different types of vehicles such as public transportation or personal cars. Our drowsiness detection system will be efficient with this hardware/software mapping.

3.4 Persistent data management

The persistent data management strategy for our system is designed to associate with the dynamic nature of the data we generate during the real-time video processing on the Raspberry Pi 4 board. We are planning to take advantage of Redis for in-memory storage to address the instant processing needs. Redis is a high-performance key-value store. Redis is useful in helping us with quick access to data and allows us to get quick responses to detect moments of driver's drowsiness and helps us get triggers on time.

At the same time, we are planning to implement a relational database for long-term data holding using PostgreSQL. It will store historical data such as captured photos when drowsiness is detected and video streams while driving (to be stored for 3 days). A relational database will help us to organize and streamline querying and facilitating activities.

To increase data security and privacy, we are planning to establish user roles and permissions within the SafeAwake system at the memory and database parts. This step will ensure that our private information such as facial features and drowsiness events are protected against unauthorized access. In our system, only authorized personnels will have access control to the data.

3.5 Access control and security

SafeAwake, our Driver Drowsiness and Yawn Detection System, has been designed with a strong focus on access control and security. This system aims to monitor driver alertness and prevent accidents caused by drowsiness or fatigue, ultimately contributing to safer road conditions, and reducing the likelihood of accidents.

Access control limitations are crucial to ensure that only authorized personnel can manage and monitor the SafeAwake system. Access to the system will be directly integrated into the vehicle's existing systems, and specialized training will be provided to designated individuals within the transportation company to operate and maintain the system effectively. Furthermore, access to the SafeAwake system and its data will be restricted to a limited number of individuals based on their authority.

User roles and permissions will be established within the SafeAwake system to control access and actions. While designated administrators can configure system parameters, access to sensitive operations such as accessing historical data will be restricted to authorized personnel to prevent misuse or unintended modifications.

Data security is of paramount importance, and SafeAwake aims to take strong measures to protect sensitive information. Therefore, efforts will be made to ensure that data is transmitted securely between system components and to reduce the risk of unauthorized access or tampering.

The system will be designed to operate without a constant internet connection, minimizing the risk of external attacks through the internet and focusing instead on internal security measures to protect against potential threats from within the organization.

To address potential security issues caused by malicious employees, SafeAwake enforces strict access controls and creates user roles and permissions to prevent unauthorized access to the system.

In conclusion, SafeAwake places a high emphasis on access control and security to ensure the reliable and secure operation of the system. By mitigating risks associated with internal and external threats, SafeAwake aims to contribute to safer road conditions and reduce the likelihood of accidents caused by driver drowsiness. Through its comprehensive access control and security measures, SafeAwake is committed to enhancing road safety and promoting a secure driving environment for all.

3.6 Global software control

The Global Software Control of the SafeAwake Drowsiness Detection and Alerting System is a critical component that governs the overall functionality, coordination, and adaptability of the software modules embedded in the system. This section outlines the key aspects of the global software control mechanism, encompassing various functionalities and ensuring seamless integration across the entire system architecture.

System Initialization and Configuration

The SafeAwake system undergoes a meticulous initialization process upon vehicle startup. The software control module is in charge of coordinating the system's preparedness, setting up the camera, and confirming that the camera is positioned correctly for best performance. Configuring settings with an intuitive interface is another step in this phase that guarantees usability and practicality.

Real-time Monitoring and Analysis

The software control's core functionality is its ongoing monitoring and analysis of real-time data from the camera module. The device evaluates drivers' eye movements and facial expressions using sophisticated image processing and machine learning algorithms, quickly identifying indicators of fatigue. The major objective of the system, to improve road safety, is met by the software control, which guarantees the efficiency and accuracy of this analysis.

Alerting and Warning Systems

The alerting and warning mechanisms are activated by the software control when it detects possible drowsiness. To properly communicate with the driver, this entails turning on visual alerts, auditory alarms, and relevant messaging. The control module enhances the synchronization and timing of these notifications, which adds to the dependability and user-friendly architecture of the system.

Performance Optimization

The software control is essential to maximizing system performance, and efficiency is a fundamental component of SafeAwake. The control module maintains high levels of accuracy and responsiveness by optimizing settings, fine-tuning algorithms, and implementing optimizations through systematic monitoring and frequent evaluations.

3.7 Boundary conditions

SafeAwake is intended to operate under specific boundary conditions to ensure optimum performance, transparency, and a commitment to continuous improvement in increasing driver safety on the road.

Firstly, environmental conditions must be taken into account. It is acknowledged that SafeAwake may have difficulty working optimally in challenging lighting scenarios, including both day and night conditions. Additionally, adverse weather conditions such as rain, fog, extreme temperatures, or humidity are recognized as potential challenges within the system's boundary conditions. In addition, the system must be resilient to vibrations and impacts commonly encountered in a vehicle. Physical obstructions or damage to the camera may compromise image quality, influencing the accuracy of drowsiness detection. Despite these difficulties, ongoing efforts are made to enhance the system's adaptability to the unpredictable nature of real-world driving environments.

Secondly, recognizing the diversity among drivers is paramount to the success of our Driver Drowsiness and Yawn Detection System. The boundary conditions address factors such as age, gender, and physical conditions to ensure the inclusivity of our technology. While designed to cater to a broad user base, these conditions acknowledge potential variations, guaranteeing that the system remains effective across a wide range of driver demographics.

Thirdly, the effectiveness of the system's facial analysis and warning system depends on the unobstructed visibility and clarity of the driver's face within the camera's field of view. In scenarios where the face cannot be fully detected, limitations may arise in capturing facial expressions, eye and mouth movements. This may cause false positive or false negative alarms. Recognizing the potential for false positives and negatives is central to our commitment to accuracy. However, by focusing on constantly improving these boundary conditions, we aim to ensure that the system strives for the highest level of precision, thus minimizing the occurrence of both false positives and false negatives.

Lastly, compliance and safety regulations must be adhered to. The system is required to comply with all applicable safety and traffic laws and regulations in the regions where it is deployed. Additionally, the system must operate seamlessly without interfering with the vehicle's normal functions or compromising other safety systems. Any interference may compromise the overall safety of the vehicle and its occupants. These conditions are crucial to ensure that the system does not cause any safety hazards or legal issues.

Finally, the data handling and deletion policy must be considered. The system's ability to permanently delete video streams within 3 days while retaining photo frames in the database is contingent upon the proper implementation of data deletion and handle mechanisms. Any failure in this process may lead to unintended data retention beyond the specified timeframe.

In addition to this, external noise, music volume and possible malfunctions may affect the audibility of the alarm, which will be set to a minimum sound level of 60 decibels, which may affect its ability to effectively warn the driver.

SafeAwake system is an innovative solution to prevent accidents caused by driver fatigue and drowsiness. However, to ensure its effectiveness, it is crucial to consider the boundary conditions that may impact its performance. Environmental conditions, lighting and road conditions, compliance and safety regulations, and data handling are all critical factors that must be taken into account when deploying the system. By considering these boundary conditions, we aim to ensure that the SafeAwake system operates effectively and contributes to safer driving experiences.

4. Subsystem services

SafeAwake is a hardware device with 4 different subsystems. These subsystems are “user interface”, “object detection”, “facial feature extraction”, and “database”. These subsystems work in communication with each other. The “object detection” and “facial feature extraction” subsystems will start with the display of the “The camera will start...” message on the user interface. While the system tracks the driver's face and eye movements throughout the drive, it stores the video it takes in the database, deleting it from the database after 3 days, and keeping the photo it takes permanently.

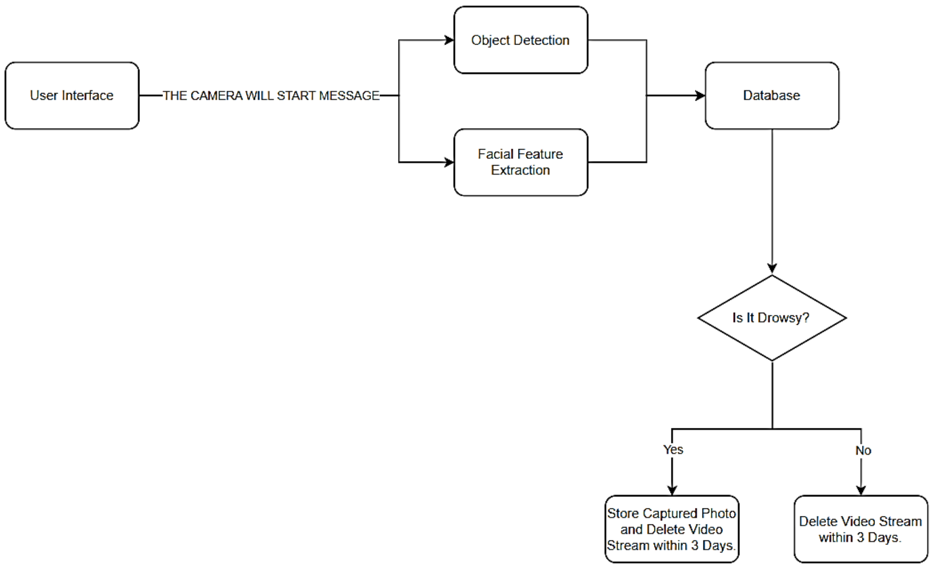

Figure 1: Relationship Between Subsystem.

5. Glossary

Object detection: It is a computer vision task that involves identifying and locating multiple objects within an image or video frame and assigning them specific labels or categories.

Image Processing: It refers to the application of algorithms and techniques to analyze and improve the images taken by the camera.

Machine Learning: It is a subset of artificial intelligence. It allows computers to learn from data and improve their performance over time without any programming.

User Interface: It is a point of interaction with the user and the system.

Facial Feature Extraction: This is extraction of the facial expressions and characteristics from the images taken from the camera.

Python 3: It is the latest version of Python programming language. It is a high-level language and its readable and easier to use than the other programming languages.

OpenCV: OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library which provides us with a wide range of tools and functions for image processing, vision tasks and machine learning applications.

Dlib: It is a versatile C++ library. This library includes tools and algorithms for machine learning, image processing and vision tasks. For our system it is for facial feature extraction.

TensorFlow: It is an open-source machine learning framework developed by Google, primarily used for building and training neural network models.

PyTorch: It is an open-source machine learning framework that facilitates building and training neural networks using dynamic computational graphs.

Redis: Redis is an open-source, in-memory data structure store known for its speed and versatility, often used as a cache, message broker, or database.

PostgreSQL: It is an open-source relational database management system. For our system it is useful for long-term data storage, and it facilitates structured data management, efficient querying and supports tasks such as trend analysis and model training.

Imutils: It is a library for OpenCV. It simplifies common image processing tasks and provides a set of utility functions such as resizing, rotating, and displaying images.

Argparse: It is a module Python Standard library which facilitates the parsing of command-line arguments.

NumPy: It is a powerful numerical computing library for Python.

6. References

- Benantar, M. (2005). Access Control Systems: Security, Identity Management and Trust Models. Springer Science & Business Media.
- Dehankar, V., Jumle, P., & Tadse, S. (2023). Design of Drowsiness and Yawning Detection System. In Proceedings of the Second International Conference on Electronics and Renewable Systems (ICEARS-2023) (pp. CFP23AV8-ART). IEEE Xplore.
- Maheswari, V. U., Aluvalu, R., Kantipudi, M. P., Chennam, K. K., Kotecha, K., & Sain, J. R. (2022). Driver Drowsiness Prediction Based on Multiple Aspects Using Image Processing Techniques. IEEE Access, 10(1), 1-1
- http://dlib.net/
- https://www.skillmaker.edu.au/project-boundaries/
- https://redis.io/
- https://pytorch.org/
- https://opencv.org/
- https://nanonets.com/blog/machine-learning-image-processing/