ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Software Requirements Specification

CENG 407

Drowsy Driver Detection System Project

Advisor: Dr. Serdar ARSLAN

Names of members

Mert Alp Kuvandık 201911038
Bahu Tongal 201811058
Volkan Mazlum 201811045

09/12/2022

Table of Contents

List of Figures

1. Introduction

Driver drowsiness detection means that a technology to helps avoid the accidents caused by the drowsiness of the driver. According to data from the US National Highway Traffic Safety Administration (NHTSA), approximately 100,000 accidents (1.5% of all accidents) and approximately 1500 deaths per year (4% of all fatal accidents) occur each year due to tired and sleepless driving. A method that can be a solution to these problems, developing applications is a fundamental goal around the world. Object detection is the basic method we have that can detect this problem. We must develop it and turn it into an application. Different research around the world has been made and continue to be done about drowsy driver detection. In our project, we can develop a method that can result in the problems presented in the project. We can even put forward a project that can be integrated into real life by taking it further concerning other projects. In addition, we are proud to be able to apply artificial intelligence and machine learning to such a problem occurring worldwide. The core focus of our app is drivers. Operations such as detection, estimation, and inspection proceed entirely through the driver. Car companies are also the people who need to integrate this app with cars.

1.1. Purpose

In this project, our goal is to develop a system that detect the drowsiness of driver from momentarily videos of them. The main purpose of this project designs a warning system for the drivers to decrease the accidents that occur from drowsiness. This project not a solution for the drowsiness, it is just a warning and precaution system. To prevent adverse events that may occur during car driving, the images of the driver will be watched instantly and thanks to our application, the behavior of the driver will be examined. In any sleep state our alert system will work, as a result we will try to eliminate the problem without panicking the driver. While doing the project, we design to provide diverse software requirements such as Artificial Intelligence, Deep Learning, Machine Learning. This SRS documents contains the project requirements, and Software Requirements Specification for drowsiness detection.

1.2. Scope

Our project is aimed to make an application that will determine the state of sleep by examining the state and movement of the user. This determination will not be based solely on the application of computer vision. Of course, there will be instant eye, mouth, and face detection, but it is aimed to use machine learning while eyes detection. In mouth detection, using facial points obtained through different libraries such as dlib, the number of yawning will be counted instantly, and a warning will be given to the user after passing a certain threshold. Just keeping the user's eyes closed for a long time will not be the only option for detecting the user's condition. There are several different projects in this area. We aim to make an integrated system with the car and not focus only on eye conditions. In addition, it is aimed to improve the detection of drowsiness according to different environmental conditions. Our system will mainly include:

- Face (Object) Detection for detecting drowsiness
- Deployed trained model for classification
- Warning System

1.3. Product Overview

The Drowsy Driver Detection System is a project that aims to avoid deadly traffic accidents. Driver sleepiness is one of the leading causes of traffic accidents. They get weary or exhausted and may fall asleep because of long-distance travel, stress, or sleep deprivation. We want to avoid and decrease such accidents by developing a sleepy detecting system. Therefore, we will achieve this goal and launch it, thanks to the application we have developed. In this section, the IDE, algorithm, and similar things to be used in the project will be briefly mentioned. In addition, general terms to be used in the project will be briefly mentioned, and important issues such as User Characteristics, Overview of Functional Requirements will be mentioned.

1.3.1. Product Perspective

Our project includes different computer science concepts: detection algorithms (Yawning detection method, Eye blink monitoring method), machine algorithms and computer vision. We will use many programming languages for these strategies, including Python and C#. Furthermore, different algorithms (YOLO, CNN and so on), libraries (Flask, Scikit-learn, Keras, and Tensorflow), and IDEs (Jupyter Notebook, Spyder, Kagle) are utilized in the development process. Different algorithms and different IDEs can be used for different situations. Because, as we saw in the literature review, there are multiple algorithms used. No matter how much CNN we said we would use, we believe that using pre-trained models such as Inception and VGG thanks to transfer learning can give us better results in developing our application. We will not only do detection but also turn our project into a web application on a web platform. We will make an application that users will want to use, both in terms of usability and appearance.

1.3.2. Glossary

Term	**Definition**
Actor	An actor can be a user like driver. It can be also another software system that interacts with the system.

Python	Python is high-level programming language, which is preferable when Machine Learning, Deep Learning algorithms are used.
Software Requirement Specification (SRS)	A document that contains a detailed explanation of the system's operations, needs, restrictions, and operating conditions. The following document is an SRS document.
User	Users can be drivers who will drive and provide us with the dataset taken from their videos and make our application good.
Object Detection	The detection of objects is a common computer science technology. The computer decides and recognizes what the thing is using visual methods.
Convolutional Neural Network (CNN)	A particular type of deep neural network is the convolutional network, sometimes known as CNN or ConvNet. It's a feed-forward artificial neural network with a high level of complexity.
Artificial Neural Network (ANN)	An artificial neural network is a computer model made up of many processing components that receive inputs and output results depending on predetermined activation functions.

1.3.3. User Characteristics

1.3.3.3. Driver

The driver is the person responsible for driving the car. The image will be taken by the camera system. The driver is the person whose image will be processed by the other system and drowsiness detection will be made. Tasks of driver are given below:

- Driving a car
- Starting the application

1.3.3.4. System

It is the characteristic system that will do the main operations of the application. All face detection operations, warning operations, making the camera shoot video are the part of the trained model. It constitutes the main and very important part of the project. Tasks of system are given below:

- Face recognition system
- Implementation of the trained model
- Making Classification (Drowsiness detection)
- Taking video of the driver
- Obtaining frames from video
- Detection of the mouth and eye areas
- Activation of the warning system in case of detected drowsiness
- Ensuring system recovery in any system error

1.3.4. Overview of Functional Requirements

Use Case Title	Description
Capture Frame	Obtains the pictures from momentarily videos.
Capture Real Time Video	Record the real time video of driver.
Detect Mouth	Determines the position of mouth and detect yawning
Detect Eyes	Detects the eyes of driver and calculate aperture size
Detect Face	Detect the facial expression of driver
Detect Status of Head	Detect the position of the head of driver
Performs Alerts	Give a alarm when detect drowsiness
Detect Drowsiness	Calculate the data with given algorithms and detect drowsiness

1.3.5. Limitations

This system starts to take a video of the driver after startup. The frames obtained from this video; it tries to predict the driver's condition with the help of pre-trained models. According to this prediction result, the system may give an alert. That is, the system decides for itself whether there is a suspicious situation or not. In addition, thanks to the mouth opening obtained while performing face detection, it can reach the yawning number and give a warning after a certain number. Therefore, it must always work correctly. There are some cases where this system requires limitations.

- An attempt has been made to find a solution to the drowsiness problem. However, it is difficult to find a solution for each different situation. Likely, the project will not work well in certain environmental conditions.

- In the eye recognition process, which plays an important role in the perception of the drowsiness state, it becomes difficult for the system to function properly when different eye positions occur. For this, mouth detection is also used, but the same problem applies here.

- Keeping the user's head down causes the system to misunderstand the status of the drowsiness.

- There is no dual-user availability.

- If the system cannot identify the status of drowsiness in seconds, the system should give an alert about the identification.

- The model we will develop for this project strongly depends on ambient light.

- Face, eye detection etc. are strongly dependence on distance of camera from driver face.

1.3.6. Dependency and Assumptions

We will assume that this system is used under some the following assumptions and dependencies.

- The device is connected to an integrated camera system in the car.

- For now, it is assumed that the driver is driving in the dark.

2. Reference

- IEEE Computer Society. (2009). IEEE Standard for Information Technology—Systems Design—Software Design Descriptions. IEEE. New York: IEEE. IEEE Std 1016
- Sommerville, I. (2016). Software engineering. Boston: Pearson Education Limited

3. Specific Requirements

3.1. External Interfaces

3.2. Detailed Description of Functional Requirements

3.2.1. Use Case Diagram

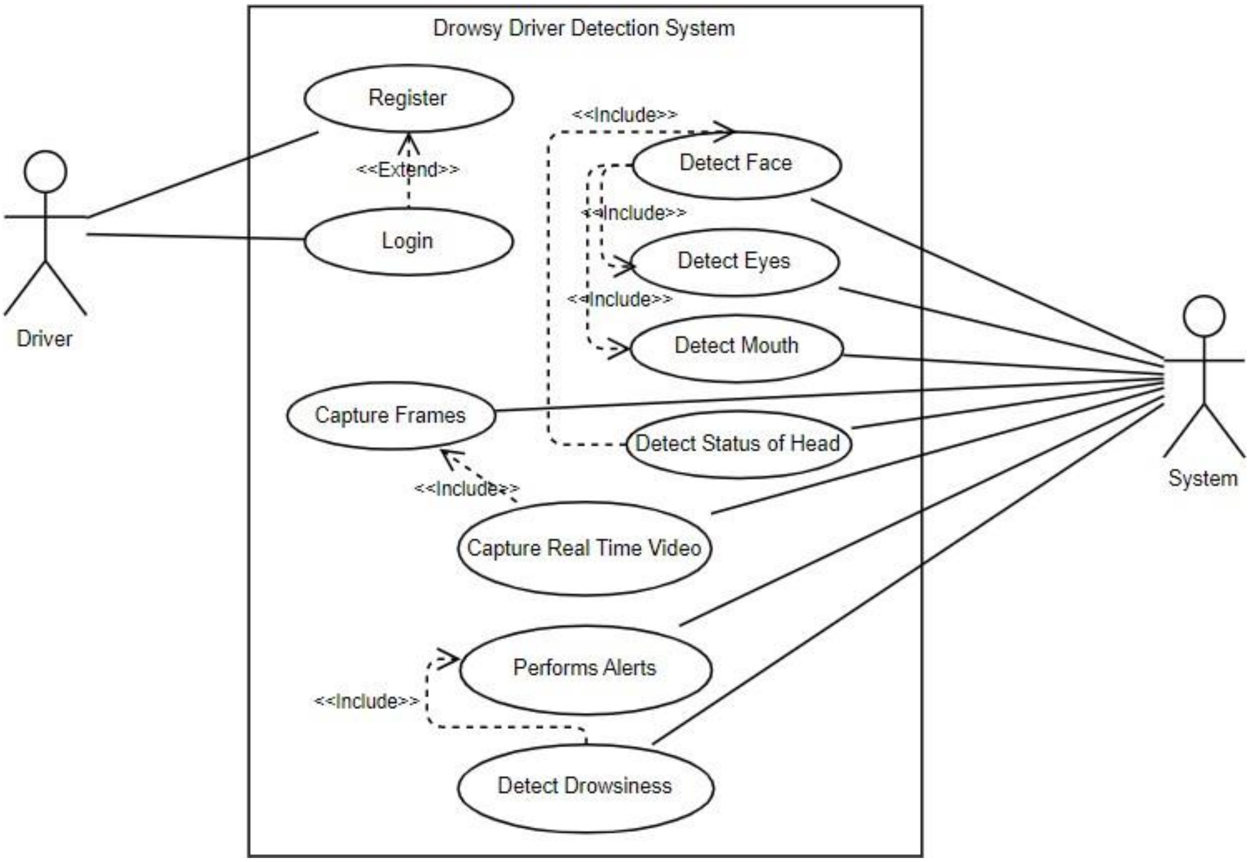

Figure 1: Student Use Case Model

3.2.2. Driver Use Cases

Use case Id	1
Use case Name	Register
Description	It is the action that the user must do in order to use the system features.
Precondition	The user must enter e-mail, name, surname, username, and password to register in the system.
Related Use Cases	**-**
Postcondition	The user has successfully registered in the system.
Main Flow	1. The user first opens the application. 2. User enters their first and last name, mail, username, and password. The user also ticks the "I am 18 years old or older" and "I accept the End-User Agreement" boxes. 3. The system checks the entered information and approves it. (A1) 4. The user successfully registers in the system. 5. The user logs into the system.
Alternate flows	<A1 – A. If the first or last name, email, username, or password entered by the observer is/are not valid System displays "first or last name, email, username or password is not valid" error message. User will enter valid information. Use case returns on step 2 of normal flow. B. If the email entered by the student exist in the system. System shows "The email address you entered already exists in the system. Please enter another email address or click "forgot my password" button" error message. Use case returns on step 2 of normal flow. >

Use case Id	2
Use case Name	Login
Description	It is the system that user loads the information system to connect and use it.
Precondition	The user must be enrolled the system to login it.
Related Use Cases	Register
Postcondition	The user successfully login and use system.
Main Flow	1. The user opens the system 2. Going to login page 3. Enters the information (A1) 4. When user enters the login information, system will start. 5. Camera system will start automatically.
Alternate flows	<A1- The user opens the system, if she/he is registered, she/he enters by entering her/his information, if she/he is not registered, logs in as a registration. >

3.2.3. System Use Cases

Use case Id	1
Use case Name	Capture Frame
Description	It is the process of obtaining pictures into frames for use in the machine learning algorithm from the video obtained by the camera.
Precondition	The camera system must be able to capture the video and process it in the background.
Related Use Cases	Capture Real Time Video- 2
Postcondition	Pictures, frames from the video were successfully extracted.
Main Flow	1. The camera system is activated. 2. The video of the driver begins. (A1) 3. The resulting videos are transmitted to the system instantly. 4. The video received by the system is converted into frames. (A2) 5. These frames are made ready to be used by the algorithm.
Alternate flows	<A1- A. The camera system may not have been successfully initialized. If this happens, the system will be restarted immediately, and an attempt will be made to restore it. > <A2- B. Received videos may not be framed. The process needs to be done again to automatically detect the problem and prevent the system from being disrupted. If this is not successful, the system should give an alert to warn the driver. >

Use case Id	2
Use case Name	Capture Real Time Video
Description	It is the process of recording the real time videos of drivers momentarily.
Precondition	The driver must be register and starts the system before driving.
Related Use Cases	Capture frame-1, Detect Mouth-3, Detect eyes-4, Detect face-5, Detect status of head -6
Postcondition	Starts and registers the system successfully
Main Flow	1. Register the system (A1) 2. The system starts 3. Turning on the camera 4. Starting video recording of the driver 5. Instantly sending the video to the system to obtain the frames (A2) 6. Completion of the transaction
Alternate flows	<A1- A. The user may not have starts and register the system. If this happens system does not work and cannot gives warning messages > <A2- A. An error may occur while sending videos of the camera system to the host system. The video may be incomplete or not sent at all. In this case, either the driver should be shown an error message, or the system should be restarted automatically. >

Use case Id	3
Use case Name	Detect Mouth
Description	It is the process that will determine the number of yawning of the mouth opening with the necessary mathematical operations and algorithms from the obtained face picture.
Precondition	The facial view of the driver must have been obtained from the videos.
Related Use Cases	Detect Face- 5, Capture Frame- 1
Postcondition	The driver's mouth has been successfully detected.
Main Flow	1. The video images of the driver have been uploaded to the system and the frames have been obtained. 2. From these frames, it is determined whether his mouth is open or closed, thanks to the mathematical formulas at hand. (A1) 3. Opening each driver's mouth wide counts as a yawn and is retained in the system. 4. After this number passes a certain stage, the user is warned.
Alternate flows	<A1- A. Error in detecting the mouth may occur because the picture is not clear. Either it is necessary to obtain frames from the video again, or the picture needs to be made clearer with different applications. >

Use case id	4
Use case name	Detect Eyes
Description	It is the process that will detect the aperture size the eye of user.
Precondition	The detection algorithms must be defined to system and measurement of aperture size calculated with maximum efficiency.
Related Use Cases	Detect Face- 5, Capture Frame- 1, Detect drowsiness - 8
Postcondition	The drivers eye aperture size has been successfully calculated and detect.
Main Flow	1. The video images of the driver have been uploaded to the system and the frames have been obtained. 2. From these frames, it is determined whether his eye is open or closed, thanks to the mathematical formulas at hand. (A1) 3. Calculating the aperture size and detect eyes are open or not. 4. After this number passes a certain stage, the user is warned.
Alternative Flow	<A1- A. Error in detecting the eye aperture size may occur because the picture is not clear. Either it is necessary to obtain frames from the video again, or the picture needs to be made clearer with different applications. >

Use case id	5
Use case name	Detect Face
Description	It is the process that will detect the facial expression of the user.
Precondition	Instant videos of the driver must be transmitted to the system and frames must be obtained.
Related Use Cases	Detect Status of Head- 6
Postcondition	The facial expression of the driver was successfully obtained.
Main Flow	1. The frames of the driver obtained from the video are found in the system. 2. The algorithm that will reveal the facial points is applied to these frames. 3. According to these facial points, eye and face extraction processes are continued in different places. (A1) 4. Thanks to machine learning algorithms (the sleep-wake state has been taught with different datasets before), the sleep-wake processes of the driver are continued in the detect drowsiness part.
Alternative Flow	<A1- A. Error in detecting the facial points may occur because the picture is not clear. Either it is necessary to obtain frames from the video again, or the picture needs to be made clearer with different applications. >

Use case id	6
Use case name	Detect Status of Head
Description	It is the process that will detect the position of the user which means that the users head down or upward.
Precondition	The position of the driver must have been obtained from the videos.
Related Use Cases	Detect face -5
Postcondition	The driver's head has been successfully detected.
Main Flow	1. The video images of the driver have been uploaded to the system and the frames have been obtained. 2. From these frames, it is determined whether his mouth is open or closed, thanks to the mathematical formulas at hand. (A1) 3. Looking the position of driver’s head and decide the change of station with that driver is awake or not. 4. After this number passes a certain stage, the user is warned.
Alternative Flow	<A1- A. This case process with the facial point detection algorithm. If all facial points cant detect with the algorithm, system must give a notice to the user. Whether this not works properly, lastly system will shut down and restart. >

Use case id	7
Use case name	Performs Alerts
Description	It is the process that gives a warning message to the driver when detects the drowsiness.
Precondition	The data collected from recording videos of driver.
Related Use Cases	Detect drowsiness - 8
Postcondition	When drowsiness detect it gives a warning message to driver.
Main Flow	1. The data collected from driver with camera system processed. (A1) 2. Respect to the parameters defined to the system, it decides to the driver drowsy or not. 3. If driver is drowsy, system gives an error message to the driver. (A2)
Alternative Flow	<A1- A. If there is a problem with image processing, the whole work must be done from the beginning. > <A2- B. Error may occur while system is generating warning message. If this happens system can be restarted, or a message sent to the drivers phone. >

Use case id	8
Use case name	Detect Drowsiness
Description	It is the part where algorithms that will detect the drowsiness state of the user are applied.
Precondition	The states of the user's face, mouth, and eyes must be obtained from the video.
Related Use Cases	Detect Face, Detect Mouth, Detect Eyes
Postcondition	The drowsiness status of the user could be detected accurately.
Main Flow	1. Our algorithm is trained thanks to the available dataset. 2. Training continues according to the obtained accuracy values. (These operations are not done instantaneously; the model has been trained before.) 3. When the results are improved, the eye expressions that will come from the driver instantly become suitable for detection thanks to this algorithm. 4. Thanks to the algorithm, the status of the user is determined. (A1) 5. According to this result, the main system is directed to the alarm system. (A2)
Alternative Flow	<A1- The user's condition may not be well predicted by the developed model. Instant mathematical operations that will control the eye-opening can enter the circuit. > <A2- The alarm system may not work, in this case, the alarm system must be restarted and ensured to work. If this is not the case, the driver must be notified of the problem differently. >

3.3. Non-Functional Requirements

3.3.1. Performance Requirements

Performance Requirements	**Definition**
Response Time	Real time video must be displayed on application screen with at most 4 seconds' delay. When face detection algorithm runs, delay of video will increase by 4 seconds. Also, when drowsiness is detected, alerts should be shown to driver less than 5 seconds.
Error Handling	When unpredictable failure occurs, system should inform driver about system's status. System should re-start as soon as possible.
Workload	The system tries to handle multiple different subsystems at the same time. System is going to be able to handle face detection, sending video via camera system, and getting frames from this video at the same time.
Scalability	This system will be accessed by a single device via our app. As a result, several people cannot utilize our system at the same time, and the system will not fail due to scalability issues.
Application requirements	There must be 20MB of free space on the smartphone for our application. The device's CPU speed or RAM is unimportant.

3.3.2. Safety Requirements

Safety Requirements	Description
Safe Controlling	When the system becomes unusable, the driver must be informed, and the system must be restarted.
Accurately and Safely Alerting	The system must report system faults by alerts accurately and safely.
Detection	The system needs to detect face, mouth, and eyes from frames of video and clarify the state of the driver if it detects drowsiness.

3.3.3. Security Requirements

Security Requirements	Description
Wi-fi Connection	The system should send video continuously via Wi-Fi if we use a web application. This connection should be proper.
Application Access	A system will be developed in which the videos will be automatically transferred to the system. These images must be transferred in a protected manner to apply the algorithm and clarify the state of the driver.

3.3.4. Software Quality Attributes

Software Quality Attributes	**Definition**
Reliability	Every functionality of the code that we created should be able to work without error in any typical scenario.
Robustness	The system will work properly in different environmental conditions and in different views.
Portability	The system should work on Windows.
Correctness	System must predict state of driver by drowsiness with high accuracy.
Learnability	System will be easy to understand and simple.
Maintainability	When a failure happens, the system must recover it without causing fatal changes to the code structure and it should be restarted to work properly.
Extensibility	System can be improved with additional features in terms of different algorithms. Therefore, new system is extendable.
Testability	The system must function without mistakes and be tested with various situations.
Efficiency	The system should work with maximum performance.
Usability	By using our system which is easy to use, our system can be operated.

3.4. Functional Modeling (DFD)

3.4.1 DFD Diagrams

In this section, data flow diagram is stated.

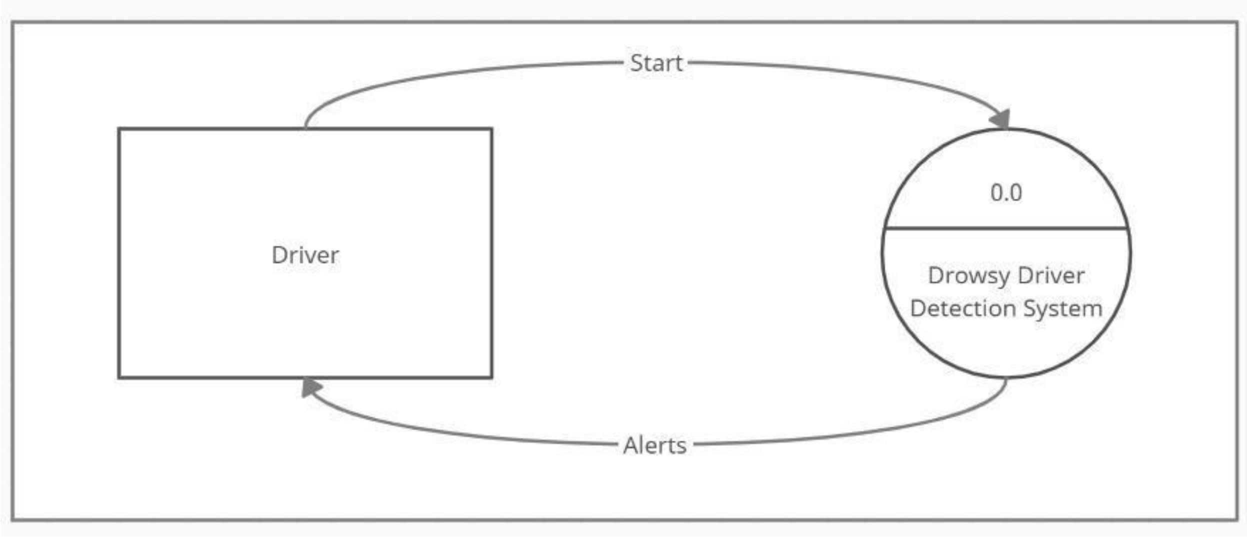

Figure 5: Context Diagram

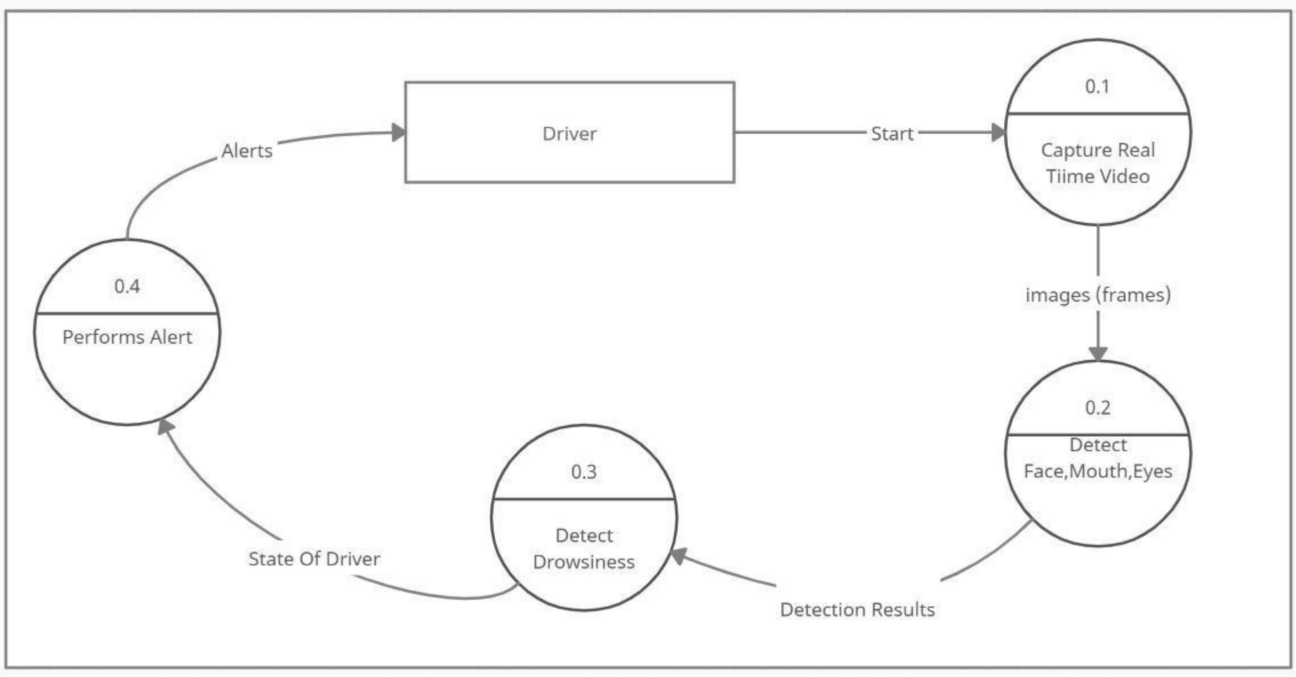

Figure 6: DFD Level-1

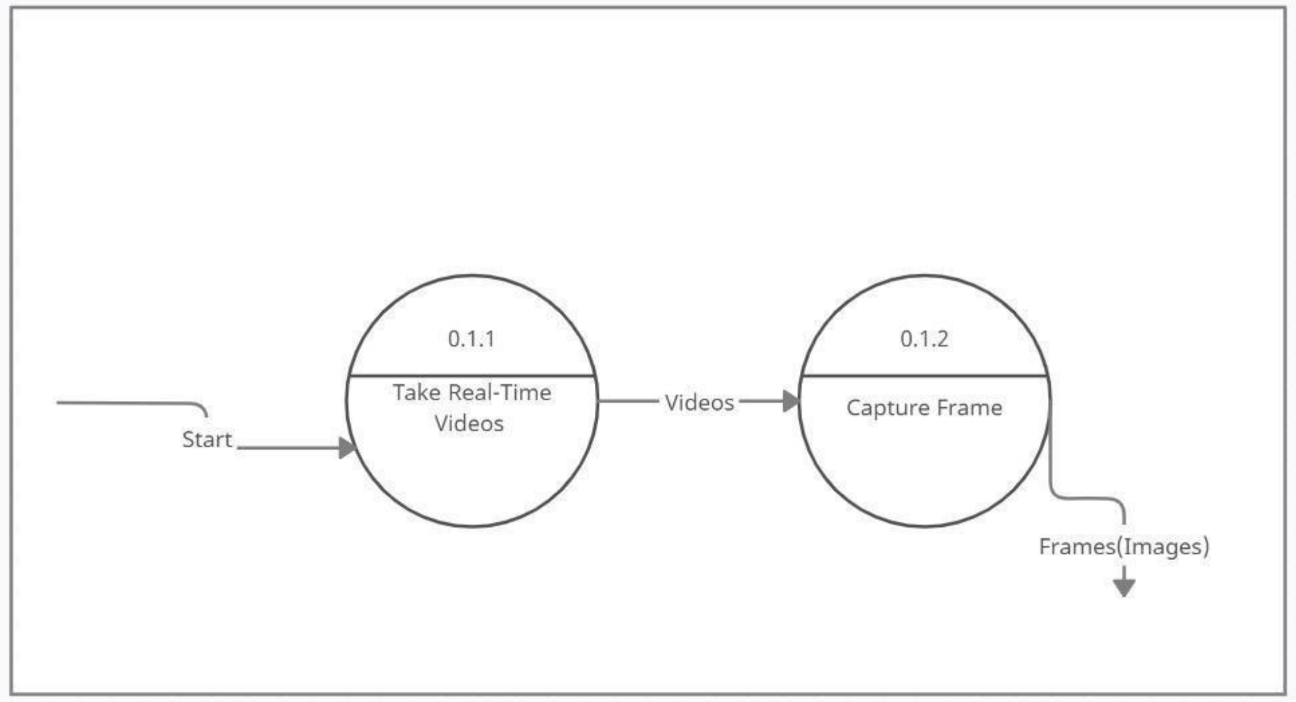

Figure 7: DFD Level-2 Capture Real-Time Video

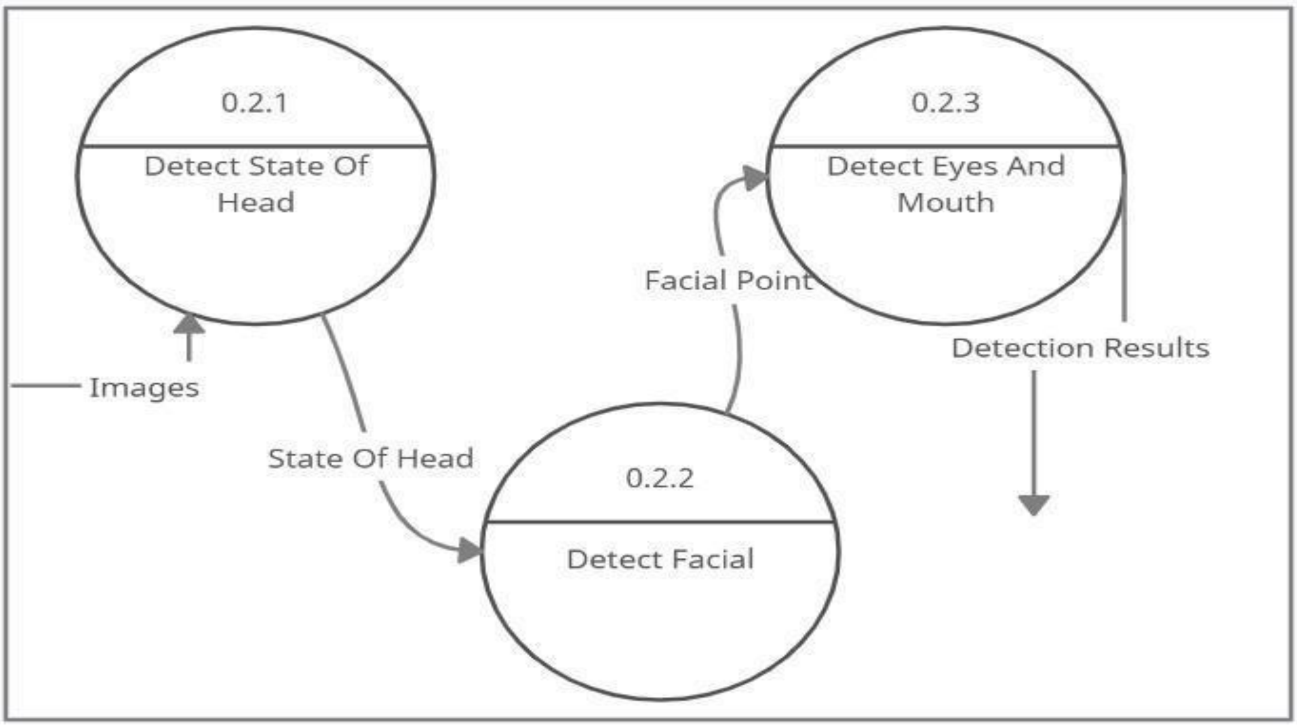

Figure 8: DFD Level-2 Detect Face, Mouth, Eye

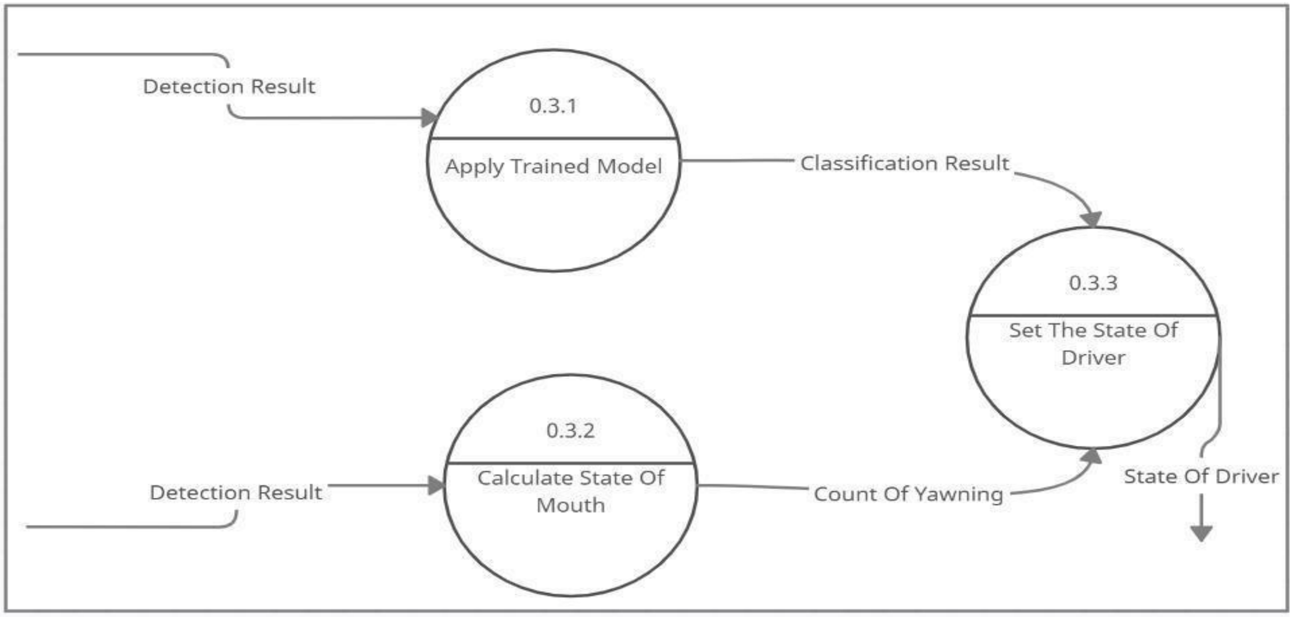

Figure 9: DFD Level-2 Detect Drowsiness