ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARMENT

Project Report

CENG 407

Innovative System Design and Development I

Advisor: ***Dr. Serdar ARSLAN***

Prepared By	**ID**
Volkan MAZLUM	201811045
Bahu TONGAL	201811058
Mert Alp KUVANDIK	201911038

06.01.2023

Contents

Abstract

The Drowsy Driver Detection System is a project aimed at preventing fatal traffic accidents. One of the primary causes of road accidents is driver fatigue. Because of long-distance travel, stress, or sleep deprivation, they get tired or fatigued and may fall asleep. By inventing a drowsiness-detecting system, we want to avoid and reduce such incidents. There are several projects and hundreds of papers based on the Drowsy Driver Detection System. Because sleepiness and weariness are two of the most common causes of accidents. Finding answers to these issues both lessens the pain load on countries' health systems and has the potential to reduce the number of deaths in accidents drastically. In the report we just stated, several methods and solutions devised for these have been published. These algorithms primarily aim to extract some aspects from facial expressions such as yawning, eye, mouth, and head movements. The research concludes with a thorough bibliography that includes brief explanations of each publication. Our major aim is to try to avoid accidents using the ways we will employ and will be utilized by us, as well as to reduce the number of deaths using new algorithms in our system.

Key words:

Object Detection, Machine Learning, Artificial Neural Network

Özet

Drowsy Driver Detection Sistemi, ölümlü trafik kazalarını önlemeyi amaçlayan bir projedir. Trafik kazalarının en önemli nedenlerinden biri sürücü yorgunluğudur. Uzun mesafeli seyahat, stres veya uykusuzluk nedeniyle sürücüler yorgunluk veya bitkin düşşme sonucunda uykuya dalabilirler. Bir uyuşukluk tespit sistemi icat ederek, bu tür olaylardan kaçınmak ve azaltmak istiyoruz. Uykulu Sürücü Tespit Sistemi'ne dayalı birkaç proje ve yüzlerce makale var. Çünkü uykululuk ve yorgunluk kazaların en yaygın nedenlerinden ikisidir. Bu sorunlara cevap bulmak, hem ülkelerin sağlık sistemleri üzerindeki ağrı yükünü azaltır hem de kazalardaki ölüm sayısını önemli ölçüde azaltma potansiyeline sahiptir. Bu raporda bunlar için geliştirilen çeşitli yöntemler ve çözümler yayınlandı. Bu algoritmalar öncelikle esneme, göz, ağız ve baş hareketleri gibi srücünün jest ve mimiklerini baz alarak çıkarım yapmayı amaçlar. Araştırma, her yayının kısa açıklamalarını içeren kapsamlı bir bibliyografya ile sona ermektedir. Tespit ettiğimiz ve hedeflediğimiz yolları kullanarak kazaların önüne geçmek ve sistemimizde yeni algoritmalar kullanarak ölüm sayılarını azaltmak temel amacımızdır.

Anahtar Kelimeler:

Nesne Tespiti, Makine Öğrenmesi, Yapay Sinir Ağları

1. Introduction

This paper contains a detailed description of our system. To that end, we will present a project Literature Review and explain the project's Software Requirement Specification Software Design Description. We will also present our Work Plan, which will outline how we built our project.

1.1. Problem Statement

It is a project that has the potential to give some improvement to automobile accidents, which are a worldwide problem that cannot be ignored. In this project, we will strive to solve several challenges and achieve a better outcome than previous projects in the literature. With its alert system, our system, which can offer categorization in various scenarios, will attempt to keep the driver away from situations that may harm the driver's life. Our project, which also focuses on other aspects, is anticipated to bring answers to technical issues such as data set challenges. When combined with machine learning, this project can operate more independently than existing detecting systems.

1.2. Solution Statement

To overcome this difficulty, we intended to apply algorithms that would identify solutions to problems found in the literature. We will aim to expand it further, mostly through CNN. Furthermore, it will be attempted to more properly categorize the photos acquired using approaches such as Inception v3. All these tools and strategies will be covered in the SRS and SDD publications mentioned in this article. We will obtain real-time footage of the driver's face, mouth, and other features by employing the camera. Face, mouth, and eyes are recognized utilizing dlib libraries, algorithms, and other artificial intelligence principles. As a result, our system will be able to incorporate detection, classification, and alerting systems.

1.3. Motivation

This project includes many engineering concepts such as Object detection, Machine learning, and many more. Thanks to this project, we improve ourselves regarding these concepts. The most important motivation behind this project idea is helping people by improving the previous system in terms of face detection and alert system.

2. LITERATURE SEARCH

Abstract

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. There are many projects developed on the Drowsy Driver Detection System and dozens of articles. Because drowsiness and fatigue are some of the main problems that cause accidents. Finding solutions to these both reduces the pain burden on the health systems of countries and can significantly reduce the number of deaths in accidents. Different algorithms and solutions developed for these have been published in the articles I just mentioned. These algorithms mainly focus on extracting certain features from facial expressions such as yawning, eye, mouth opening, and head movements. The study closes with a comprehensive bibliography that includes brief descriptions of each publication. Our main goal is to focus on trying to prevent accidents via the methods we will use and to be used by us and focus on reducing the number of deaths with new algorithms of our system.

2.1. Introduction

The Drowsy Driver Detection System is a program that controls and processes the drivers' states and movements, and facial expressions and draws the road accordingly. What we mean by drawing a road is that it includes different reactions according to the driver's situation. The system itself must decide this and keep the driver awake. On the other hand, it is thought to be a system that can give warnings to stop the driver from driving and to get enough sleep. In general, it is a project that every driver candidate will need and even should have.

According to the World Health Organization (WHO), millions of people die or become disabled in traffic every year in the world. According to statistics, the main cause of these accidents is driver fatigue and carelessness. According to the report of the American Automobile Association, 7% of all accidents and 21% of fatal accidents are due to driver fatigue. According to another study conducted in the USA, 2% of the accidents that occurred in 2009 were caused by users being sleep-deprived. Also, according to a 2017 Organization for Traffic Safety survey, 42.4% of drivers drove without at least one or more days of sleep, fewer than six hours of sleep, and severe for 87.9% of drivers. [1]

Sleeping during a trip is not something that happens suddenly. Specifies a process. During this process, some statuses indicate different situations such as yawning, the head falling forward, and the eyes being closed to closing. These need to be determined beforehand. Our work will also focus here. Our main goal is to examine the eye and mouth states with different formulas and algorithms and to detect the sleep state early. If this is achieved, many unfortunate situations are likely to be avoided.

In Section 2, we offered an overview of our research on the Drowsy Driver Detection approach and the problems that we can run into. In Section 3, we discussed our overall impressions of the literature, with a focus on the basic algorithms and different methods that we discovered. The following section also includes a condensed list of "Top Articles" that offer a detailed overview of the papers we evaluated, and we thought that they are very appropriate for our research. In Section 5, we organized our collection of papers further into the challenge areas that each article tries to solve. We conclude with an annotated bibliography that captures the most recent relevant research in the field of Computer engineering.

2.2. Drowsy Driver Detection

Using machine learning/deep learning algorithms, there are numerous studies on drowsy driving detection systems. The two types of monitoring systems that make up the real-time detection system are those that are the driver- and vehicle-oriented. Using information gathered by the car's sensors, such as the location of the vehicle in the lane, steering wheel motions, pedal pressure, and other factors, the car-oriented device analyzes the driver's habits to determine whether the driver is sleepy or not. The driver-oriented system uses facial traits to detect sleepiness. The human face is animated and capable of displaying a variety of emotions. Being one of the most noticeable features of the human face, human eyes are crucial for object recognition and facial expression studies. It is helpful to be able to identify eyes when identifying facial features. Recognizing eyes is helpful when identifying features of a face. The data used in the articles is not the same for every article. For example, in the paper in [7], the authors used facial expressions to develop the model. In the article in [9], eye ratio and facial expressions are used as input. In the article in [16], since the relationship between steering and the level of sleepiness is tried to be examined, steering data is used as input. In the article in [8], the authors tried to find and predict the distraction level and drowsiness level using smart glasses. As we mentioned above, although human eyes, faces, and similarities are of critical importance, different data are used to develop models.

We are considering using computer vision algorithms used for face recognition. In addition, using artificial neural networks (CNN), we will ensure that the project includes deep learning algorithms and learn the status of the driver from the data itself. We will check the accuracy of this learning with different metrics and try to maximize it. In addition, we will make it permanent through an android or web application by using the models derived because of this. Thus, the system will become usable for a long time. Although I said only CNN above, different methods are used in different articles that we will see in the future. To give a short example of these, RNN and RF algorithms are used in the article [11]. The articles in [9] and [10] used CNN, the method we are talking. In the article in [13], unlike the others, ANN combined with DFT was used. The articles in [18] used artificial neural networks (ANN), support vector machines (SVM), and adaptive boosting to improve accuracy of model. As a final example, the article in [7] can be given. QDNN (Quantum Dot Neural Network) algorithm was used to achieve the desired results.

In short, different algorithms can be used according to the purpose of the article. Figure-1 can be shown in the below that you can examine results of articles.

Article	Accuracy	Algorithms
Real time Driver Detection [1]	98.81%	M-ConNN
Automatic Driver Detection [13]	72.25%	ANN–EAR–MAR
Driven Facial Emotion Recognition [7]	99.31%	QDNN
Real-time Driver Drowsiness [2]	95.2% - 93.5%	CNN-MTCNN
Driver Drowsiness Detection [9]	83%	CNN
Eye Detection for a Real-time [18]	-	ANN–SVM–AdaB
Driver Safety Detection [3]	99%	FD–NN–TL-VGG
Drowsiness Detection Techniques [4]	-	VSM–HMM–SVM

Figure-1 Algorithms are used in articles

2.2.1. Challenges of Drowsy Driver Detection System

Some problems may occur during the implementation phase of this project. We need to anticipate them. Some of the problems that may occur in the project are:

- Errors that may occur when using the pre-defined dataset
- Difficulty finding training data
- Errors that may occur while performing Face Recognition

If these problems occur, they will be tried to be overcome with different engineering methods and the project will be successful.

2.2.1.1. Finding Appropriate Dataset and Pre-defined Dataset Problems

This problem has been mentioned in most articles used in the literature. Every dataset

published so far has issues or deficiencies, albeit different from each other. For instance, no videos were taken in the dark in one, or a data set was not created for individuals using glasses in the other. Due to these problems, when the system encounters such data, the accuracy value will decrease, and its reliability will decrease. To avoid such issues, both the datasets in the literature are used, and a dataset created by us is intended to be used in our project. Thus, we will bring a new dataset to the literature and try to increase the evaluation values of the project.

2.2.1.2. Face Recognition Problem

While using the obtained datasets, the videos will be framed by different libraries used for computer vision such as OpenCV. It is aimed to make face detection on the obtained frames. However, the fact that the person in the photo has glasses or the photos are taken in dark environments can easily prevent detection. To prevent these, different algorithms or different formulas that can be found in the literature that can detect eye and mouth deficits can be used. These formulas will be applied to the system by using the facial landmarks to be obtained and calculations will be made.

2.2.2. Field of This Project and Algorithms that are used

The fields of this project are mainly Software and its engineering and Artificial Intelligence. According to the methods to be used in its content, computer vision, Deep Neural Networks (CNN, LSTM, etc.) can be added. This list of fields can be further developed with different algorithms to be added or added. In addition, this project will be developed in python language and basic python libraries such as Scikit-learn, Keras, Tensorflow, PyTorch, and YOLO will be used. Some of the methods and libraries mentioned here need to be explained and detailed. In another part, we will examine different methods and libraries.

2.2.2.1. CNN

For picture identification, CNN, or the Convolutional Neural Network object detection technique, is utilized. In this approach, we choose distinct areas from various portions of a picture and attempt to categorize the presence of an item in that region. The CNN algorithm, on the other hand, can only select a restricted number of areas.

2.2.2.2. YOLO

The YOLO (You Only Look Once) algorithm is quite different from prior approaches. It has been taught to do bounding box prediction and classification simultaneously. As a result, the YOLO algorithm is quicker than the CNN method. However, YOLO has similarities with earlier algorithms. It also has certain disadvantages. Because each grid may only have two bounding boxes, YOLO may fail if items are too close together. Furthermore, because to spatial constraints, it has difficulty identifying tiny objects.

2.2.2.3. Sckit-learn, Keras and Tensorflow

Scikit-learn is a popular Python tool for data research and machine learning. It supports a wide range of operations and offers a variety of algorithms. Scikit-learn also includes documentation for its classes, methods, and functions, as well as the algorithms that are employed.

TensorFlow is a deep learning math library that is open source. TensorFlow, which is part of the machine learning cluster, also supports classical machine learning methods and was created largely for internal Google usage by the Google Brain team. TensorFlow is defined as a deep learning (Deep learning) library with open-source code. With a single API (Application Programming Interface), TensorFlow's flexible structure allows calculations on all systems.

Keras is a deep learning package that works atop the previously stated TensorFlow. It is a Python package that does not define nearly any deep learning model. It is a high-level neural network API that can be used with not just TensorFlow but also Theano and CNTK.

2.2.2.4. RNN- LSTM

The results of people's experiences and activities provide some opportunity for learning. To move on, these inferences are stored in memory and reused as necessary. In neural networks, fresh outputs are produced and stored in memory while some inputs are used repeatedly as needed. Traditional neural networks lack a system for remembering previously learned information. Recurrent neural networks (RNNs) are intended to analyze input in time and in sequence and to transfer information in various steps. A neural network can analyze data that is presented as a time series and has a relationship between the individual time steps.

A typical step inside the RNN is differentiated by the LSTM. This difference is that new

parameters are created and stored via a variety of mathematical processes, as opposed to a conventional basic neural network construction. So, when necessary, it is possible to obtain the input or output data from the preceding steps. By transmitting the data from the previous layer to the necessary ports and performing mathematical operations, it simultaneously stores an output value and delivers it as an input to the next layer.

2.3. Main Findings

While scanning articles, most of the articles prepared on the drowsy driver detection system focused on 3 basic features. There are 3 basic measurement techniques used to develop the Drowsy driver detection system. These are Image measure, biological measure, and vehicle measure. First, let's examine the image measure techniques, which we will also use.

2.3.1. Image Measure on Drowsy Driver Detection

It is aimed to conclude by focusing on the face and eye features through photography. They try to examine the driver's drowsiness with different algorithms (like CNN-LSTM). In addition, factors such as blinking and yawning are also used, apart from the state of the plain and face and the state of the eyes.

2.3.2. Biological Measure on Drowsy Driver Detection

Psychological-Biological scales that can be obtained by methods such as EEG, and the other is physical scales that focus on facial, mouth, and body movements. Some models give very good results in both. These sibyls are one of the methods used to further strengthen the system for drowsy driver detection systems. Examples of these signals are factors such as brain activity, heart rate, breathing rate, and body temperature. It can produce more accurate results than those used in the Image method. Examples of these methods are EEG, ECG, and PPG. In short, to be clear, EEG examines and categorizes brain signals. ECG examines the electrical activity on the skin. PPG, on the other hand, focuses on factors such as cam pressure and oxygen content. Again, algorithms such as LSTM can be used with these data.

2.3.3. Vehicle Based Measure on Drowsy Driver Detection

These measurements focus on the analysis and study of driving patterns. That is, situations

such as tire movements are examined. But it is not very preferred to be used alone. It is generally used in conjunction with other measurement methods. Examples of these measurements are the steering wheel and the lateral distance.

As we can understand from the articles in the literature, the main problem in this project is the diversity of the datasets. Some datasets only contain images in bright environments, while others have recordings in dark environments. Some of them also have different labels such as using and not using glasses. Therefore, the datasets used also significantly affect the results of the article. In short, data set selection is of great importance. Generally, NTHU Drowsy Driver Detection (NTHU-DDD) Dataset has been used in the literature. For example, this dataset is used in the article in [5]. Also, in the article in [1], YawdDD and NTHU are used together, because those datasets are a large content dataset. In [15], [16] articles, different datasets are used for different purposes. After a general explanation, we move on to the part where we will examine in detail some of the articles that we consider very important and selected from the general literature review. Figure-1 can be shown in the below that you can examine.

Article	**Dataset is used**
Real time Driver Detection [1]	YawdDD and NTHU
Automatic Driver Detection [13]	Their own datasets
Driven Facial Emotion Recognition [7]	FER-2013 and CK+
Driver Drowsiness Detection [5]	NTHU
Driver Safety Detection [3]	Mixture dataset of ZJU

Figure-2 Dataset is used in articles

2.4. Top Papers

Although each article has valuable information, we selected some of the "top papers" from our literature review. Our aim is to collect the existing data in a general framework. In each article, different problems were encountered, and different solutions were sought. Examination of these guides us in our own project. Therefore, it is important to re-examine each article step by

step. But in order not to waste time describing the whole literature, we will examine a few in detail to provide general information.

Our first article was written by Wanghua Dengi and Ruoxue Wu [2], and the DriCare system, which can detect fatigue status states such as yawning and blinking from video recordings, is proposed. In addition, unlike the algorithms used in other literature, a new face-tracking algorithm has been developed to improve accuracy. This model has 68 basis points. They argue that by using these points, the driver's status can be easily determined. They say that an accuracy of 92% is obtained from the model developed using features obtained from the mouth and eye. The model basically detects eye closure, blinking, and yawning. In addition, the Multiple Convolutional Neural Networks (CNN)-KCF (MC-KCF) algorithm was used to obtain better results in low-light situations. Multitask convolutional neural networks (MTCNN) are used to detect some situations that the KCF algorithm cannot detect. Another method used is CNN. They used CNN to detect the state of the eyes. In short, human face tracking is done with the help of MC-KCF from the data obtained from the video stream. Then, feature extraction is performed, including mouth and eyes. As the last stage, there is the evaluation part.

In the article [3], they tried to make a real-time application using CNN. Accuracy has been tried to be increased and the algorithm has been tried to be accelerated as much as possible. Also, a Fully Designed Neural Network (FD-NN), Transfer Learning in VGG16 and VGG19 with extra designed layers (TL-VGG) were used to make eye status classification. Considering the scarcity of the dataset on eye closure detection in the literature, a different dataset (mixture dataset of ZJU) is presented in the article. It has been used as 3 category drowsiness detectors as Vehicle-based, Signal-based and Facial feature-based. In the decision-making phase, unlike other articles, they adjusted the system according to the actual psychological values. For example, for a healthy person, blinking is between 100-400 ms. If this number increases to 1 second, it is perceived by the system as a symptom of fatigue. In addition, if the person is detected blindfolded in 12 consecutive photographs in the data set, the warning system works. Finally, a very high value for accuracy, such as 99%, is achieved.

In another article [4], the methods found in the literature are divided into 3 categories: Behavioral, Vehicular, and Physiological parameters-based techniques. Apart from these, the most used supervised learning algorithms in the literature were tried to detect drowsiness.

PERCLOS was used for eye-based drowsiness detection. In the article, SVM is used for mouth and yawning classification. The main purpose of using SVM is to reduce the cost. For different situations such as poor lighting, the driver wearing classes, and the Driver with a mustache, experiments were made according to each categories mentioned above. The most appropriate category for each situation was found, providing a basis for further research. Although mainly focused on SVM, CNN and HMM are also used. The differences between them are revealed, so HMM shows a less error rate, but both CNN and HMM are slow in training and expensive as compared to the SVM classifiers. The main purpose of this article is to get results by trying different algorithms and categories for different situations. These results were considered by the authors to form a basis for further research.

In the article [5], deep architecture called deep drowsiness detection is proposed. This structure includes 3 deep networks to obtain environmental variations and learn local facial movements and head gestures. The results of these 3 networks are combined with softmax. An accuracy of around 73.06% was achieved. The networks used are AlexNet, VGG-FaceNet, and FlowImageNet. It was used to collect features related to AlexNet drowsiness. VGG-FaceNet has been used to obtain facial features for different situations such as ethnicities, and hairstyles. FlowImageNet was used to find out whether facial and head movements from sequential images were related to drowsiness. Every three networks are independently fine-tuned for multi-class drowsiness classification. In addition, 2 different ensemble strategies were used. One of them is independently averaged architecture (IAA), and the other is feature-fused architecture (FFA). The NTH Drowsy Driver Detection dataset is used in the article.

In another article [1], studies were carried out on 2 different datasets. These are the YawdDD and NthuDDD datasets. Unlike other articles, it is recommended by the authors to use Multi-tasking Convolutional Neural Network (ConNN*) to find drowsiness in this article. As a feature of this algorithm, both eye and mouth information can be categorized at the same time. This feature provides the model with both great speed and storage space. Calculations of the duration and percentage of closed eyes (PERCLOS) as well as the frequency and duration of mouth and yawning sneezes are used to assess driver fatigue (FOM). As a result of this study, a very high value of 98.81% was obtained. The proposed system can simulate how the interaction between the mouth, eyes, and face.

This article [11] explains the technology developed to detect drowsy behavior based on the

result that drowsy driver exhibits more drowsy behavior. Since it is difficult to measure physiological signals in the developed behavior detection technology, it is preferred to use driving information. Driving information was obtained from experiments using simulators. The input parameters used consist of lateral acceleration, longitudinal acceleration, steering angles. These parameters were investigated in different time periods. ANN and RF were used as algorithms. It turned out that the RF algorithm is better at examining data sets than the ANN algorithm. The studies conducted in the article provided a high accuracy of 84.8% and it was concluded that it can be used in real vehicles.

In the article [17], These characteristics are included in the new sleepy driving prevention system that is proposed in the current development. The suggested system, known as a wakefulness-keeping support system (WKSS), is made up of an active gaming system (AGS), which corresponds to the second and third characteristics, and a drowsiness detection system (DDS), which corresponds to the first feature. These characteristics are included in the new sleepy driving prevention system that is proposed in the current development. The suggested system, known as a wakefulness-keeping support system (WKSS), is made up of an active gaming system (AGS), which corresponds to the second and third characteristics, and a drowsiness detection system (DDS), which corresponds to the first feature.

According to the research [12], 23.5% of traffic accidents were caused by the driver being sleepy. This article focuses on two models to find a solution to this problem. Two models are proposed in this research. The first model can determine whether a driver is sleepy. The first model is successful in detecting whether the driver is drowsy, while the second Model is designed to detect if the driver yawns, blinks or speaks. A total of 12 different situations were detected and defined by the second model. Indicators showing that the driver is sleepy are defined using multitask learning. The accuracy of the model was 92.40%. The second model is a multi-class model that can recognize the driver's condition, such as whether they are chatting, blinking, or yawning. Driver drowsiness signs are discovered through multitasking learning.

In the last article [6], a different parameter was used compared to other articles. Gold standard brain bio-physiological signal and facial expression data were used to detect early drowsiness. Data were obtained from a group of 10 people. ANN was used to process them and reveal a model. Different features such as EAR, MAR, and FL are used. (ie, eye aspect ratio (EAR),

mouth aspect ratio (MAR), face length (FL)) It has been suggested that the model created with ANN together with these features gives the best results.

These articles are part of our overall literature review. You can find many more detailed articles in the References section.

2.5. Conclusion

Drowsy Driver Detection System has recently become the most important vehicle part used by vehicle companies. Because the number of people who lost their lives in traffic is also increasing. Therefore, R&D resources transferred to these systems are increasing day by day and research is increasing. Therefore, every effort and time given to this project is very valuable to us. As we can see from our literature review, ANN and its derivative algorithms are generally used in this field. Because they both give very good results and are less expensive than other algorithms. In addition, facial expressions are mainly used for drowsiness detection, although there are approaches in more than one category. We will focus on this in our project. Another issue is the dataset problem. Although there is more than one dataset in the market, we have found the most suitable one thanks to this research. With different evaluation results, the best dataset became easily selectable. Finally, this literature review we have done has been written to form a general idea and to form a basis for the project to be done.

2.6. References

[1] B. K. Savaş and Y. Becerikli, "Real Time Driver Fatigue Detection System Based on Multi-Task ConNN," in *IEEE Access*, vol. 8, pp. 12491-12498, 2020

[2] Deng, W., & Wu, R. (2019). Real-time driver-drowsiness detection system using facial features. Ieee Access, 7, 118727-118738.

[3] Hashemi, M., Mirrashid, A., & Beheshti Shirazi, A. (2020). Driver safety development: Real-time driver drowsiness detection system based on convolutional neural network. SN Computer Science, 1(5), 1-10.

[4] Ramzan, M., Khan, H. U., Awan, S. M., Ismail, A., Ilyas, M., & Mahmood, A. (2019). A survey on state-of-the-art drowsiness detection techniques. IEEE Access, 7, 61904-61919.

[5] Park, S., Pan, F., Kang, S., & Yoo, C. D. (2016, November). Driver drowsiness detection system based on feature representation learning using various deep networks. In Asian Conference on Computer Vision (pp. 154-164). Springer, Cham

[6] Inkeaw, Xu, L., Li, S., Bian, K., Zhao, T., & Yan, W. (2014, February). Sober-Drive: A smartphone-assisted drowsy driving detection system. In 2014 International conference on computing, networking, and communications (ICNC) (pp. 398-402). IEEE.

[7] A. Mustafa Hilal, A., Elkamchouchi, D. H., Alotaibi, S. S., Maray, M., Othman, M., Abdelmageed, A. A., ... & Eldesouki, M. I. (2022). Manta Ray Foraging Optimization with Transfer Learning Driven Facial Emotion Recognition. Sustainability, 14(21), 14308.

[8] -J. Chang, L. -B. Chen and Y. -Z. Chiou, "Design and Implementation of a Drowsiness-Fatigue-Detection System Based on Wearable Smart Glasses to Increase Road Safety," in *IEEE Transactions on Consumer Electronics*, vol. 64, no. 4, pp. 461-469, Nov. 2018, doi: 10.1109/TCE.2018.2872162.

[9] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen and K. Barkaoui, "Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application," *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 237-242, doi: 10.1109/ICIoT48696.2020.9089484.

[10] BenJi, Y., Wang, S., Zhao, Y., Wei, J., & Lu, Y. (2019). Fatigue state detection based on multi-index fusion and state recognition network. IEEE Access, 7, 64136-64147.

[11] Wang, M. S., Jeong, N. T., Kim, K. S., Choi, S. B., Yang, S. M., You, S. H., ... & Suh, M. W. (2016). Drowsy behavior detection based on driving information. International journal of automotive technology, 17(1), 165-173.

[12] Seetharaman, R., Sridhar, S., & Mootha, S. (2020, October). Detection and State

Analysis of Drowsiness using Multitask Learning with Neural Networks. In 2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS) (pp. 1-8). IEEE.

[13] Inkeaw, P., Srikummoon, P., Chaijaruwanich, J., Traisathit, P., Awiphan, S., Inchai, J., ... & Theerakittikul, T. (2022). Automatic driver drowsiness detection using artificial neural network based on visual facial descriptors: pilot study. Nature and science of sleep, 14, 1641-1649.

[14] Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting driver drowsiness based on sensors: a review. Sensors, 12(12), 16937-16953.

[15] Dwivedi, K., Biswaranjan, K., & Sethi, A. (2014, February). Drowsy driver detection using representation learning. In 2014 IEEE international advance computing conference (IACC) (pp. 995-999). IEEE.

[16] Eskandarian, A., & Mortazavi, A. (2007, June). Evaluation of a smart algorithm for commercial vehicle driver drowsiness detection. In 2007 IEEE intelligent vehicles symposium (pp. 553-559). IEEE.

[17] Ibe, T., Fujiwara, K., Hiraoka, T., Abe, E., & Yamakawa, T. (2020). Development of Game-Like System Using Active Behavior Input for Wakefulness-Keeping Support in Driving. IEEE Transactions on Intelligent Vehicles, 6(2), 323-332.

[18] Coetzer, R. C., & Hancke, G. P. (2011, June). Eye detection for a real-time vehicle driver fatigue monitoring system. In 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 66-71). IEEE.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1. Introduction

Driver drowsiness detection means that technology to helps avoid 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 continues 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.

3.1.1. Purpose of This Document

In this project, our goal is to develop a system that detects the drowsiness of drivers from momentary videos of them. The main purpose of this project designs a warning system for drivers to decrease the accidents that occur from drowsiness. This project is not a solution for drowsiness, it is just a warning and precautionary 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, and Machine Learning. This SRS document contains the project requirements and Software Requirements Specification for drowsiness detection.

3.1.2. Scope of the Project

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 an 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

3.1.3. 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.

3.1.3.1. 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.

3.1.3.2. Glossary (Definitions, Acronyms, and Abbreviations)

Term	**Definition**
Actor	An actor can be a user like driver. It can be also another software system that interacts with the system.
Python	A document that contains a detailed explanation of the system's operations, needs, restrictions, and operating conditions. The following document is an SRS document.
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.

3.1.3.3. 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 ofHead	Detect the position of the head of driver
Performs Alerts	Give a alarm when detect drowsiness
Detect Drowsines s	Calculate the data with given algorithms and detect drowsiness

3.1.3.4. Product Functions

This section will explain the basic functions that will be used in the project. These functions will be explained in more detail later in the srs document.

- **Detection of Face, Eye and Mouth:** In this part, the face frames of the driver will be obtained thanks to libraries such as OpenCV, dlib. Thanks to these frames, the system will perform the other requirements by applying classification. It is one of the most important parts of the project.
- **Capture Real-time Video**: Thanks to the camera system, the driver's video, which will contain all her facial expressions, will be automatically uploaded to the system, and the processes will continue through these photos.
- **Alert System**: According to the result of the classification system, if drowsiness is detected in the driver, the driver will be warned with a non-irritating signal so that the user can focus on the road and regain herself.
- **Classification of Driver's Status:** The pre-trained model will be applied on the available

frames. As a result of this application, it will be discovered whether the user is drowsy, and after these actions, a warning will be sent to the alert system so that he can act.

3.1.3.5. User Classes and Characteristics

3.1.3.5.1. 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. The tasks of driver are given below:

- Driving a car
- Starting the application

3.1.3.5.2. System

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

- Face recognition system
- Implementation of the trained model
- Making Classification (Drowsiness detection)
- Taking a 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

3.1.3.6. Constraints

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.

3.1.3.7. Dependency and Assumptions

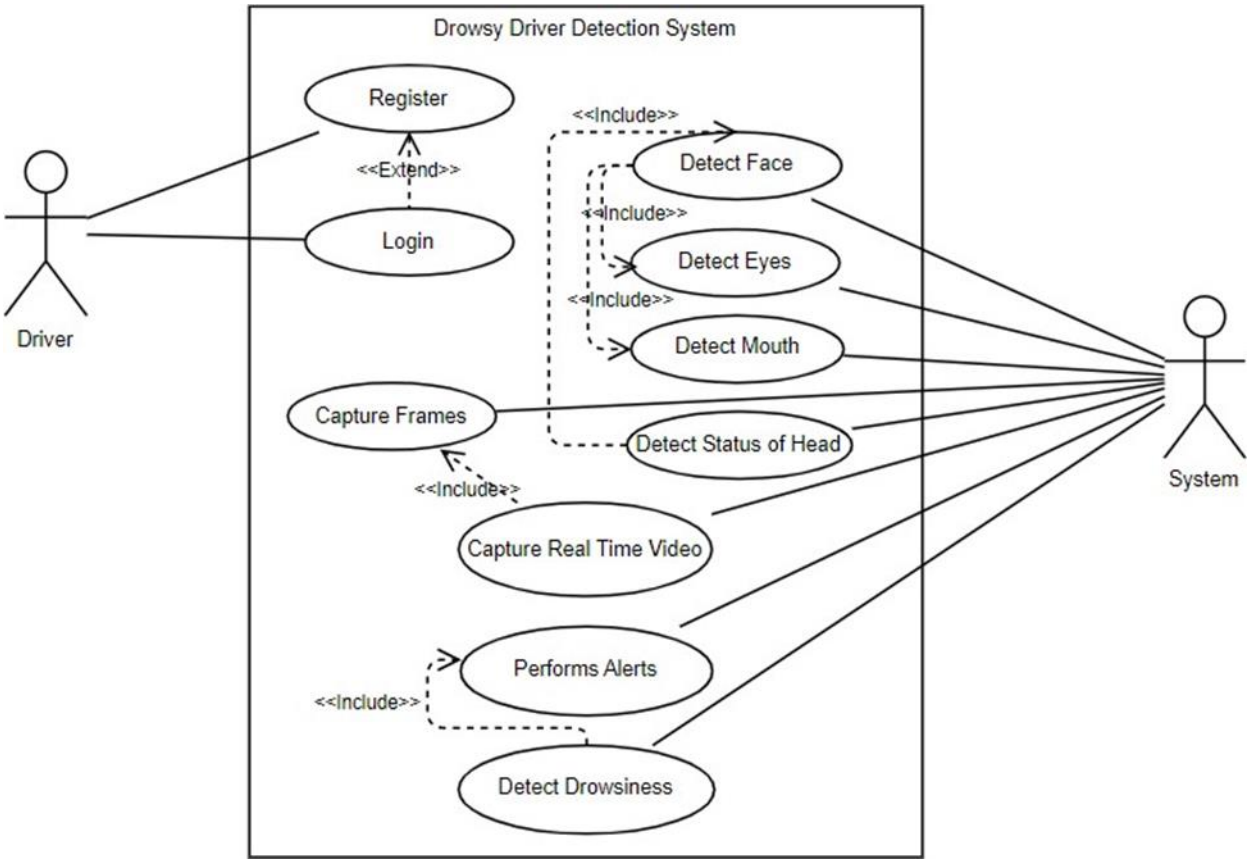
We will assume that this system is used under some the following assumptions and dependencies.

- 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.

3.2. Requirement Specification

3.2.1. Detailed Description of Functional Requirements

3.2.1.1. Use Case Diagram



3.2.1.2. Use Cases

3.2.1.2.1. 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. >

3.2.1.2.2. 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. >

3.2.2. Non-Functional Requirements

3.2.2.1. 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.

3.2.2.2. 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.

3.2.2.3. 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.

3.2.2.4. 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.

3.3. References

- 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

4. SOFTWARE DESIGN DESCRIPTION

4.1. Introduction

This Software Design Description Report provides understandable information on what kind of system software should include in our project. This document includes the design of the system in the project we will develop, and a general description of the methods used. In this project, we will use software techniques while developing the system. For this software part, we will take help from python language. In general, our system includes a camera system, detection system, classification system, and alert system. We will take help from the OpenCV library and the Dlib library for the Camera and Detection system. OpenCV will be used for both object detection and face detection. The dlib library will act as an auxiliary library for face detection. For the Classification system, it is possible to use the CNN-Inception v3 algorithm included in the TensorFlow library. It can be replaced with a better algorithm in future development stages. For the alert system, help will be taken from the libraries in python. The Flask framework will be used for a web application. It is considered to use this framework as it will be easier to integrate with Python. These techniques will be explained in detail in later sections.

4.1.1. Purpose of This Document

This document outlines what needs to be done and how the system can be built. As a result, the paper includes all the interfaces, diagrams, and interactions that we used during the project's development.

4.1.2. Scope of the Project

Our project scope is to create an application that analyzes the user's behavior and status to identify their current level of sleep. The use of computer vision will not be the only factor considered in making this decision. Naturally, instantaneous face, mouth, and eye detection will also be possible, but the goal is to apply machine learning for eye detection. The number of yawning will be immediately counted in mouth detection utilizing face points gathered from various libraries, such as dlib, and a warning will be given to the user after exceeding a specified threshold. The user's status cannot be determined by keeping their eyes closed for an extended period. In this area, numerous initiatives are underway. Instead of just focusing on eye issues, we want to create a system that works with the car. Additionally, it aims to enhance sleepiness detection considering various environmental factors. Our system will mostly consist of:

- Face Detection for checking drowsiness
- Eye, Mouth Detection for checking yawning and drowsiness
- Classification with Machine Learning Model
- Alert System for avoiding drowsiness

4.1.3. Glossary (Definitions, Acronyms, and Abbreviations)

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.

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.
Inception v3	It is used for classification of images and object detection. It is thought of as a database of classified objects.
SDD	A document that offers a detailed description of the system's design, requirements, and conditions for method execution.
Flask	Flask is used for making web application to integrate with backend. Also, Flask is a micro web framework which is written in Python.
Jupyter Notebook	Jupyter Notebook is an opensource development platform for Python.
Object Detection	Object detection is used for applying computer vision algorithms to perform detection of object to image.
OpenCV	This is a library that is used to get video streaming and provides Computer Vision library.
Scikit-learn	This is a library that is used for applying machine learning algorithm. It includes algorithms like SVM, clustering etc.
Tensorflow	This is a library that is used for applying machine learning algorithms. It includes algorithms like ANN etc.
Pandas	This is a pyhton library for offering different operations for manipulating data.

4.1.4. Overview of This Document

We attempted to present a detailed design description of Drowsy Driver Detection System in this SDD. As a result, we expressed our system's design assertions. In the first section, we present a brief overview of our system and discuss its essential characteristics. We classified the first part as 1.1. Purpose and Scope. Glossary 1.2, Stakeholders and Their Concerns 1.3. In the 1.1. Purpose of this Document section, we indicated what this document seeks to convey, as well as the scope of this project. All special terminology used in this report are defined in 1.3. Glossary. Also, in the 2. Reference part, we indicated references that we used while creating this document.

Following that, in the 4. System Design part, we revealed our system's full design architecture. It is divided into three sections: 4.1. Architectural Design and 4.2. User Interface Design, 4.3. Requirement Matrix. In 4.1. Architectural Design, we presented many schematics of our system and explained the context of the design idea behind the system. The User Interface Design section in 4.2 outlines the overall structure of our system. These designs are presented in the form of figures with brief descriptions. We supplied a table structure in the 5. Requirement Matrix section that demonstrates the link between components and requirements contained in the SRS document. We created a link between system components and requirements for this purpose by describing which requirement is achieved by which component. We verified that each need is linked to the required component. As a result, this paper will serve as a guideline for drivers to comprehend the Drowsy Driver Detection System's overall design architecture.

4.2. System Design

This part of system design includes the Architectural Design of the system, problem description, and it includes which technologies will be used in our project, User Interface Design, and Hardware Design. Also, it contains some diagrams such as Sequence Diagram, Activity Diagram, Data Flow Diagram, Class Diagram.

4.2.1. Architectural Design

To offer a clearer understanding of our architectural design, we explained the challenge, technologies employed, and many diagrams in this part. As we discussed in earlier parts, our system will perform three major functions. As a result, we will create an architecture that includes these duties. The primary construction of the Drowsy Driver Detection System is shown

in Figure 1.

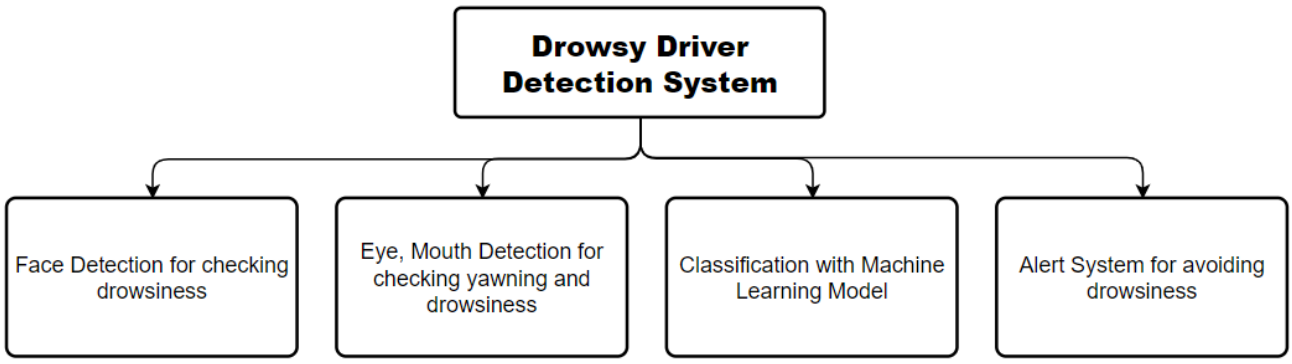


Figure 1: Brief of System Architectural Design

4.2.1.1. Problem Description

It is a project that has the capacity to provide a certain amount of improvement to car accidents, which are a problem that cannot be ignored around the world. We will try to find solutions to some problems in this project, which we set out to obtain a better result than other projects in the literature. Our system, which can provide classification under different conditions, will try to keep the driver away from situations that may endanger the driver's life with its alarm system. Our project, which also focuses on different features, is expected to provide solutions to technical problems such as data set problems. This project, integrated with machine learning, can work more autonomously than other detection systems. From the moment the driver starts the application, our project will do its best to ensure the driver's safety.

4.2.1.2. Technologies Used

There are all the software tools we will use in our system below:

- Python
  - We use Python for many operations in our system such as Image Processing, Object Detection and Machine Learning.
- Jupyter Notebook
  - Jupyter Notebook is an opensource development platform for Python.
- Image Processing
  - In field of computer engineering, image processing is used for applying computer algorithms to perform image processing on images.
- Object Detection
  - In field of computer vision, object detection is used for applying computer vision algorithms to perform detection of object to image.
- Machine Learning
  - We will use Machine Learning Techniques for classification of drowsiness and detection of closed, open eyes in our system. We will use CNN-Inception v3 algorithms. It is best option for classification of images and object detection. It is thought of as a database of classified objects.
- Web Application
  - We will use Flask for making web application in our system to integrate with backend of our system. Flask is a micro web framework which is written in Python.

4.2.1.2.1. Software Requirements Specification

- Python
  - Python 3 or higher version
  - Flask: Flask is a micro web framework which is written in Python. It is used for creating web applications.

- Libraries
  - NumPy: This library is used for working with arrays.
  - Dlib: This library is used for finding human face and pose of face via facial

landmarks.

- OpenCV: This library is used to get video streaming via Camera.
- Scikit-learn: This library is used for applying machine learning algorithm. (SVM, Clustering etc.)
- TensorFlow: This library is used for applying machine learning algorithms. (ANN etc.)
- Pandas: This library offers different operations for manipulating data.
- Matplotlib: This library is used for visualization, etc.

- Operating System
  - Windows or Linux

4.2.1.2.2. Hardware Requirements Specification

- Laptop with basic hardware
- Camera

4.2.1.3. Data Flow Diagram

In figure 2,3,4,5,6 of this section, data flow diagrams are stated.

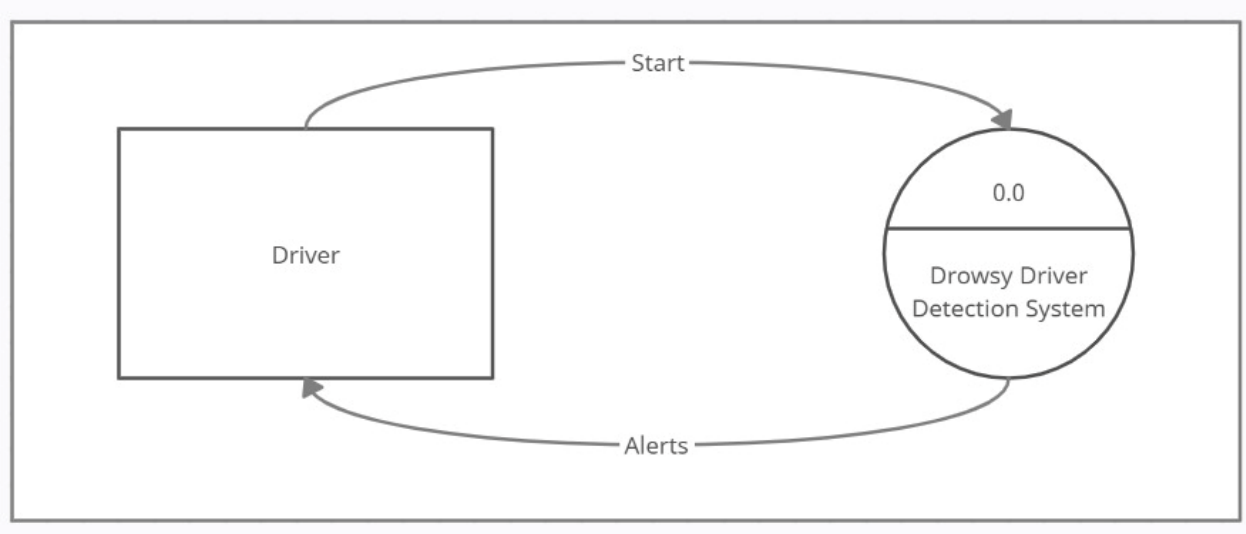


Figure 2: Context Diagram

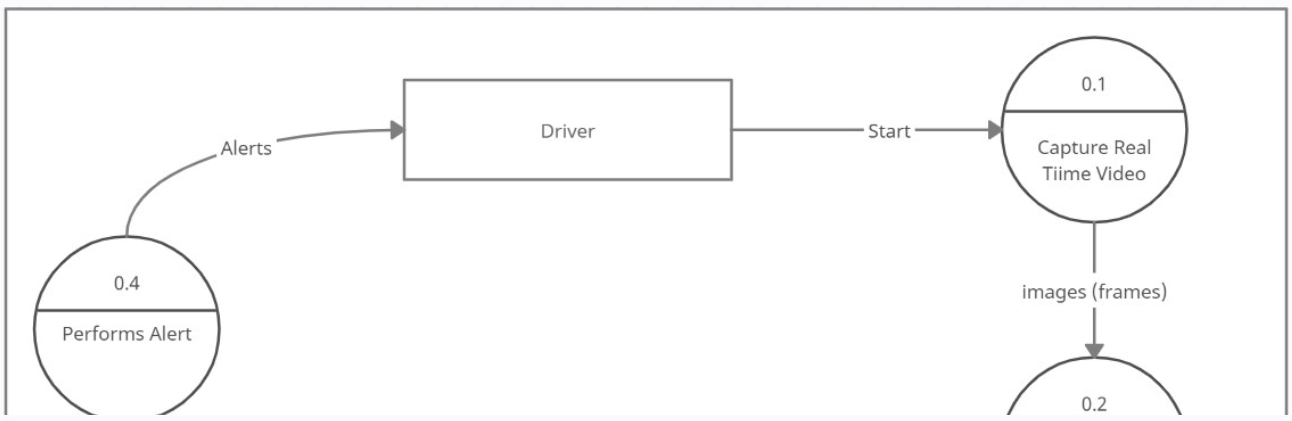


Figure 3: DFD Level -1

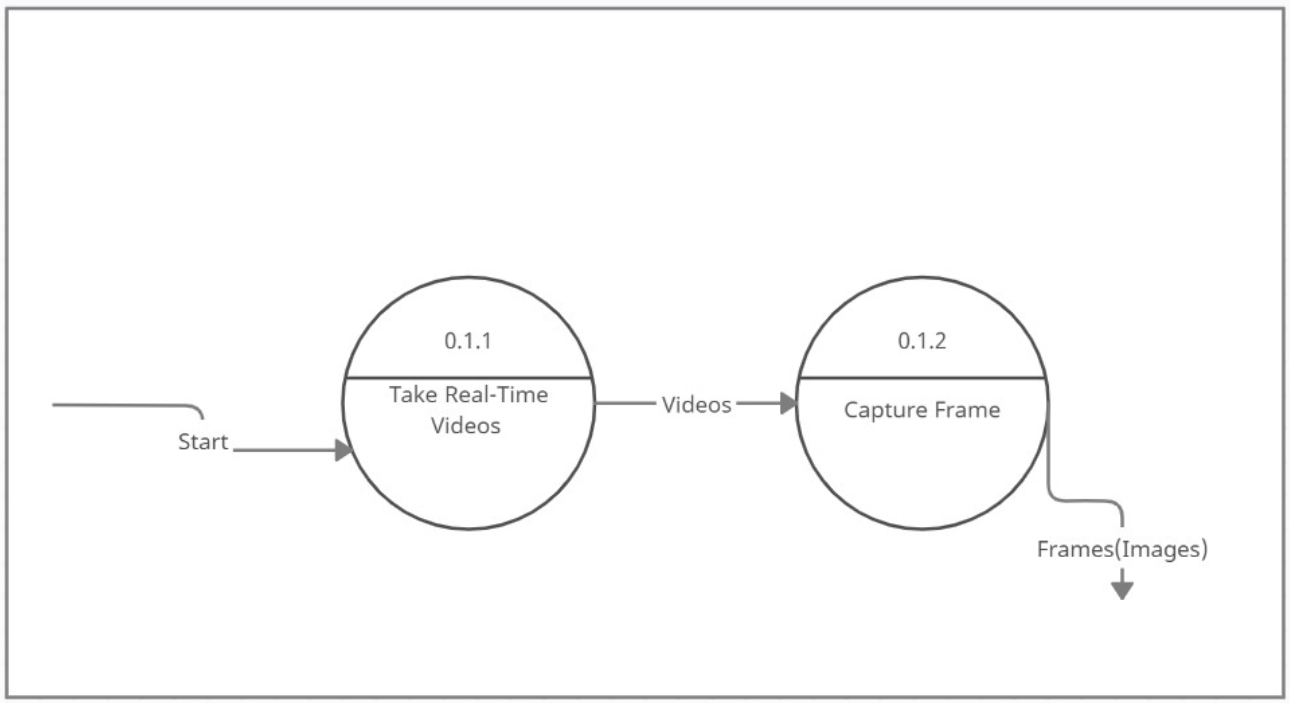


Figure 4: DFD Level-2 for Capture Real-time Video

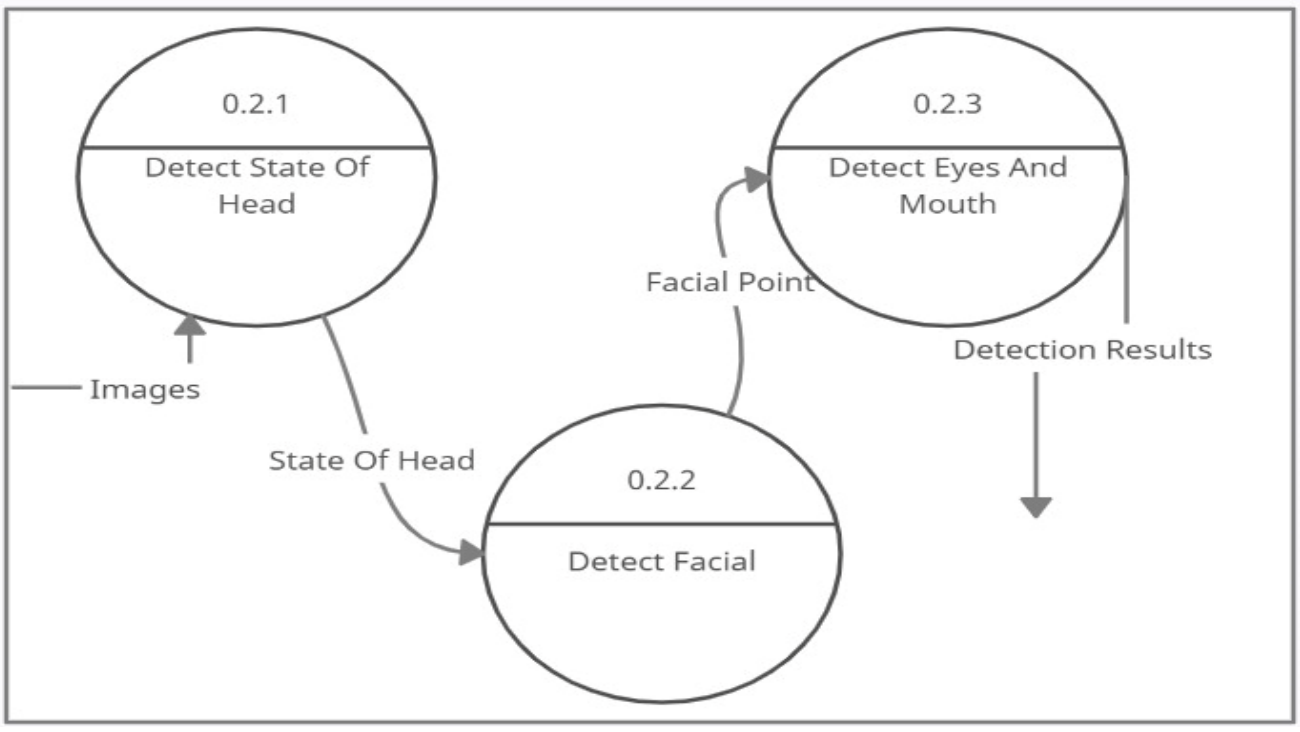


Figure 5: DFD Level-2 for Detect Face, Eyes, and Mouth

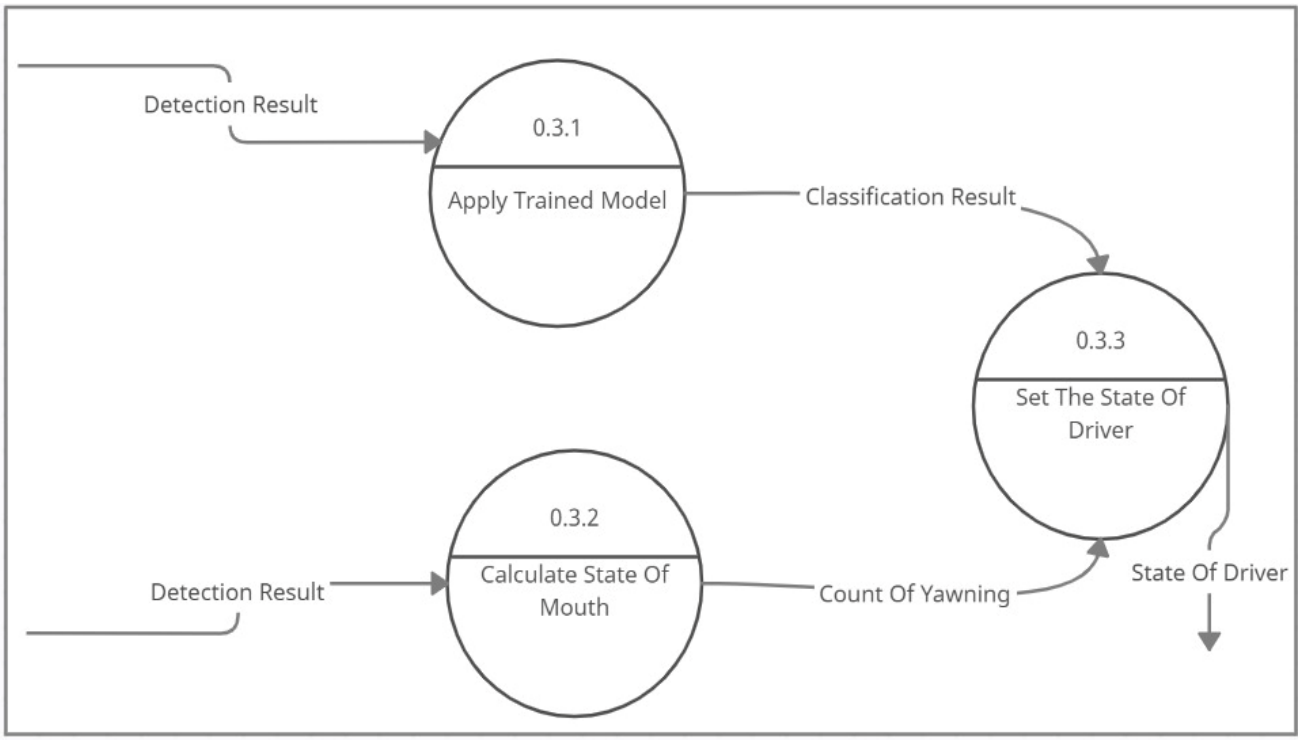


Figure 6: DFD Level-2 for Detect Drowsiness

4.2.1.4. Activity Diagram

In figure 7, activity diagram is shown.

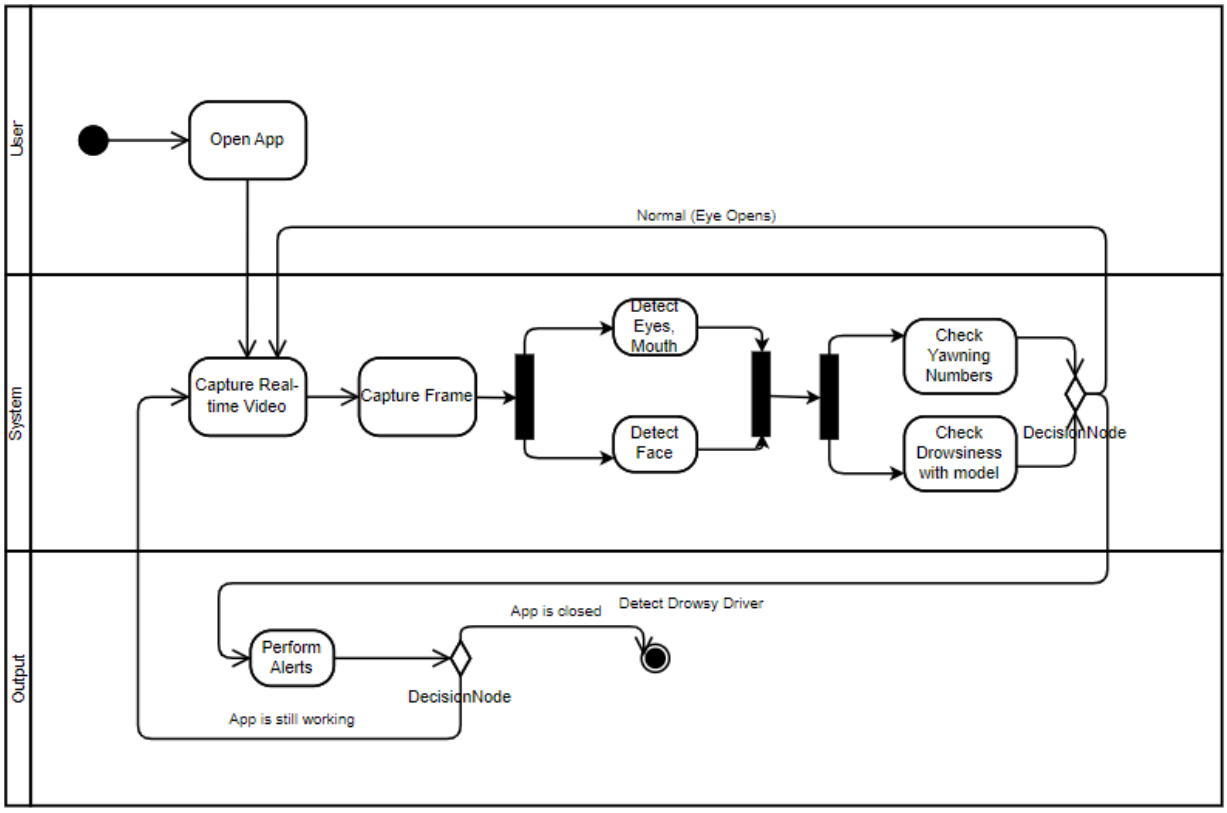


Figure 7: Activity Diagram for Drowsy Driver Detection System

4.2.1.5. Class Diagram

In figure 8, class diagram is shown.

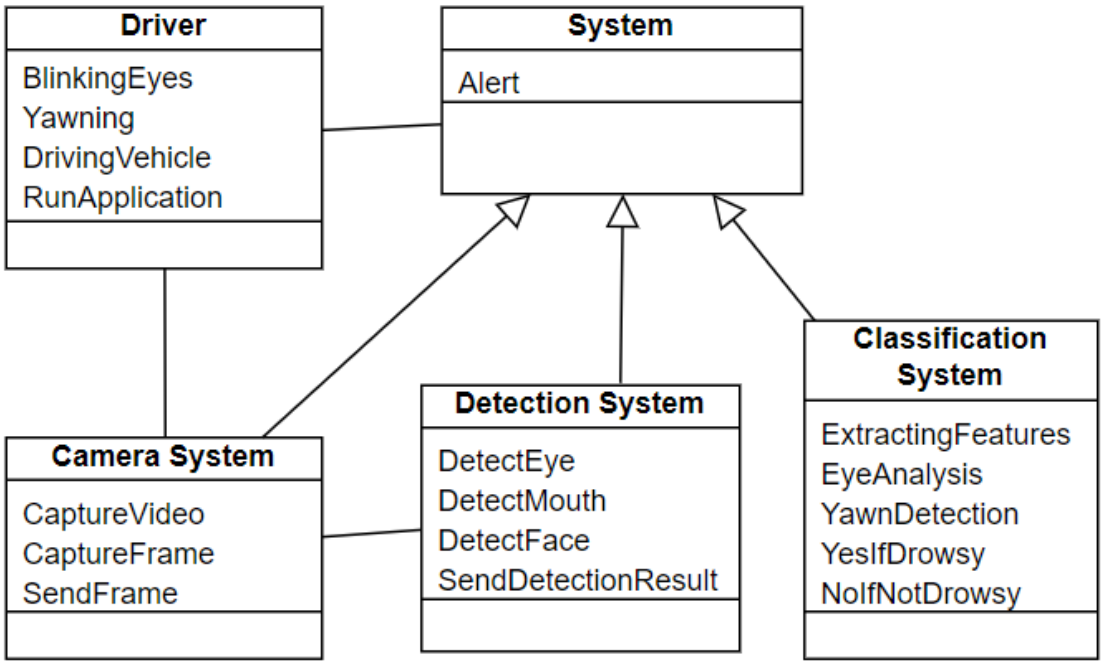


Figure 8: Class Diagram for Drowsy Driver Detection System

4.2.1.6. Sequence Diagram

In figure 9, activity diagram is stated.

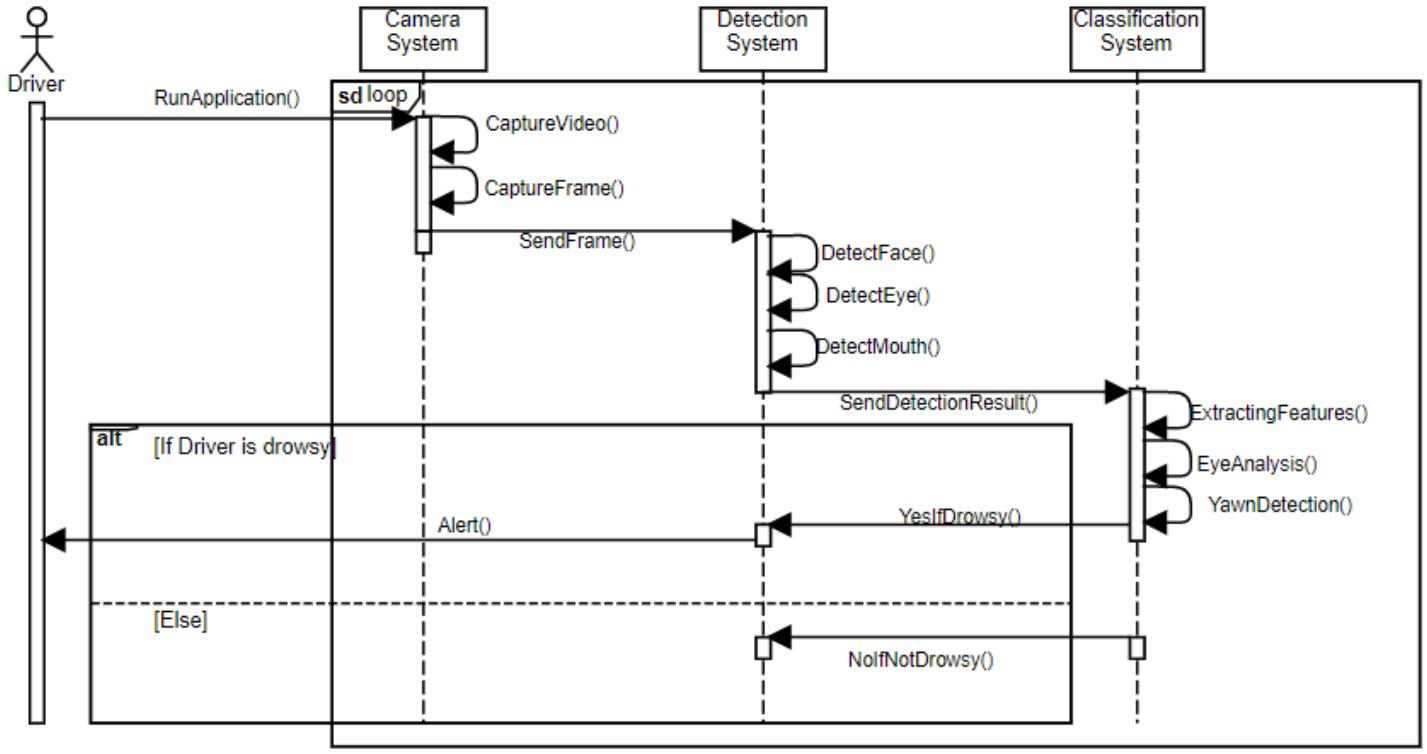


Figure 9: Sequence Diagram for Drowsy Driver Detection System

4.2.2. User Interface Design

4.2.2.1. Login Page

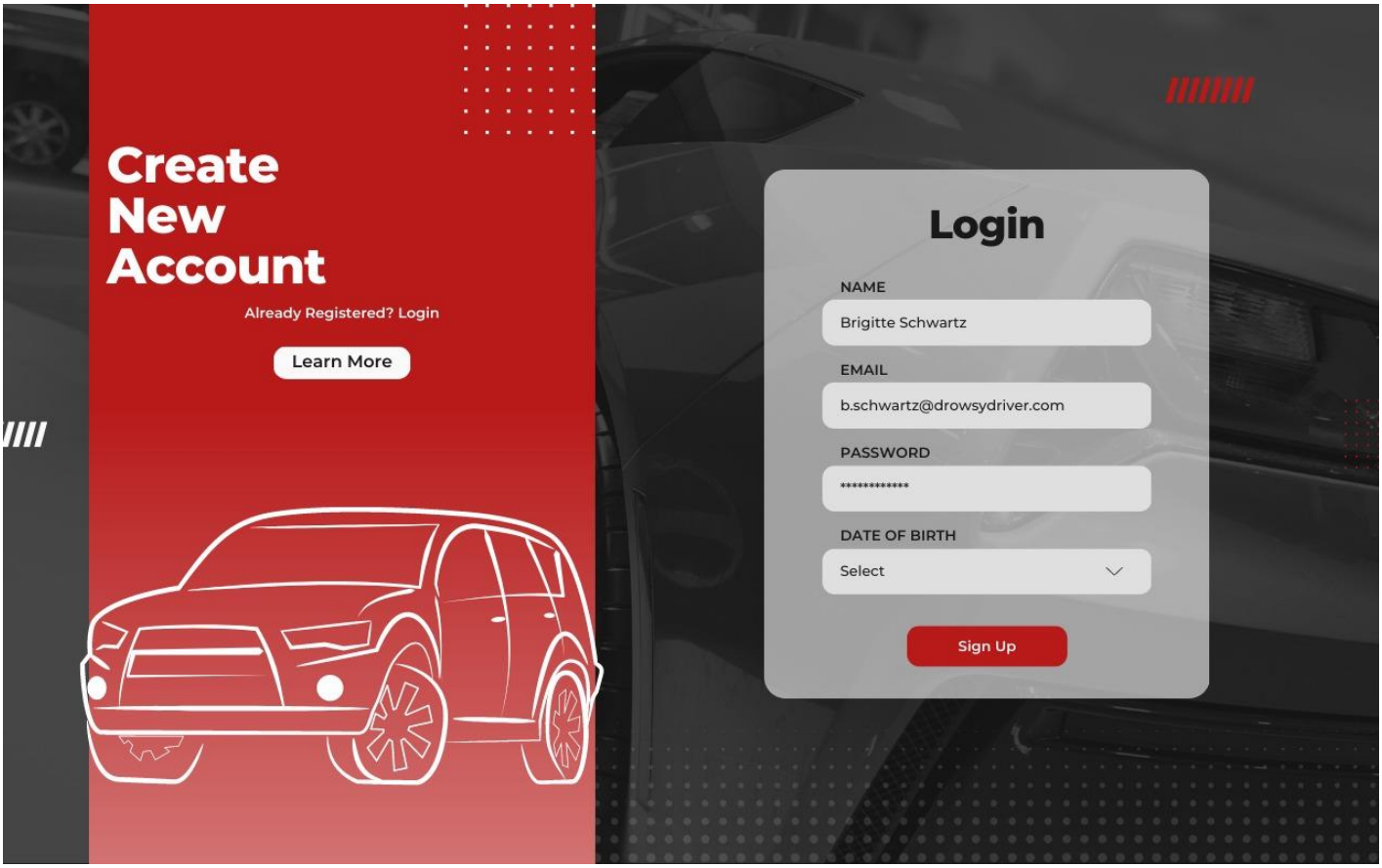


Figure-10 UI for Login Page

In the figure above, there is the login page of our application. Driver must login before using our app. After this entry, the processes will start automatically. It is designed so that the user can use it easily and get used to it immediately. Thanks to this design, it is aimed to attract the attention of the user to the system and to increase the efficiency.

4.2.2.2. Password Page

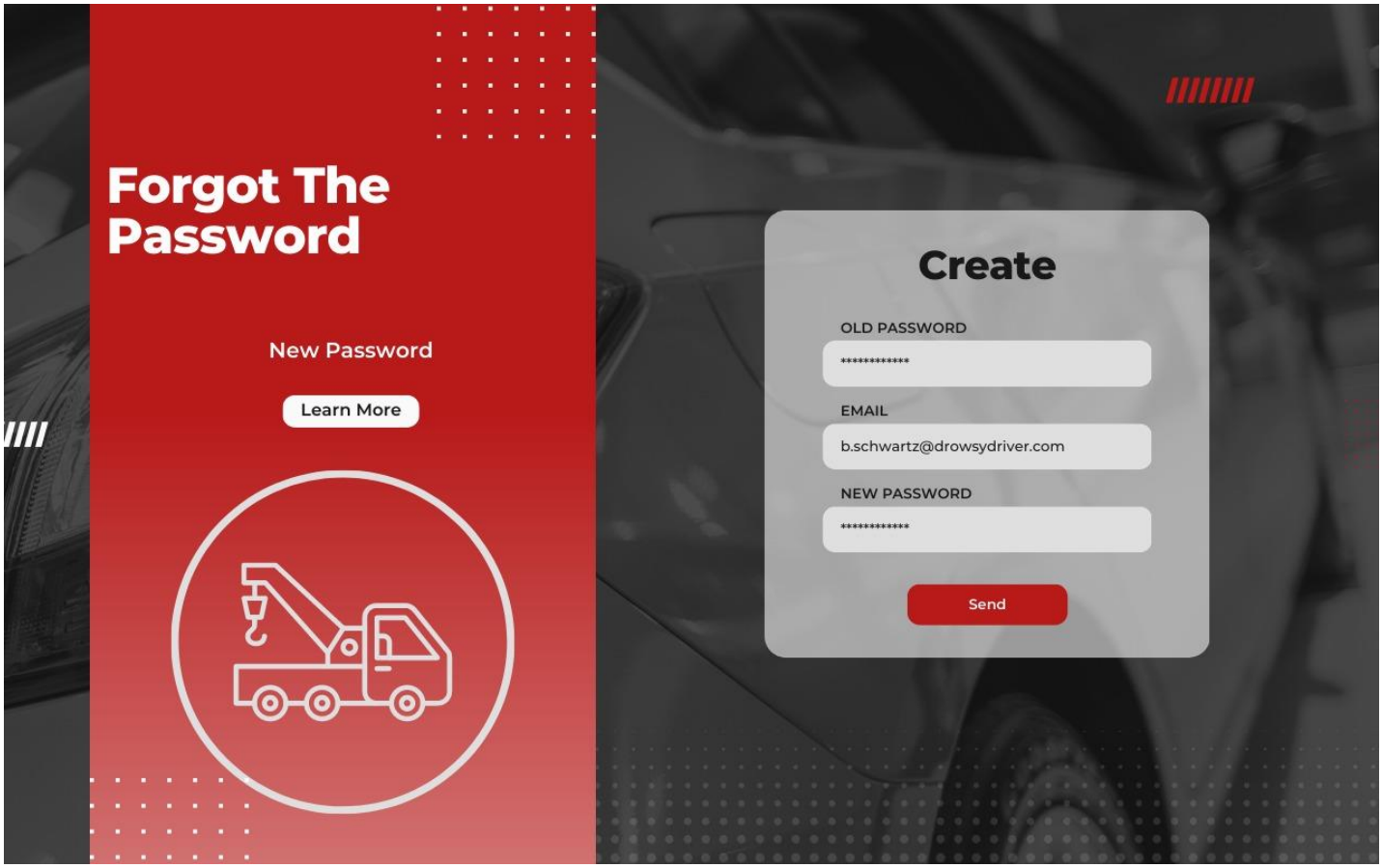


Figure-11 UI for Password Page

In this figure, the system is the interface to be used to enable the driver to log in again when an error occurs in any input.

4.2.2.3. Detection Page

Figure-12 UI for Detection Page

This part is the main part of the application. There is a section where the classification and detection results are shown, and even the values and warning system according to these results. It is a very important interface for the system. It is designed simple so that the user can easily perceive the alert system in any situation.

4.2.2.4. Create Account Page

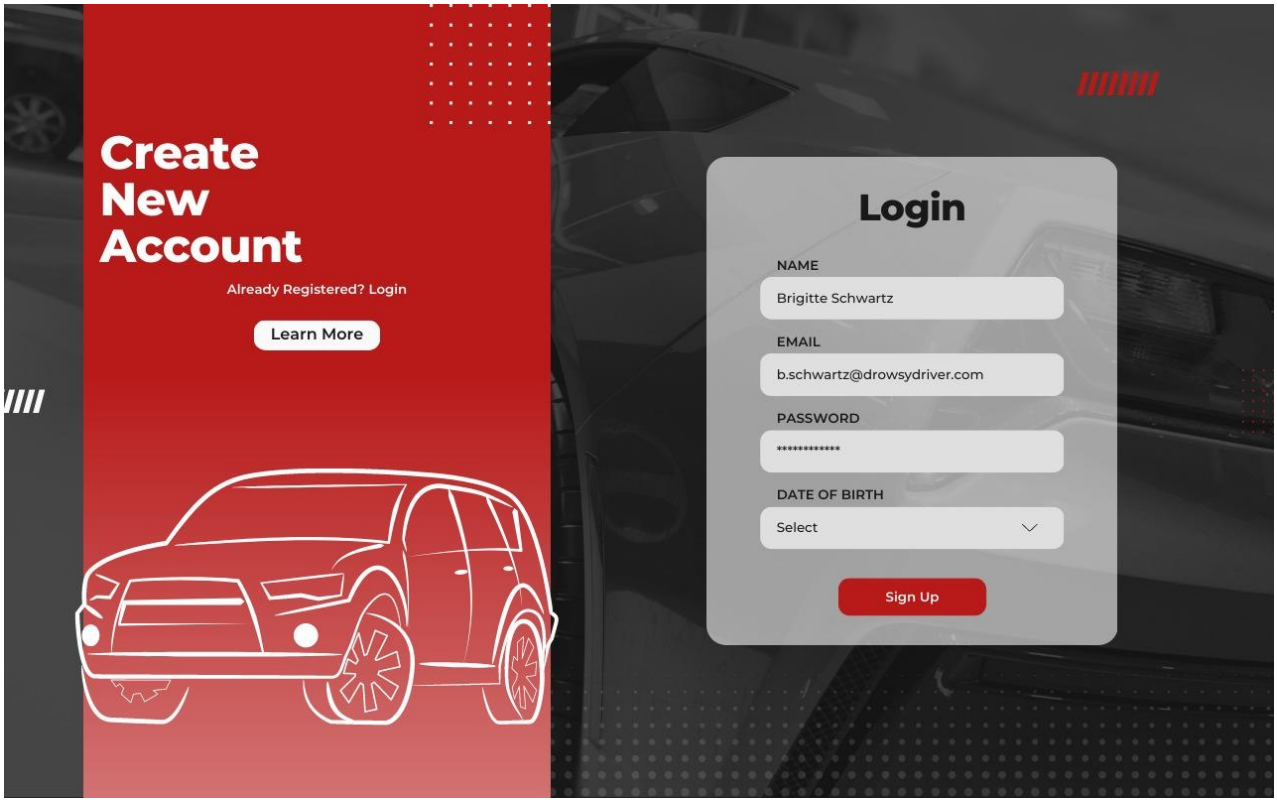


Figure-*12:* UI for Creating an Account

In the figure above, there is the create account page of our application. Driver must create an account before loginning our app. After this entry, the system will direct user to login page automatically. It is designed so that the user can use it easily and get used to it immediately. Thanks to this design, it is aimed to attract the attention of the user to the system and to increase the efficiency.

4.3. Requirements Matrix

Component	Comp-01	Comp-02	Comp-03	Comp-04
Requirements				
Capture Frame	*X*			
Capture Real-time Video	*X*			
Detect Mouth	*X*	*X*		
Detect Eye	*X*	*X*	*X*	
Detect Face	*X*			
Detect Status of Head	*X*			
Performs Alert				*X*
Detect Drowsiness			*X*	

Comp-01 = Face detection for checking drowsiness
Comp-02 = Eye-Mouth Detection for Checking Yawning and Drowsiness
Comp-03 = Classification with Machine Learning Model
Comp-04 = Alert System for Avoiding Drowsiness

Figure-14: Requirement Matrix of Our System

5.1. References

- 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.
- Öztürk, M., KüçükmaniSa, A., & Urhan, O. (2022). Drowsiness detection system based on machine learning using eye state. Balkan journal of electrical and computer engineering, 10(3), 258-263.
- Park, S., Pan, F., Kang, S., & Yoo, C. D. (2017). Driver drowsiness detection system based on feature representation learning using various deep networks. In Asian Conference on Computer Vision (pp. 154-164). Springer, Cham.
- Santoshi, M. G. (2020). AUTOMATED DRIVER DROWSINESS DETECTION FOR NON 2 WHEELERS (Doctoral dissertation, Andhra University).
- BAHARU, B. (2013). DRIVER DROWSINESS DETECTION BY USING WEBCAM.

6. WORK PLAN

The figure below depicts our work plan, which defines our tasks for completing the project. We split them into rows and assigned due dates to each task. We divided each assignment evenly as a team. As a result, each team member worked on each job listed on the Work Plan table. After completing each job, the adviser provides comments.

Advisor: Serdar ARSLAN

Project Work Plan	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14	WEEK 15	WEEK 16
	Oct 14	Oct 17	Oct 28	Nov 4	Nov 11	Nov 18	Nov 25	Dec 1	Dec 9	Dec 16	Dec 23	Dec 30	Jan 6	Jan 13	Jan 21
Team Setup	TEAM														
Project Selection Form	TEAM														
GitHub Repository		TEAM													
Project Work Plan			TEAM												
Literature Review			TEAM	TEAM	TEAM										
SRS Document					TEAM	TEAM	TEAM	TEAM	TEAM						
Project Webpage									TEAM	TEAM					
SDD Document										TEAM	TEAM	TEAM			
Project Report / Project Tracking Form												TEAM	TEAM		
Presentation														TEAM	TEAM

TEAM	Shows what was done in the project.	TEAM	Shows what will do in the project plan

In Figure 15, Work Plan for Our Project