DRIVER DROWSINESS DETECTION

by

Ayşenur Yılmaz 150114002
Mahmut Aktaş 150115010
Mustafa Abdullah Hakkoz 150117509

CSE497 / CSE498 Engineering Project report submitted to Faculty of Engineering
in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

Supervised by:
Prof. Dr. Çiğdem Eroğlu Erdem

Marmara University, Faculty of Engineering

Computer Engineering Department

2021

DRIVER DROWSINESS DETECTION

by

Ayşenur Yılmaz 150114002
Mahmut Aktaş 150115010
Mustafa Abdullah Hakkoz 150117509

CSE497 / CSE498 Engineering Project report submitted to Faculty of Engineering in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

Supervised by:
Prof. Dr. Çiğdem Eroğlu Erdem

Marmara University, Faculty of Engineering

Computer Engineering Department

2021

12/02/2021

ABSTRACT

Driver drowsiness is a continual risk for drivers and it is one of the causes for road accidents. According to statistics, drowsiness causes 11.09% of the total number of accidents. Drowsiness has a huge effect on the drivers' wheel control and driving capability. Hence in this project we built a system which detects the level of drowsiness of the driver and prevents any accidents by giving a warning message to the driver.

Four main aims were determined at the beginning and we reached all aims at the end of the project. The designed system: i) can detect drowsiness with high accuracy (with F1 scores of 89% and 84% on RLDD [9] and NTHU [10] datasets), ii) can detect drowsiness 2 secs early by using an LSTM model, iii) has real-time performance and a simple GUI application and iv) can adapt to subject characteristics (eye anatomy etc.) by using several normalization techniques.

There were some constraints and assumptions as well; for example, we assumed that all datasets are correctly annotated and the drivers' face fits the camera view well and the camera is approximately at one arm length distance.

Our proposed system starts by reading every frame in the video and extracting the facial features. There are a total of 8 hand-crafted facial-geometry-based features, which are extracted from normalized data. In order to achieve better adaptivity for every subject we performed subject-wise normalization as well as minmax scaling for getting better performance from the classifiers. In the classification phase, we followed two different approaches. First approach is a frame-based approach, where we used 12 different classifiers: Logistic Regression, KNN, SVM, Decision Tree, Random Forest, Naive Bayes, Extra Trees, AdaBoost, XGBoost, LightGBM, CatBoost, BaggingClassifier. Second approach uses temporal (sequential) models, where we used LSTM with 5 different variations. LSTM-vanilla, LSTM-stacked, LSTM-bidirectional, CNN-LSTM and Conv-LSTM.

The experimental results show that, best models are XGBoost for the frame-based approach and LSTM-vanilla for the sequential approach. They (or simpler techniques like decision tree / feature thresholding for lower system configurations) are used in the real-time application simultaneously to detect the current drowsiness status and to generate an early drowsiness alert.

ACKNOWLEDGEMENTS

We take this opportunity to express our gratitude to our supervisor Prof. Dr. Çiğdem Eroğlu Erdem and all faculty members of the Computer Engineering Department for their help and support. We also thank our families for the unceasing encouragement, support and attention.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Problem Description and Motivation

Driver drowsiness continues to cause an ongoing risk for the drivers and road safety which makes it one of the major shareholders of the road accidents. According to the statistics, highway road crashes are responsible for 11.09% of the total number of accidents. There are various other reasons for the road accidents alongside with the driver drowsiness, such as high speed, illegal overtaking, parking, unsafe lane changes, overloading etc. It estimated that driver drowsiness causes 2%-23% of the road crashes which can't be ignored. These percentages are still conservative estimations since it is not an easy job to estimate the exact number of the fatigue-related accidents [1].

Light and dark cycles have an impact on sleepiness and wakefulness. That's why humans are mostly awake during daytime and asleep at the nighttime. People who work at nights such as night workers, aircrews, travelers may sleep out of this cycle and can experience sleep loss and drowsiness [2].

Depending on these factors, most drowsiness-related accidents occur during regular sleeping hours, and the more the crash has heavy consequences, there is more probability that the driver was drowsy. Drowsiness is the reason for almost one out of three single-vehicle accidents in the rural areas. It is a misconception that drowsiness is only a problem for the long-distance drivers, but it is also a problem for the short-distance drivers also. Driving a vehicle usually is not the utmost reason for the drowsiness. Drivers are usually already tired when they get into the car because of lack of sleep, long working hours, sleep apnea or shift work.

There are several reasons for drowsy driving;

- A lack of quality sleep,
- Driving when you would normally be sleeping (overnight),
- Sleep disorders such as sleep apnea, a sleeping condition that causes tiredness throughout the day.

All people should acknowledge that, people can't fight with sleep [3].

Drowsiness affects drivers' wheel control and driving capability and risks their life by emerging accidents. According to a research, being awake for 17 hours has the same effect as BAC (blood alcohol concentration) of 0.05 in case of driving ability [4, 5]. Staying awake for 24 hours has the same effect as BAC of 0.1 which is the double of the legal limit [5,6].

Driving while being drowsy or tired can cause:

· Slower reaction times,

· Lack of concentration - errors in calculating speed and distance are common,

· Reduced vigilance and poor judgment,

· Nodding off - even for a few seconds can result in awful consequences,

There are very clear indicators that tells that a driver is drowsy, such as:

· Frequent yawning,

· Inability to keep eyes open,

· Swaying the head forward (i.e., head nods),

· Face complexion changes due to blood flow.

There are numerous ways of avoiding driver drowsiness:

· Getting a good night's sleep before heading off on a long trip,

· Not travelling for more than eight to ten hours a day,

· Taking regular breaks at least every two hours,

· Sharing the driving wherever possible,

· Not drinking alcohol before a trip. Even a small amount can significantly contribute to driver fatigue,

· Not traveling at times when you'd usually be sleeping,

· Taking a 15-minute power nap if drowsy feeling starts [7],

Although these methods can help avoid driver drowsiness, we needed to make sure that drivers are awake while driving.

In this project, we aimed to design a system that tracks the driver's facial behaviors, mainly eyes and mouth, and to be able to detect the drowsiness status of the driver in real-time. In order to achieve this goal, first we did detect the driver’s face and extracted some facial features. After the extraction of these features, we trained a model with drowsy and non-drowsy test data. By using this model, the application can decide if the driver is drowsy or not.

Detecting the drowsiness status for a driver was a challenging goal to achieve since we needed to take several subjects into considerations. For example, collecting real-life data was very hard since it should be in a real car in traffic with not-acting drowsy drivers. Generally, driver drowsiness datasets are divided into two types. First type consists of subjects that are acting as a drowsy person. The second type consists of subjects that are really drowsy on the videos. Although the second one gives more accurate results, both types of datasets were captured in laboratory environments rather than a real car in traffic. We used both types of datasets to train our model to acquire more reliable results. We used both types of datasets individually to train models along with the merging the both of them to train models.

The other difficulty we have faced was individual differences of each face. Everybody has his own characteristics, including eyes and face shape. For instance, some people have more rounded eyes while some people have slanted eyes. Similarly, there are people with long or short eye laces or expressive and non-expressive faces. These differences were causing low drowsiness detection accuracy in models since thresholds for features were the same for each person [8]. The last but not the least difficulty we have encountered was detection faces in different angles, covered with hats or glasses, or being under different light conditions. In order to overcome this obstacle, we used Dlib's face detection algorithm. This algorithm can detect the faces with glasses on, or turned away from the camera at a reasonable angle.

1.2 Aims of the Project

There were four main goals that we set at the beginning of the project. These aims were:

- **Detecting Driver Drowsiness with High Accuracy**

The main aim of this project was to detect driver drowsiness. We have developed a system that is able to detect drowsiness status of the drivers with 89% accuracy by only using the features that we extracted from the human face and eyes with an average laptop webcam. This score is 27.6% higher than the state-of-art result on the UTA-RLDD dataset [12].

- **Early Detection**

The other essential aim for this project was predicting the driver's drowsiness before the driver falls asleep. Drivers must be warned before they fall asleep while driving. We have achieved this goal by predicting the driver's drowsiness status 2 seconds ahead. We did several experiments with different

LSTM models and determined that LSTM-Vanilla produces the best result among them.

- **Real-Time Performance**

Another important and crucial aim for this project was real-time performance. The system can evaluate features and predict the drowsiness state two seconds earlier and the current drowsiness status in real-time using an average laptop with less than 2 seconds delay.

- **Adaptivity to the Subject**

The final aim of this project was to build a system that will be able to detect the drowsiness of all people from different ethnicities and personal characteristics. Since this project is about detecting driver drowsiness based on the facial features of the human face, we adjusted it so that it is adaptive to the different kinds of facial characteristics. We achieved this goal by estimating the driver's face characteristics before predicting the drowsiness status. The project uses the first 5 seconds to calibrate itself by evaluating the driver's awake face's features. It sets dynamic thresholds for each person and then starts predicting depending on these thresholds.

2. DEFINITION OF THE PROJECT

2.1 Scope of the Project

The main aim of this project was to build a system that detects the drowsiness of the driver and gives a warning to the driver in real-time using image processing and machine learning techniques in order to minimize the traffic accidents due to fatigue. The project consisted of two phases. In the first phase, we developed the frame-based facial feature extraction system and normalized the values by using each subjects' personal facial characteristics. In the second and the final phase, we did experiments to determine the appropriate classifier methods for both real-time detection and predicting the future. We have used several classifiers in our experiments such as SVM, Decision Tree, XGBoost etc. for real-time detection and LSTM for future prediction.

In this project, UTA Real Life Drowsiness Dataset [9] and NTHU-DDD [10] is used for both training and test data. Since this work was a research-based project, an implementation on a computer system was a primary objective of ours. Thus we have developed an application that predicts both current and future drowsiness state prints the features and the results on the screen. In order to develop such an application, we have used Python for prediction, Flask [61] for sending the real-time camera capturing and sending the prediction and the feature results to the frontend, ReactJS [63] for printing these data.

2.2 Success Factors and Benefits

2.2.1 Success Factors

There were four aims for our project denoted in section-1.2. First one was detecting driver drowsiness with higher accuracy. We used the accuracy metrics to check if results are reaching the baseline level on UTA-RLDD dataset's state-of-the-art results. We have reached 89% F-1 score which is 19% higher than the state-of-art result on the UTA-RLDD dataset.

Second aim was early detection of potential sleepiness of the driver by using sequential methods like LSTM. Success metric for this aim was determined as how many seconds before the danger level could be detected and the desirable result was at least 3 seconds. We achieved early detection with 2 seconds for the future.

And the third aim of the project was working with real-time performance. Despite its complicated workflow, the proposed system must not be delayed much more than 2 seconds for a typical 10 minutes video of the UTA-RLDD dataset, since predictions of a potential accident will be in a three seconds limit as explained above. We achieved this aim with about 1 second delay in our real-time demo application.

The last aim was adaptivity to the subject's personal and racial characteristics such as eye shape, skin color, blink behaviors etc. Success factor for this aim was consistency on accuracy levels across subjects in the videos. We achieved this aim with normalization phase and testing accuracy metrics on selected false-positive samples with different characteristics from the datasets.

2.2.2 Benefits

With the success of this project, we get four benefits. First benefit of this project is driving safety. This project can help people drive safely without causing an accident due to drowsiness. Second benefit of this project is preventing economic loss. Since this project aims to reduce road crashes, the cost of these road crashes reduces also. Third benefit of this project is to lower the environmental impact of the accidents caused by road crashes. And the final benefit would be offering improvements to autonomous cars with provided features.

2.3 Professional Considerations

2.3.1 Methodological Considerations and Engineering Standards

We used Github [66] as a version control system. Other than the source codes, experimental notebooks are also available publicly on Kaggle Kernel [67] platform. As decided at the beginning of the project, we worked on Python programming language due to its wide variety of libraries on computer vision and machine learning domains and also Python's Flask framework has been used for the backend part of the demo application. For the frontend part of the demo application, ReactJS has been used. As computer vision libraries, Python forks of OpenCV [51] and Dlib [69] have been used. To implement machine learning and deep learning models, Scikit-Learn [59] and Keras [60] has been used.

2.3.2 Societal/Ethical Considerations

- **Economical**

Every year economic and societal impact of road crashes costs United States citizens $871 billion. $380 million of this money is directly spent on medical costs [11]. Considering these facts, we can say that road crashes have a huge impact on the economy. This project aimed to reduce car accidents caused by drowsy drivers. By avoiding this cause of road accidents, this project can reduce the accident's impact on the economy.

- **Health and Safety**

The main purpose of the project is to avoid road crashes caused by drowsy drivers. Therefore, this project can save many lives by warning the sleepy drivers before an accident occurs. Since a road accident may involve more than one person, this project may also be preventing the damage to the other drivers or pedestrians on the road at the time of the accident. Lastly accidents can cause mental illnesses on the victims' loved ones, so this mental damage would be prevented with this project.

- **Environmental**

Road crashes not only affect health and economy, but they also affect the environment. Car accidents often cause gas and liquid leaks emitting harmful chemicals into the environment that can poison grass and neighboring plants and harm wildlife. The other damage environment takes after the accidents is landfills. When a car is totaled, most insurance companies determine it's more economical to replace the vehicle than repair it. Though many car parts can be recycled, most of the vehicles end up in a landfill where it will take thousands of years for all the pieces to decompose. Many car parts are also left on the side of the road where they can harm animals or plants. The effects on soil, water, and air pollution influence the entire ecosystem [46].

2.3.3 Legal Considerations

Python's open-source libraries were used for the project. Likewise, we used open datasets such as NTHU-DDD and UTA-RLDD. Some of the participants (24 out of 60) in the UTA-RLDD don't allow their faces to be published in any future paper. A Complete list of them can be seen here [9]. Also, there is a license agreement for NTHU-DDD [10] to be signed for the usage of databases for research purposes. Flask is licensed under a three clause BSD

License. It basically means: do whatever you want with it as long as the copyright in Flask sticks around, the conditions are not modified and the disclaimer is present. ReactJS is licensed with MIT license which gives users express permission to reuse code for any purpose, sometimes even if code is part of proprietary software. As long as users include the original copy of the MIT license in their distribution, they can make any changes or modifications to the code to suit their own needs.

In the case of the proposed project, although the system does not keep any personal data of subjects, users still need to allow the system to capture themselves while the system is on.

2.4 Literature Survey / Related Works

2.4.1 Models and Techniques

Since the early 2000s, the automobile industry has spent a huge amount of time and resources with the collaboration of researchers from academia to build

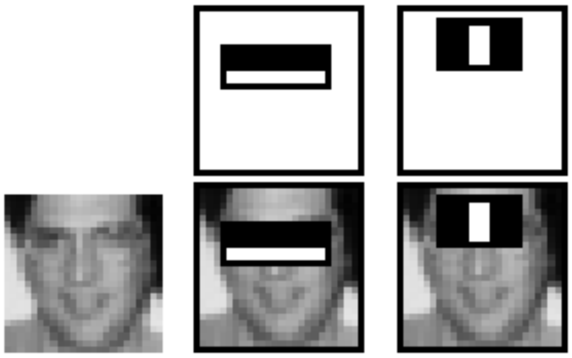

Figure 2.1 An original figure from the 2001 Viola-Jones paper [26].
Detecting eyes and nose with two significant Haar-like features.

proper **DDD** (Driver Drowsiness Detection) systems [13, 14, 15, 16, 17, 18]. While some of researchers prefer to use intrusive methods **physiological sensors** such as ECG (Electrocardiogram) [19], EEG (Electroencephalogram) [20], EOG (Electrooculogram) [21] and **vehicle-based methods** such as observing steering wheel movements [22] and lane deviation [13]; there is an increasing trend of using computer vision and machine learning techniques on the driver's video to examine facial behavioral signs (**head position** [23, 2], **yawning** [24], **blinks** [12], or other facial actions like state of **eyebrow**, **lip** or **jaw** [25]), since they are non-intrusive and highly accurate methods.

Generally, in computer vision systems object detection is based on methods extracting features from pixel data with different techniques. Same concept also goes in drowsiness detection systems, first it’s necessary to detect the driver's face, then detect facial members to produce features by interpreting them

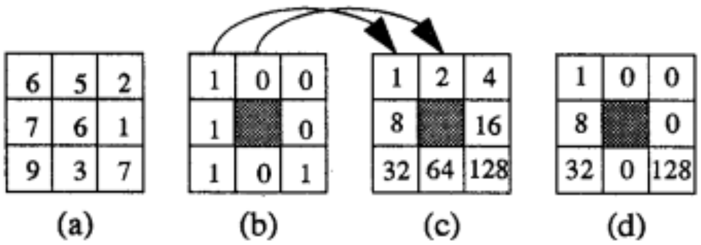

Figure 2.2 An original figure from the 1994 LBP paper [27]. It uses center as threshold, marks neighborhood and converts the whole square to binary code.

in algebraic and algorithmic processes. Some of the widely used face detection methods are; **Viola-Jones algorithm** [26, 24] which uses **light intensity** difference on eye-nose regions (see Figure 2.1), **color-based methods** that use a range for color tones of human skin [20, 24], **LBP (Local Binary Patterns)** [27, 28] to detect micro-textural patterns by using regional color intensities (see Figure 2.2), **Gabor Filters** to enhance topological structure of human face, see the changes occurring and detect faces this way [29], **Circular Hough Transform** [30] to catch infrared sensitivity of eye-pupil, **feature descriptor algorithms** such as **SIFT** (Scalar- Invariant Feature Transform) [31], **SURF** (Speeded-Up Robust Features) [31], **BRIEF** (Binary Robust Independent Elementary Features) [32], **ORB** (Oriented FAST and Rotated BRIEF) [32] to find key points then describe their importance with different heuristics by using **DoG** (Difference of Gaussian) [30, 33, 34], and finally a fast and robust algorithm **HOG** (Histogram of Oriented Gradient) [35,24,36] to use the distribution of directions of gradients (see Figure

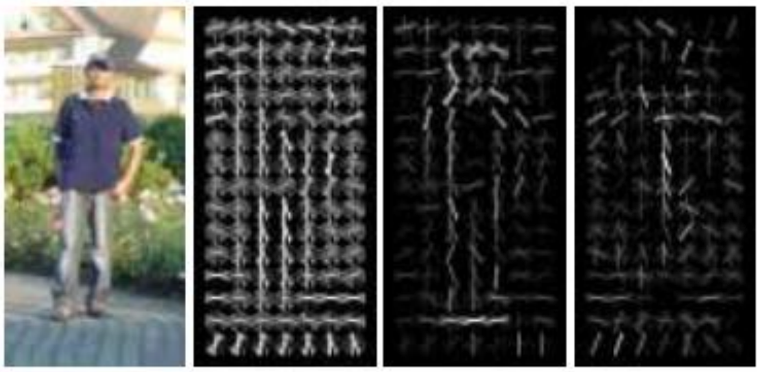

Figure 2.3 An original figure from the 2005 HOG paper [35] showing orientations of gradients.

2.3). The last method, **HOG**, is also used by a popular computer vision library **Dlib** [36] and in our project we decided to use it due to its lower computational cost and high accuracy.

After detection of face and facial members, it's necessary to produce some meaningful numerical values to predict drowsiness of the subject. Some popular methods are **PERCLOS** (Percentage of Eye Closure) [37, 38, 39], **MAR** (Mouth Aspect Ratio) [24, 40, 41] and **EAR** (Eye Aspect Ratio) [42]. Since they are also used in our project, they will be explained in their own sections 4.2.1 in detail.

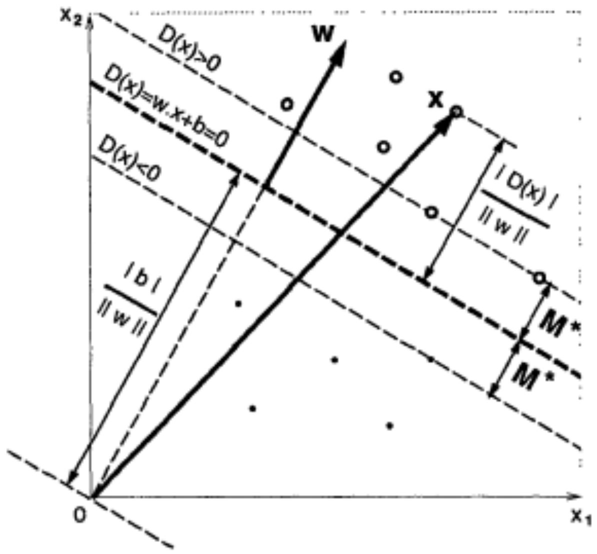

Figure 2.4 An original figure from the 1992 SVM paper [45] showing decision.

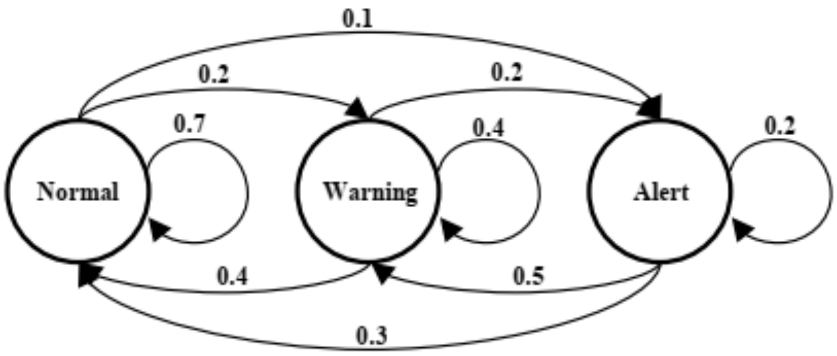

Figure 2.5 An original figure from the 2016 paper [44] showing probabilistic transitions between three possible states: *Normal, Warning, Alert.*

After extracting features from raw data and constructing training datasets, there are also many choices of classifiers to predict drowsiness level. Some of them are; **simple thresholding** with EAR [34, 42] or nodding [2], conventional machine learning tools like **Logistic Regression**, **Decision Tree**, **k-**

NN (K-Nearest Neighbors), **NB** (Naïve-Bayes) [25, 27, 40, 41], **SVM** (Support Vector Machine) [21, 24, 30, 2, 33, 38, 42, 40], **Random Forest** [24, 27, 41], **HMM** (Hidden Markov Model) [1, 25, 43, 29, 44], **AdaBoost** (Adaptive Boosting) [26, 29, 2] and even **XGBoost** (Extreme Gradient Boosting) [27, 41]. Amongst them most popular ones are **SVM** and **HMM** (see Figure 2.4 and 2.5).

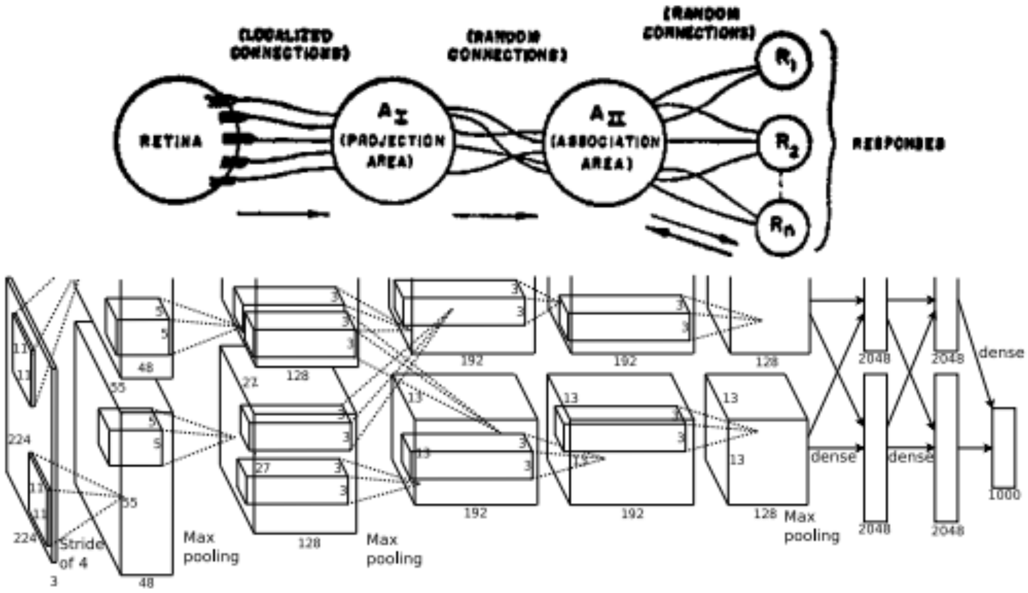

Figure 2.2 First is original figure from the 1958 Perceptron paper [47] and second is ImageNet2010 contest winner AlexNet, a CNN implementation [48].

In drowsiness detection domain, although some papers use vanilla neural networks [19, 34], in most cases researchers prefer to use **CNN** (Convolutional Neural Network) as classifier [38, 41] almost as common as **SVM** and **HMM** (see Figure 2.6). There's also an increasing popularity of **RNN**s (Recurrent Neural

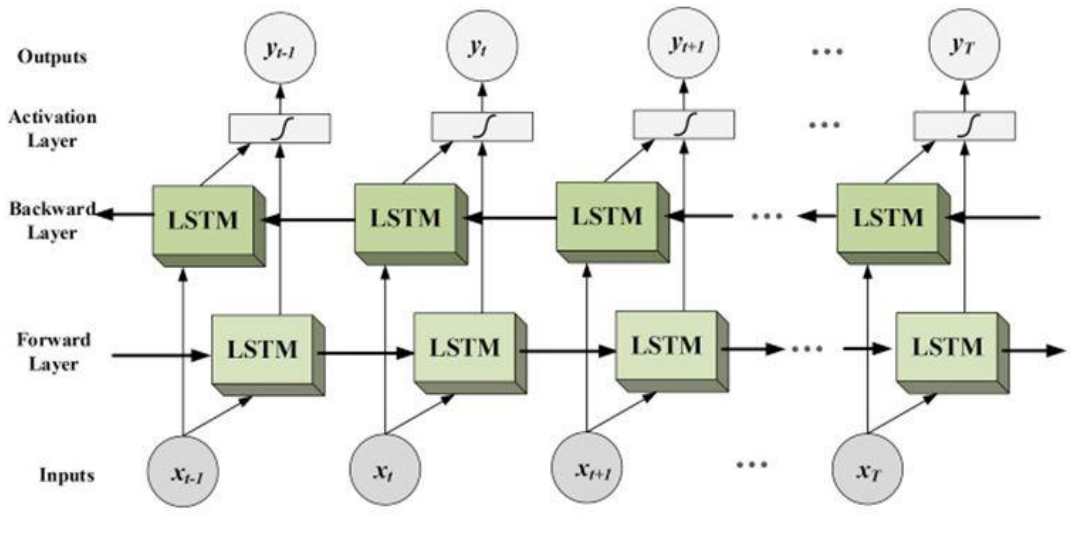

Figure 2.3 An example architecture of Bidirectional LSTM. It process input sequence in two directional: *Forward and backward* (Image source: [58]).

Networks) to open the possibility of using sequential data just like HMMs**. LSTM** (Long-Short Term Memory) also has a promising future in drowsiness detection systems, although its usage recently started with just a few examples [12, 41].

In our project, we used **simple thresholding** and all conventional machine learning tools available in **Scikit-learn** library [59] along with XGBoost, LightGBM and CatBoost. We also designed a set of experiments with LSTM variations available in **Keras** library [60].

2.4.2 State-of-Art

The primary challenge in DDD literature is each of research using different datasets [49] and absence of a standard, large and realistic datasets that can be used as benchmarks [12]. There is an example of the effort in 2017, about comparing different works on different datasets by using a meta-analysis approach (see Figure 2.8) which indicates **CNN** as a most successful classifier against **SVM** and **HMM** [49]. Yet, it's still not enough to predicate state-of-art results because some of the databases are not open to public access and open ones mostly consist of actor subjects or non-realistic environments and these are the main reasons shadowing accountability of works in the literature.

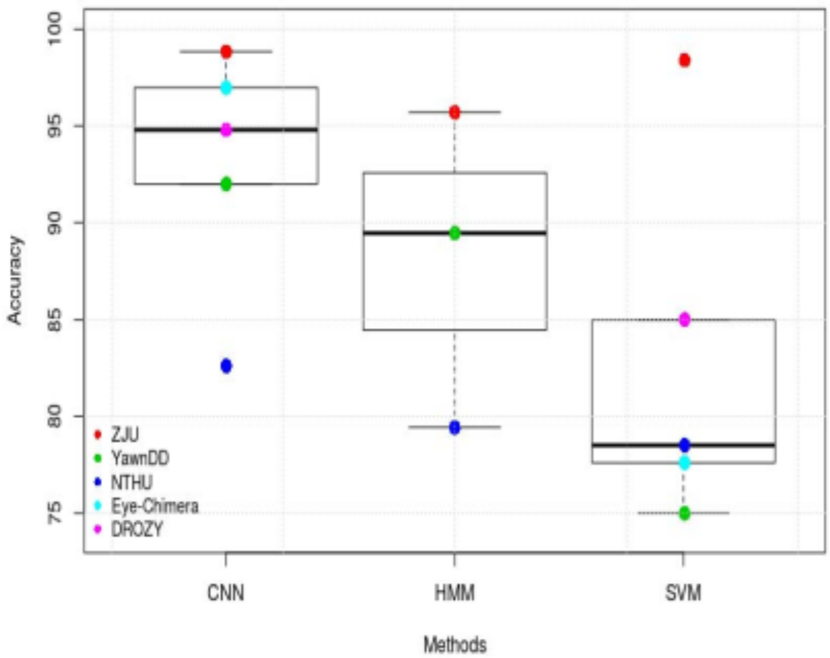

Figure 2.8 An original figure from the 2017 survey paper [49] showing boxplots for comparing ML methods on most common datasets by the date.

Model	Evaluation Metric			
	BSRE	VRE	BSA	VA
HM-LSTM network	**1.90**	**1.14**	**54%**	**65.2%**
LSTM network	3.42	2.68	52.8%	61.4%
Fully connected layers	2.85	2.17	52%	57%
Human judgment	—	2.01	—	57.8%

Figure 2.9 An original figure from the 2019 LSTM paper [12] showing results of Video Accuracy (VA) and some other proposed evaluation metrics.

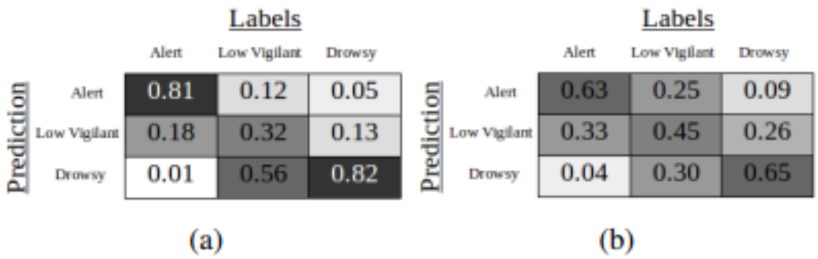

Figure 2.10 An original figure from the 2019 LSTM paper [12] showing confusion matrix of LSTM model (a) and human judgement line (b).

But recently there is one novel work by Ghoddoosian et al. [12], which aims to fill the gap of benchmark datasets and introduces **UTA-RLDD** dataset with a baseline method on it. They propose an LSTM-HM model and declare their results on their own dataset as **62.5%** (70% F-1 score on Drowsy label can be calculated from confusion matrix in Figure 2.10-a) where human judgement is **57.8%**. Hence, these can be used for future researchers to compare their works. Baseline results obtained for this dataset in the original paper are given in Figure 2.9 and 2.10.

We also worked on this dataset so results above are selected as state-of-art because it's a novel work and there's no official research using the database yet rather than a blog post [41]. Additionally, surpassing it is determined as one of the aims of the project since the level of goal is not so much high unlike the other datasets in the literature.

On the other hand, for NTHU-DDD dataset, there is one novel work by Hu et al. [50], which aims to fill the gap of benchmark datasets and uses NTHU-DDD dataset with a baseline method on it. They propose 3D Conditional GAN and Two-level Attention Bi-LSTM model and declare their results on NTHU-DDD dataset as in Figure 2.11. Hence, these can be used for future researchers to compare their works. Quantitative results also show that the 3DcGAN network achieves the total accuracy rate of 82.8% with detection rate 82.3% and false

Model	DR(%)	FAR(%)	AR(%)
3DDIS	74.1	29.0	72.6
3DDIS-A	74.7	27.7	73.6
3DDIS-B	77.6	23.9	76.9
3DGAN	76.4	25.6	75.4
3DcGAN-A	76.9	24.5	76.2
3DcGAN-B	81.9	18.6	81.7
3DcGAN	82.3	16.5	82.8

Model	DR(%)	FAR(%)	AR(%)
3DcGAN+LSTM	83.8	15.6	84.1
3DcGAN+BiLSTM	85.5	15.0	85.3
3DcGAN-ALSTM-A	86.2	14.1	86.0
3DcGAN-ALSTM-B	86.0	14.9	85.6
3DcGAN-TLALSTM	86.9	14.0	86.5
3DcGAN-BiALSTM-A	86.5	13.8	86.3
3DcGAN-BiALSTM-B	86.6	14.4	86.1
3DDIS+TLABiLSTM	82.1	18.8	81.7
3DcGAN+TLABiLSTM	87.5	13.3	87.1

Figure 2.11 The ablation analysis of the 3D-gan networks on NTHU-DDD testing dataset (left) and several LSTM-based techniques on NTHU-DDD testing dataset (right) [50].

alarm rate 16.5% amongst other types of generative models. Then they improved these results by using LSTM-based techniques. The model inputs consecutive short-term drowsiness-related representation, captures temporal dependencies and outputs the long-term drowsiness score of each frame.

2.4.3 Selected Works

There are three research papers having significant importance in the project, so they are explained briefly in this section.

- **A Practical Driver Fatigue Detection Algorithm Based on Eye State**

In the article of Liu et al., 2010 [39], authors aim to detect driver drowsiness by calculating **PERCLOS**. Firstly, they collected the tested videos in the natural driving conditions with the active infrared camera fixed on the car dashboard, and the videos were taken under day and night with different drivers.

In order to achieve detecting driver drowsiness firstly they detect faces with Viola-Jones algorithm and they detect the eyes from detected faces then they adopt mean shift algorithm in case the eye detection fails. After detecting the eyes, they adjusted the contrast of the eyes in order to remove the influence of bright spots or specular reflection caused by glasses or hard light. Next, they used their filter to detect the eye corners. Finally, they striped the area between right and left eye corners into five and applied a simple filter to each area. Maximum and minimum responses that are given to this filter are up and down eyelids respectively. After extracting the eyelids, they calculated the distance between eyelids. By looking at the distance of eyelids, eye closeness has been detected. And finally, by counting eye-closeness they calculate PERCLOS.

In conclusion the algorithm developed in this article is capable of detecting eye closure at high speed after locating the eye. Although it is irrelevant to the subject's gender and day and night, glasses are a problem for this algorithm.

We used this paper to implement the PERCLOS feature for our project. Instead of using the proposed algorithm, we used **Dlib facial landmark detector** to detect the eye landmarks. After that we applied the proposed PERCLOS formula and extracted this feature.

- **A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection**

Main motivation of the paper (Ghoddoosian et al., 2019) [12] is collecting the largest, public, realistic dataset to 2019: UTA-RLDD. Additionally, introducing a baseline method to detect early, subtle cues which uses Hierarchical Multiscale Recurrent Neural Networks, specifically HM-LSTM, resulting in a higher accuracy (65%) than human observers (under 60%).

The Proposed Baseline Method uses Dlib's **face detector** (HoG version) and detecting blinks by an improved algorithm of [42] by specializing for consecutive quick blinks. Input of the blink detection module is frames of last minutes of real time video and output of the blink detection module is a sequence of blink events: {$blink_1$, …, $blink_k$} where each $blink_i$ is a 4D vector [duration, amplitude, eye opening velocity, frequency]. After preprocessing and normalization, the model continues with a 4-D feature transformation layer with following **HM-LSTM** layer and finally four **Fully Connected Layers**.

Consequently, the paper represents publicly available, real-life dataset **RLDD** which is the largest to date (2019) along with end-to-end baseline method using the temporal relationships between blinks to detect early signs before an accident. Overall, the paper hopes that the proposed public dataset will also encourage other researchers to work on drowsiness detection and produce additional and improved results that can be duplicated and compared to each other.

This is one of the most important articles for this project. We extracted the facial features with Dlib same as this article. Although we did not implement the blink-based features, we used the proposed UTA-RLDD dataset to train our models. In order to train our future prediction models, we used LSTM as proposed in this paper.

3. SYSTEM DESIGN AND SOFTWARE ARCHITECTURE

3.1 System Design

In this project, we studied about how to implement a real-time drowsiness detection system with our four predefined aims: high accuracy, real-time performance, early detection and subject adaptivity.

3.1.1 System Model

The driver must set up a camera before getting behind the wheel in a proper way so that the driver's full face can be captured by the camera while driving. Then the camera captures the face and our system is able to detect facial landmarks using the Dlib library. In that way, the system is able to extract the handcrafted facial features. During this, the driver should look directly to the camera for five seconds before starting to drive, we used these five seconds to use the feature normalization phase. From there our system arranged the corresponding threshold values for each feature by the driver's face. This enabled the system to become adaptive by the subject (Figure 3.1).

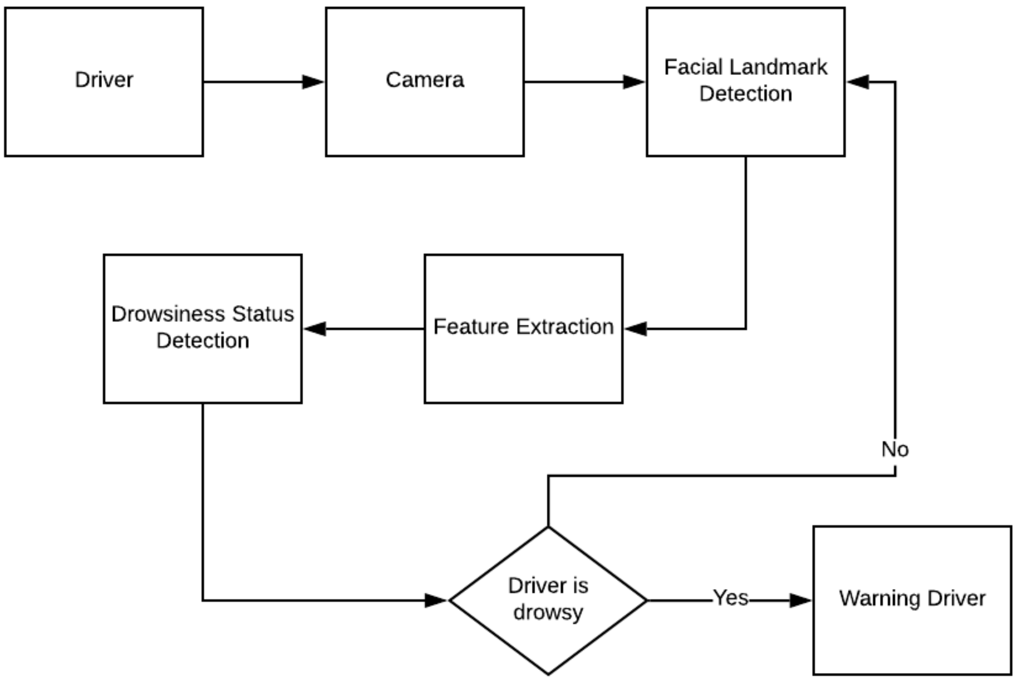

Figure 3.1 A real-time Driver Detection System Model. Using predictions of the DDD system to warn drivers in dangerous situations.

Consequently, drowsiness status of the driver is estimated using a pre-trained model. Depending on this status, we gave a warning to the driver if it was necessary. If the system detected the driver is not drowsy then basically it continued to evaluate the driver's status until the system detected the driver as drowsy or there was some obstacle in front of the face.

3.1.2 Flowchart

The system started by reading videos, processing it with detecting faces and creating facial features. We made subject-wise normalization to handle adaptivity issues. The system produced frame-based features. We had two different datasets, NTHU-DDD dataset provides frame-based labels with two classes: drowsy or alert; on the other side RLDD dataset provides video-based labels with three classes: drowsy, low-vigilant or alert (Figure 3.2).

In this project we worked with two different approaches in the classification phase: frame-based models and sequential models. For the frame-based models we used SVM, Decision Tree, k-NN, Random Forest, XGBoost,

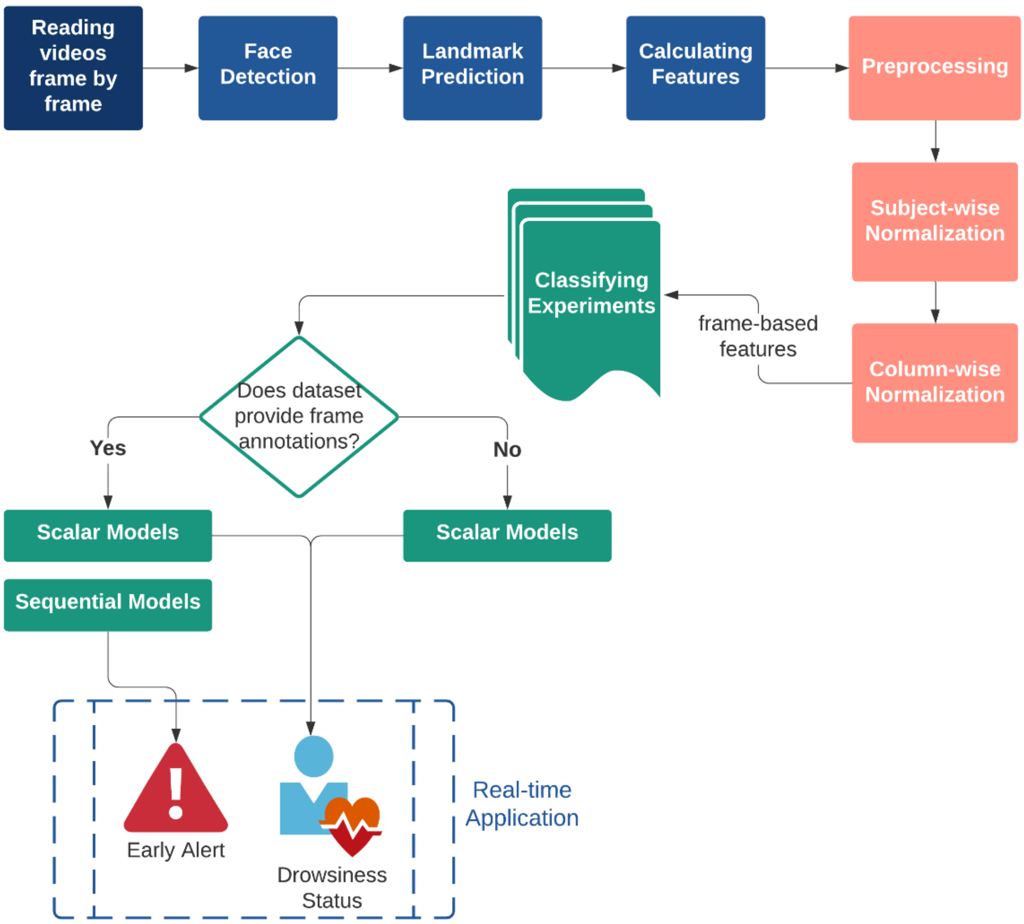

Figure 3.2 Flowchart diagram of the driver drowsiness detection system.

Gradient Boost, ExtraTrees and Naive Bayes algorithms. For sequential models we used LSTM models with its variations: LSTM-Vanilla, LSTM-Stacked, LSTM-Bidirectional, Conv-LSTM and CNN-LSTM.

3.1.3 Comparison Metrics

For classification, we used standard comparison metrics defined in Table 1 and 2.

Table 3.1 Symbol definitions for comparison metrics.

	Actual-Drowsy	Actual-Not Drowsy	Total
Predicted-Drowsy	a	b	a+b
Predicted-Not Drowsy	c	d	c+d
Total	a+c	b+d	a+b+c+d

Table 3.2 Formulas for comparison metrics.

Metric Name	Definition	Ideal Value
Accuracy	(a+d) / (a+b+c+d)	1
False Positive Percentage	b / (a+c)	0
False Negative Percentage	c / (b+d)	0
Precision	a / (a+b)	1
Recall	a / (a+c)	1
F-1 Score	(2*precision*recall) / (precision + recall)	1
AUC score	Area under ROC curve	1

For regression tasks, we used Mean Squared Error (MSE) as defined below:

$$MSE = \sum_{i=1}^{n} \frac{(\widehat{y_i} - y_i)^2}{n}, \qquad (Equation\ 3.1)$$

Where $\widehat{y_i}$ represents predicted results, y_i represents ground truth labels and n is the size of a test fold.

3.1.4 Dataset or Benchmarks

Specifications of datasets are used in the project are listed below:

UTA-RLDD: UTA Real Life Drowsiness Dataset [9] is used for both training and test data. It is created in the University of Texas at Arlington by a research group to detect multi-stage drowsiness. This dataset is obtained with the participation of 60 healthy people and three different videos of each participant is taken: alertness, low vigilance and drowsiness that is a total of 180 videos. It is composed of around 30 hours RGB videos. There were 51 men and 9 women from different ethnicities like 10 Caucasian, 5 non-white Hispanic, 30 Indo Aryan and Dravidian, 8 Middle Eastern, and 7 East Asian and age ranges between 20 and 59 and there are 21 out of 180 participants were wearing glasses and 72 out of 180 participants had facial hair.

NTHU-DDD: This dataset [10] consists of 36 subjects of different ethnicities. It includes many variations of driving scenarios such as normal driving, yawning, show blink rate and falling asleep. The total time of videos is almost 10 hours. There are 5 different scenarios; bare face, glasses, night bare face, night glasses and sunglasses. Each record is approximately 1 minute long. The participants simulated driving in a lab environment. The evaluation and testing datasets contain 90 driving videos (from the other 18 subjects) with drowsy and non-drowsy status mixed under different scenarios. Main property of the dataset is usage of active IR illumination to acquire IR videos.

3.2 System Architecture

This section explains our drowsiness detection system architecture. Figure 3.3 shows the data/control flow of the system. The data passes from **(1)** path is the dataset videos with different subjects. The system reads these videos frame by frame and with the path **(2)** it tries to detect face. If the current frame consists of a face then the system locates the facial landmarks by using Dlib **(3)**. Then the preprocessing stage ends and with the path **(4)**, the system sends landmark locations to the feature extraction stage. This stage consists of one operation. The operation is extracting frame-based features. Frame-based model evaluates the features in each frame individually. The system executes feature

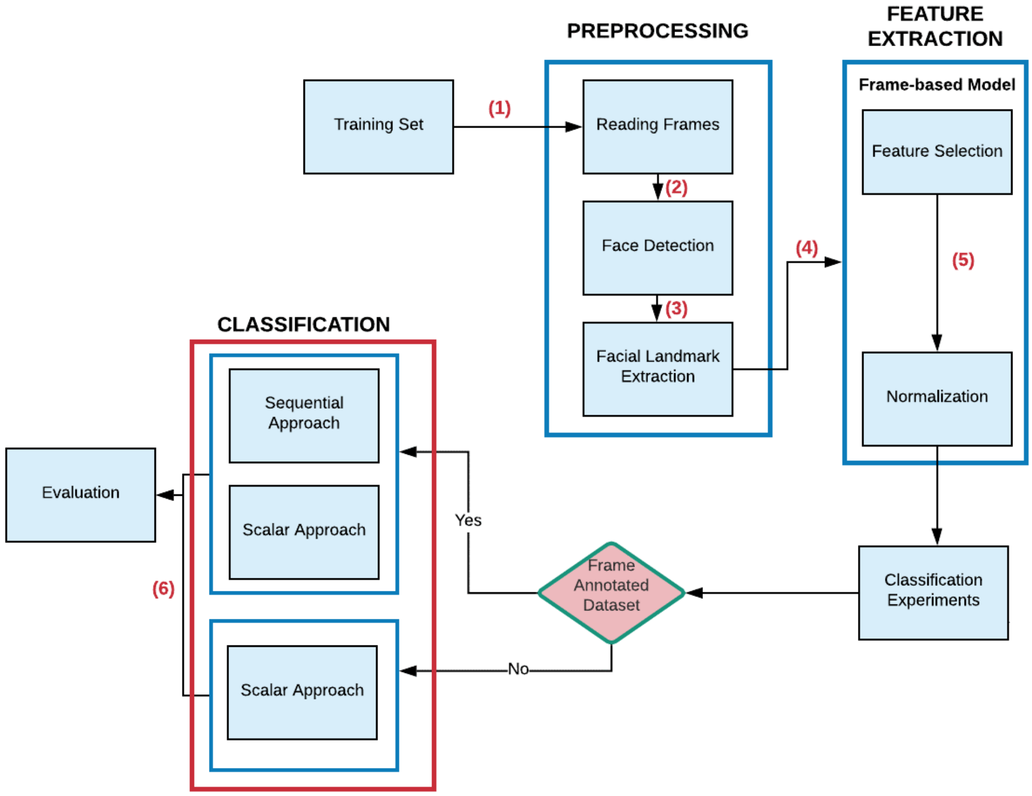

Figure 3.3 Data / Control Flow of the Driver Drowsiness Detection System. Flow of the data through the system is: videos of the dataset (1), gray scale frames of each video (2), face region of the frames (3), facial landmark locations of the face (4), calculated features (5), trained models (6).

selection algorithms for frame-based models and normalizes the features by both subject-wise and column-wise **(5)**. At the end of this stage, we have one dataframe for the frame-based model. This dataframe flow through for classification stage.

We have two datasets for this project named as NTHU-DDD and UTA-RLDD. NTHU-DDD consists of drowsiness status annotation for each frame. On the other hand, UTA-RLDD is a video-labeled dataset. So that if the current dataset is frame-annotated then we use both sequential and frame-based classification methods. If it is video-labeled, then we use only frame-based methods. After the classification stage is over, trained models flow through the **(6)** and tested with the test sets.

4. TECHNICAL APPROACH AND IMPLEMENTATION DETAILS

If we look at the system with more detail, the whole pipeline of the project comprises four main modules as they can be seen in the conceptual diagram (Figure 4.1). **Preprocessing Module** to reading and processing videos, **Feature Extraction Module** to construct ad-hoc features, **Classification Module** to train our input data and finally **Evaluation Module** to make predictions on test data and to calculate accuracy metrics.

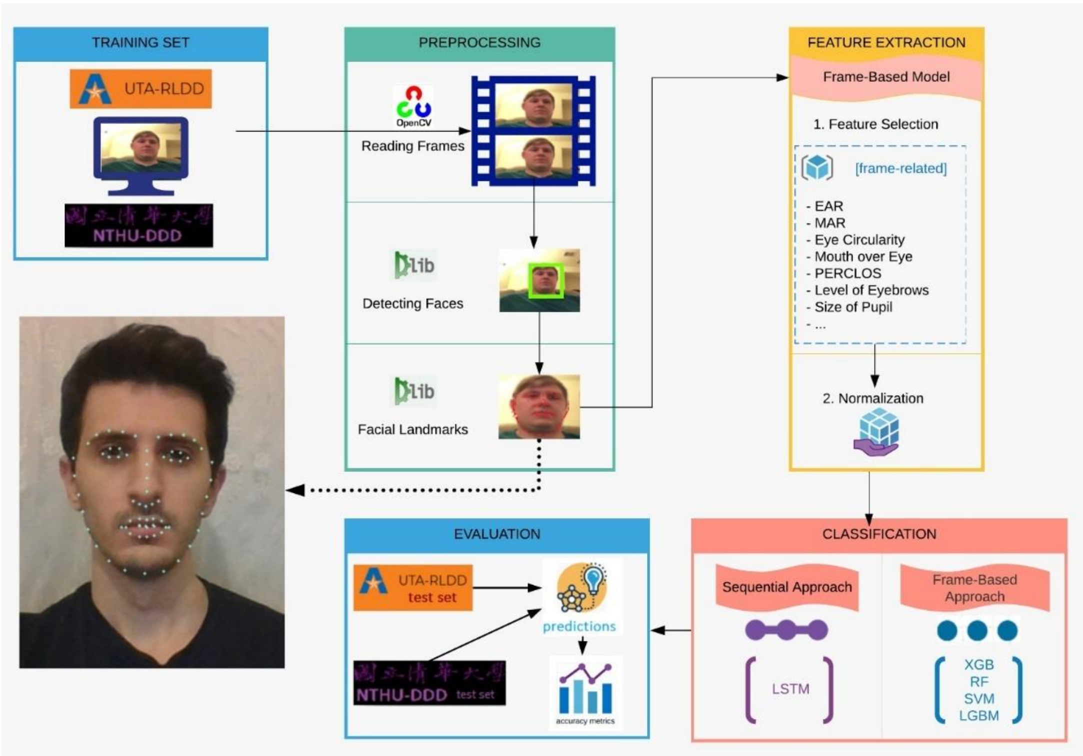

Figure 4.1 Conceptual diagram of the driver drowsiness detection system.

4.1 Preprocessing Module

Implementation of the project started with Preprocessing Module which include steps:

1. **Reading videos frame by frame**: This is done with from a dataset or with a camera in a real-time manner. For this step we used OpenCV [51] library.
2. **Detecting faces**: We used Dlib's **get_frontal_face_detector** method. It is a pretrained "Histogram of Oriented Gradients + Linear SVM" model for face detection. There is also another CNN-based method in the Dlib library but it is not suitable for using real-time purposes.

3. **Predicting facial landmarks**: We used Dlib's **shape_predictor** [53] method which is implemented by Kazemi and Sullivan. This method uses a pretrained model of an ensemble of regression trees and predicts 68 facial landmarks as can be seen from Figure 4.2. We created all hand-crafted features using this method.

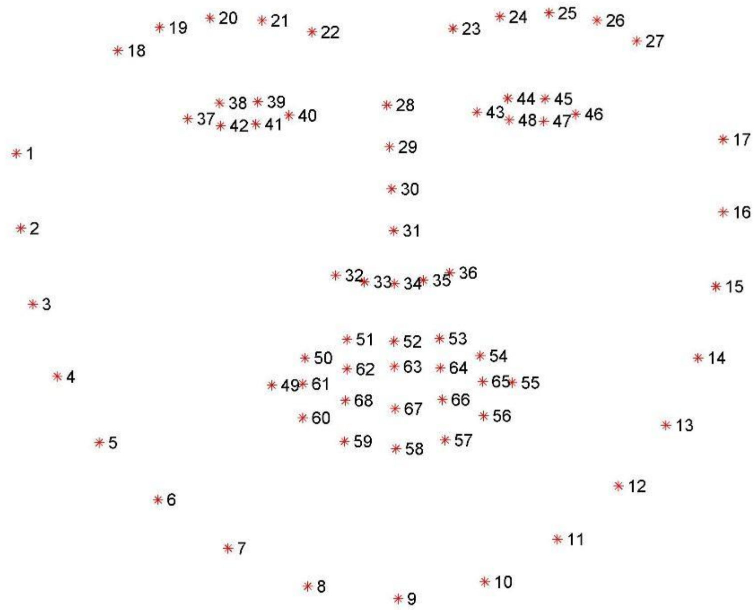

Figure 4.2 68 facial landmark coordinates of Dlib's shape_predictor method (image source: [64])

4.2 Feature Extraction Module

After the preprocessing phase, the feature extraction phase comes. In the frame-based model all features are computed for every single frame of the videos and we used that information for classification phase.

4.2.1 Feature Creation/Selection step:

This step is mainly an implementation of the work Soukupová and Chech (2016) [42] and investigates of using a simple mathematical formula for real-time purposes which is called "Eye Aspect Ratio (EAR)" (3.1) and can be extracted from eye landmark coordinates in Figure 4.2. There are also another three features (MAR, MOE, EC) defined in an online paper by Grant Zhong [49]. LEB and SOP are created originally and PERCLOS is borrowed from the literature [39].

- **Eye Aspect Ratio (EAR)**: The eye aspect ratios for all frames are calculated with the formula (Equation 4.1). The average value for both eyes is used as a feature.

$$EAR(i) = \frac{||p_{38} - p_{42}|| + ||p_{39} - p_{41}||}{2||p_{37} - p_{40}||}, \qquad (4.1)$$

In the formula here, p_{37}, …, p_{42} are 2D landmark locations of the left eye depicted in Figure 4.2 and Figure 4.3 and i is the frame index. $||p_a - p_b||$ represents the Euclidean distance between two landmark positions.

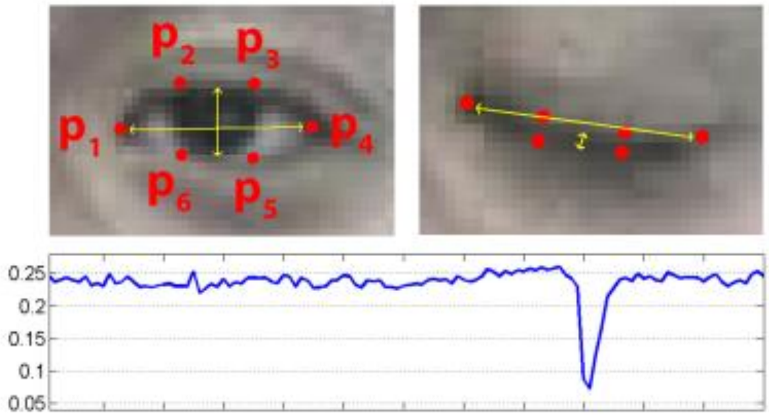

Figure 4.3 An original figure from the 2016 EAR paper [42]. Eye landmarks are used in the calculation of EAR with open/closed eye scenarios.

- **Mouth Aspect Ratio (MAR)**: The calculation of MAR is the same with EAR. It uses inner landmarks of the mouth and calculates the ratio of it. It is used for detecting yawning.

$$MAR(i) = \frac{||p_{63} - p_{67}||}{||p_{61} - p_{65}||}, \qquad (4.2)$$

In the Equation 4.2, p_{61}, …, p_{67} are 2D landmark locations of the inner mouth shape depicted in Figure 4.4 and i is the frame index. $||p_a - p_b||$ represents the Euclidean distance between two landmark positions.

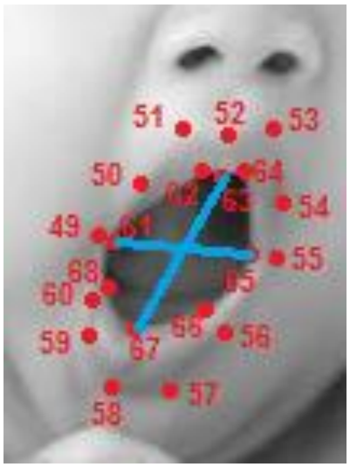

Figure 4.4 Ratio between [p63, p67] and [p61, p65] to measure Mouth Aspect Ratio (MAR).

- **Eye Circularity (EC):** It's a measure like EAR but it puts greater emphasis on the pupil area [41].

$$EC(i) = \frac{4 \times \pi \times Pupil\ Area}{(Eye\ Perimeter)^2}, \qquad (4.3)$$

$$Pupil\ Area = \left(\frac{\|p_{38} - p_{41}\|}{2}\right)^2 \times \pi, \qquad (4.4)$$

$$\begin{aligned} Eye\ Perimeter = {} & \|p_{37} - p_{38}\| + \|p_{38} - p_{39}\| + \|p_{39} - p_{40}\| + \\ & \|p_{40} - p_{41}\| + \|p_{41} - p_{42}\| + \|p_{42} - p_{37}\|, \end{aligned} \qquad (4.5)$$

In the equations above (4.3, 4.4, 4.5), p_{37}, …, p_{42} are 2D landmark locations of the left eye shape depicted in Figure 4.5 and i is the frame index. $\|p_a - p_b\|$ represents the Euclidean distance between two landmark positions. The average of both eyes is selected as a feature.

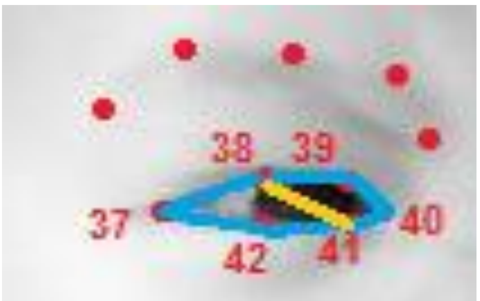

Figure 4.5 Blue lines represent Eye Perimeter and the yellow line represents Pupil Diameter which is used in the calculation of Eye Circularity (EC).

- **Mouth over Eye (MOE):** Basically EAR (Equation 4.1) over MAR (Equation 4.2) [41]. It's an additional feature which can be interpreted as true drowsiness, since some facial actions like smiling and talking may produce some fake yawning MOE values.

$$EC(i) = \frac{MAR(i)}{EAR(i)}, \qquad (4.6)$$

In addition to these 4 features, we used 3 more features that can be also produced from 68 facial landmarks:

- **Percentage of Closure (PERCLOS)**: It indicates the frequency of closed eyes up until that moment. [45] We used this feature with a small time interval, specifically for 150 frames which is equal to 5 seconds.

$$PERCLOS = \frac{count\ of\ frames\ when\ the\ eyes\ are\ closed}{total\ count\ of\ frames\ up\ until\ that\ moment} \times 100\%, \qquad (4.7)$$

- **Level of Eyebrows (LEB):** This is a good measure to detect drowsiness that calculates the average distance between the first two of inner points of eyebrows and inner corner of an eye.

$$LEB(i) = \frac{\|p_{21} - p_{40}\| + \|p_{22} - p_{40}\|}{2}, \qquad (4.8)$$

In the formula above (3.8), p_{21} and p_{22} are most inner points of the left eyebrow, also p_{40} most inner points of the left eye and represent 2D locations as depicted in Figure 4.6**.** i is frame index and $\|p_a - p_b\|$ represents euclidian distance between two landmark positions. The average of both eyes is selected as a feature.

Figure 4.6 Blue lines represent distances [p21, p40] and [p22, p40]. Average of them is calculated to measure the Level of Eyebrows (LEB).

- **Size of Pupil (SOP):** It measures the size of the pupils and not a direct relation but fluctuations of size are related to the fatigue of a subject [55]. So, the defined formula below measures the ratio of pupil diameter and eye width.

$$SOP(i) = \frac{\|p_{38} - p_{41}\|}{\|p_{37} - p_{40}\|}, \qquad (4.9)$$

In the formula above (3.9), p_{37}, …, p_{40} are 2D landmark locations of the left eye depicted in Figure 4.7 and i is the frame index. $\|p_a - p_b\|$ represents the Euclidean distance between two landmark positions. The average of both eyes is selected as a feature.

Figure 4.7 Blue line represents Eye Width [p37, p40] and orange line represents Pupil Diameter [p38, p41]. The ratio of them is called Size of Pupil (SOP).

4.2.2 Normalization step:

We applied two different normalizations to our datasets. The first one was for column-wise normalization in order to get better performance from the models. Also, the second one was subject-wise normalization. It is essential in this project because there are some people who have slanted eyes which makes EAR values different from others.

We used first 90 frames in an alert video of a specific subject to normalize features. So, this process can be formulated as,

$$normalized\ feature_{n,m} = \frac{feature_{n,m} - \mu_{n,m}}{\sigma_{n,m}}, \qquad (4.10)$$

Where, $\mu_{n,m}$ and $\sigma_{n,m}$ are mean and standard deviation of the feature n of the subject m. On the other hand, column-wise normalization is done did by scaling each feature with the corresponding mean and standard deviation for each column.

$$scaled\,feature_n = \frac{feature_n - \mu_n}{\sigma_n}, \qquad (4.11)$$

Where, μ_n and σ_n are mean and standard deviation of the feature n.

4.3 Classification Module

Up to now, we preprocessed our data and extracted hand-crafted features from them, then we came to the classification module. We have tried twelve different classifiers for frame-based models: Logistic Regression, Naive Bayes, k-Nearest Neighbour, Decision Tree, Random Forest, Support Vector Machines, Extra Trees, Extreme Gradient Boosting, Ada Boosting, Cat Boosting, Gradient Boosting and Light Gradient Boosting Machine. These models are used to calculate current drowsiness status of the driver.

Early warning is one of the four main aims. To handle this, we used sequential models such as LSTM and it's variations with different architectures. LSTM-vanilla (1-layered LSTM), LSTM-stacked (2-layered LSTM), LSTM-Bidirectional, CNN-LSTM (1 layer CNN + 1 layer LSTM) and finally Conv-LSTM.

By using these models, it's possible to train them with truth labels in two different manners:

- Ground-truth label on the drowsiness level of the subject in a frame or
- Ground-truth label on the drowsiness level of the subject in the whole video.

While NTHU-DDD provides truth labels for both of them, UTA-RLDD provides truth labels for only the second approach. So we trained our frame-based models on both datasets, by converting video labels to frame labels (by assuming same drowsiness level for all frames in RLDD videos). But we trained our time-series/sequential models only on NTHU-DDD dataset, since sequence of the same label wouldn't be meaningful in the case of RLDD videos.

4.4 Prediction and Evaluation Module

We trained our models and now it's the time for making predictions and getting results. We used two different approaches as mentioned in the previous module. For frame-based models we used F1 score as an evaluation metric defined in chapter 3.1.3 Comparison Metrics and for sequential models we used MSE (Mean Squared Error) to evaluate the results (Equation 3.1).

4.5 Real-Time Application

After constructing models by using the pipeline defined above, we developed an application to demonstrate our work in real-time.

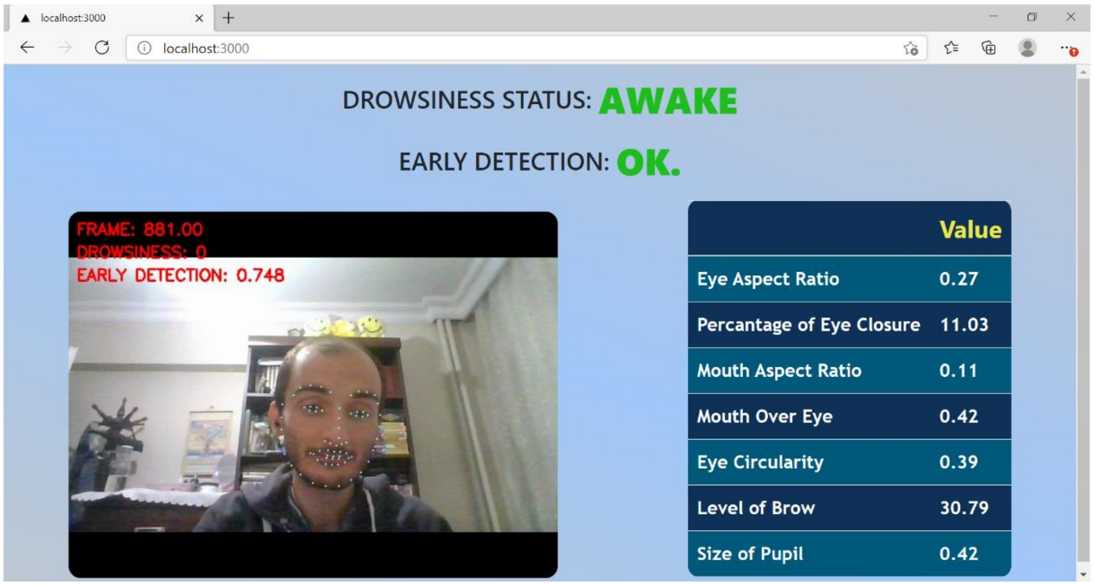

Figure 4.8 Graphical interface of the real-time application that runs on computer browser.

This application consists of two parts. The first part is the backend part which is written in Python's Flask Framework [61]. The application captures the driver face through the webcam and process camera inputs frame by frame. Then, the camera output is fed in to the frontend through Web Sockets to show the image of the driver in real-time. For each frame, application extracts the features. Then the application feeds trained models (sequential and frame-based) with these features and models predict two outputs. These outputs are drowsiness status and early detection. As default, XGBoost is chosen for predicting drowsiness status and LSTM-vanilla is chosen for predicting early detection from our pool of trained and optimized models. If the application runs on a system with low configuration, Decision Tree or simple thresholding may be optionally selected instead of XGBoost.

After that, features and the outputs are sent to the frontend part of the application through Web Sockets. The second part of the project is the frontend part. We developed this part with NextJS [62] which is a framework for ReactJS [63]. In the frontend, the application simply receives the backend outputs and prints them. On the top of the screen the application prints the prediction results. At the left side, it prints the camera in real-time and on the right side, it prints the features in real-time.

5. EXPERIMENTAL STUDY

In this section we will explain the environment and results for our experiments.

5.1 Experimental Setup

Preliminary experiments are completed in online Kaggle notebook environment [56] and we are planning to continue to work with it. Some specifications of free Kaggle notebooks are:

- CPU single core hyper threaded (1 core, 2 threads) Xeon Processors @2.3Ghz, 46MB Cache
- CPU-only notebook gives 4 cores CPU + 16 GB ram,
- GPU notebook gives Nvidia Tesla P100 16gb VRAM + 2 cores CPU + 13 GB ram,
- TPU notebook gives TPU v3-8.

5.2 Experimental Results and Discussions

Experiments will be run for 2 approaches: frame-based models and sequential models as explained in Sections 4.3 and 4.4. There will be preprocessing, feature extraction, classification and evaluation steps for both of models. Main difference is usage of input data. First approach uses frame-based features and labels on training and testing phases. On the other hand, second approach processes sequences of input data. Another difference is the dataset we used. We used only NTHU for sequential models since RLDD doesn't provide frame labels and it's not meaningful to use the same target value in the whole sequence.

5.2.1 Results for Frame-Based Models

After reading frames by OpenCV and Dlib then extracting features from landmark positions, just like explained in Section 4.1 and 4.2, these features are handed to twelve classifiers: All the classifiers available on Scikit-Learn plus XGBoost and LightGBM. After train-test split by 80%-20% and optimizing hyper-parameters on 10-fold cross-validation, we get F1 scores on three datasets: NTHU, RLDD and merged version of both (see Table 5.1). For NTHU dataset Extra Trees classifier has the best score (84%), while XGBoost takes top for RLDD and merged datasets (89% and 86%) surpassing SOTA of 70% explained in Section 2.4.2.

Table 5.1 F-1 scores of frame-based models on NTHU, RLDD and merged datasets.

Dataset:	NTHU	RLDD		MERGED	
Label:	Drowsy	Half Awake	Drowsy	Half Awake	Drowsy
Simple thresholding (EAR<23)	0.66	-	0.52	-	0.55
Support-Vector Machine	0.46	0.41	0.30	0.38	0.33
Naive-Bayes	0.51	0.30	0.45	0.32	0.45
Logistic Regression	0.75	0.32	0.51	0.20	0.58
AdaBoost	0.74	0.51	0.63	0.44	0.63
Gradient Boosting Classifier	0.78	0.58	0.70	0.50	0.68
Random Forest	0.82	0.63	0.74	0.71	0.71
LightGBM	0.80	0.73	0.79	0.75	0.75
Decision Tree	0.77	0.82	0.84	0.80	0.80
K-Nearest Neighbor	0.78	0.79	0.84	0.79	0.79
Extra Trees	0.84	0.82	0.85	0.81	0.81
Bagging Classifier	0.83	0.86	0.88	0.84	0.84
XGBoost	0.80	0.87	0.89	0.86	0.86

5.2.2 Results for Sequential Models

All of the preprocessing steps stays same for sequential methods but before training phase, we had to reconstruct our input data into sequences with the window size of 150 frames. These sequences are used to predict a label which indicates the average drowsiness of the next 60 frames (equivalent to 2 seconds). 5 variants of LSTM available in Keras library are trained and evaluated on NTHU dataset (see Table 5.2).

Table 5.2 MSE and F-1 scores of LSTM models on NTHU dataset.

	MSE	F-1
Vanilla-LSTM	0.93	0.63
Stacked-LSTM	0.96	0.71
Bi-LSTM	0.95	0.71
CNN-LSTM	0.98	0.61
Conv-LSTM	0.99	0.62

Main evaluation metric is Mean Squared Error for this approach but F-1 scores are also provided to compare models to first approach. To achieve F-1 scores output sequence is redesigned, very first frame is selected as label instead of a sequence of 60 frames. It's arguable that this way of scoring is the right way since when we change the structure of input data model weights are also change. Calculated MSE and F-1 scores may not belong to the same model but they give a clue when we compare them to frame-based models. For MSE, Vanilla-LSTM takes top with a 0.93 score in the range of [0,4] so it was selected as best model in the second set of experiment.

6. CONCLUSION AND FUTURE WORK

This project examines the current status of the driver drowsiness detection literature by borrowing a successful pipeline already used in wide; from capturing landmark positions, formulated features to machine learning classifiers. But it also has original points, additional features such as level-of-eyebrows (LEB), size-of-pupils (SOP) defined in Section 4.2, early detection mechanism by using sequential approach defined in the Section 5.2.2 and using it along with current drowsiness status in a real-time application to produce two kinds of driving alert.

Consequently, the project reached to the point where it reaches to predefined goals. After optimizing machine learning models with series of experiments, we achieved high accuracy on benchmark datasets, 89% F-1 score on RLDD which is 19% higher than state-of-art. Also, we attained early detection of 2 seconds by using sequential models. Another goal was personal adaptivity and subject-wise normalization provides it. Lastly, fourth goal was real-time performance so an application with simple GUI that runs on a laptop browser is developed.

As future work, raw input frames can be fed in to CNN or 3D-CNN architectures alternatively. Transfer learning may be another path to be followed too, since pretrained models are very successful on computer vision tasks. Besides, there is a possibility of using extra features like head position and forehead wrinkles by using OpenFace [65] library.

REFERENCES

[1] R. Fu, H. Wang, W. Zhao, “Dynamic Driver Fatigue Detection Using Hidden Markov Model in Real Driving Condition”, Expert Systems with Applications, vol.63, pp.397-411, 2016.

[2] A. Čolić, O. Marques, B. Furht, Driver Drowsiness Detection Systems and Solutions. Cham Heidelberg New York Dordrecht London: Springer, 2014.

[3] Transport Accident Commission, Avoiding Driver Fatigue. [Online]. Available: http://www.tac.vic.gov.au/road-safety/safe-driving/tips-and-tools/fighting-fatigue (Date of Access 20 / 04 /2020)

[4] A. M. Williamson, A. M. Feyer, “Moderate Sleep Deprivation Produces Impairments in Cognitive and Motor Performance Equivalent to Legally Prescribed Levels of Alcohol Intoxication”, Occupational and Environmental Medicine, vol.57 no.10, pp.649-655, 2000.

[5] D. Dawson, K. Reid, “Fatigue, Alcohol and Performance Impairment”, Nature, vol.388 no.6639, pp.235, 1997.

[6] N. Lamond, D. Dawson, “Quantifying the Performance Impairment Associated with Fatigue”, Journal of Sleep Research, vol.8, no.4, pp.255-262, 1999.

[7] Centers for Disease Control and Prevention, Drowsy Driving: Asleep at the Wheel. [Online]. Available: https://www.cdc.gov/features/dsdrowsydriving/index.html (Date of Access 20 / 04 /2020)

[8] V. Triyanti, H. Iridiastadi, “Challenges In Detecting Drowsiness Based On Driver’s Behavior”, IOP Conference Series: Materials Science and Engineering, vol. 277, no. 1, p. 01204, 2017.

[9] UTA-RLDD, UTA Real-Life Drowsiness Dataset. [Online]. Available: https://sites.google.com/view/utarldd/home (Date of Access 20 / 04 /2020)

[10] Computer Vision Lab, National Tsuing Hua University, Driver Drowsiness Detection Dataset. [Online]. Available: http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/ (Date of Access 20 / 04 /2020)

[11] Association for Safe International Road Travel, Road Safety Facts. [Online]. Available: https://www.asirt.org/safe-travel/road-safety-facts (Date of Access 20 / 04 /2020)

[12] R. Ghoddoosian, M. Galib and V. Athitsos, A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection, in: IEEE/CVF Conference on

Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
[13] Zer Customs, Volvo Driver Alert Control and Lane Departure Warning. [Online]. Available: http://www.zercustoms.com/news/Volvo-Driver-Alert-Control-and-Lane-Departure-Warning.html (Date of Access 20 / 04 /2020)
[14] Motor 1, Toyota Redesigns Crown Introduces Hybrid Model. [Online]. Available: https://www.motor1.com/news/2106/toyota-redesigns-crown-introduces-hybrid-model/ (Date of Access 20 / 04 /2020)
[15] Daimler, Attention Assist Drowsiness Detection System Warns Drivers to Prevent Them Falling Asleep. [Online]. Available: https://media.daimler.com/marsMediaSite/en/instance/ko/ATTENTION-ASSIST-Drowsiness-detection-system-warns-drivers-to-prevent-them-falling-asleep-momentarily.xhtml?oid=9361586 (Date of Access 20 / 04 /2020)
[16] Volvo Cars, Driver Alert Control (DAC). [Online]. Available: https://www.volvocars.com/en-th/support/manuals/v60/2017-early/driver-support/driver-alert-system/driver-alert-control-dac (Date of Access 20 / 04 /2020)
[17] BMW, The Main Driver Assistance Systems. [Online]. Available: https://www.bmw.com/en/innovation/the-main-driver-assistance-systems.html (Date of Access 20 / 04 /2020)
[18] NISSAN USA, Drowsy Driver Attention Alert Car Feature. [Online]. Available: https://www.nissanusa.com/experience-nissan/news-and-events/drowsy-driver-attention-alert-car-feature.html (Date of Access 20 / 04 /2020)
[19] W. Han, Y. Yang, G. Bin Huang, O. Sourina, F. Klanner, and C. Denk, "Driver Drowsiness Detection Based on Novel Eye Openness Recognition Method and Unsupervised Feature Learning," Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015, no. September, pp. 1470–1475, 2016.
[20] M. Patel, S. K. L. Lal, D. Kavanagh, and P. Rossiter, "Applying neural network analysis on heart rate variability data to assess driver fatigue", Expert Syst. Appl., 38(6):7235–7242, June 2011
[21] S. Hu and G. Zheng, "Driver drowsiness detection with eyelid related parameters by Support Vector Machine", Expert Syst. Appl., 36(4):7651–7658, May 2009.
[22] D. McDonald, C. Schwarz, J. D. Lee, and T. L. Brown, "RealTime Detection of Drowsiness Related Lane Departures Using Steering Wheel Angle," Proc. Hum. Factors Ergon. Soc. Annu. Meet., vol. 56, no. 1, pp. 2201–2205, 2012.
[23] A. Mittal, K. Kumar, S. Dhamija, and M. Kaur, "Head movementbased driver

drowsiness detection: A review of state-of-art techniques," Proc. 2nd IEEE Int. Conf. Eng. Technol. ICETECH 2016, pp. 903–908, 2016.

[24] A. Ramos, J. Erandio, E. Enteria, N. Carmen, L. Enriquez and D. Mangilaya, "Driver Drowsiness Detection Based on Eye Movement and Yawning Using Facial Landmark Analysis", International journal of simulation: systems, science & technology, 10.5013/IJSSST.a.20.S2.37., 2019.

[25] T. Nakamura, A. Maejima, and S. Morishima, "Detection of driver's drowsy facial expression," Proc. - 2nd IAPR Asian Conf. Pattern Recognition, ACPR 2013, pp. 749–753, 2013.

[26] P. Viola and M. Jones, "Robust real-time object detection", *International Journal of Computer Vision*, 2001

[27] T. Ojala, M. Pietikainen and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", *Proceedings of 12th International Conference on Pattern Recognition*, Jerusalem, Israel, 1994, pp. 582-585 vol.1.

[28] Li, Kangning et al. "Accurate Fatigue Detection Based on Multiple Facial Morphological Features." *J. Sensors* 2019: 7934516:1-7934516:10, 2019.

[29] E. Tadesse, W. Sheng and M. Liu, "Driver drowsiness detection through HMM based dynamic modeling," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 4003-4008.

[30] A. Lenskiy and J.-S. Lee, "Driver's eye blinking detection using novel color and texture segmentation algorithms", *International Journal of Control, Automation and Systems*, 10(2):317–327, 2012.

[31] Tech-Quantum, Various Techniques to Detect and Describe Features in an Image Part-1. [Online]. Available: https://www.tech-quantum.com/various-techniques-to-detect-and-describe-features-in-an-image-part-1/ (Date of Access 20 / 04 /2020)

[32] Tech-Quantum, Various Techniques to Detect and Describe Features in an Image Part-2. [Online]. Available: https://www.tech-quantum.com/various-techniques-to-detect-and-describe-features-in-an-image-part-2/ (Date of Access 20 / 04 /2020)

[33] Naz, Saima et al. "Driver Fatigue Detection using Mean Intensity, SVM, and SIFT." *IJIMAI* 5: 86-93, 2019.

[34] R. Prem Kumar, M. Sangeeth, K.S. Vaidhyanathan, A. Pandian. "TRAFFIC SIGN AND DROWSINESS DETECTION USING OPEN-CV", International Research Journal of Engineering and Technology (IRJET) vol. 06 issue 03, 2019.

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1.
[36] DLib, Face Detector. [Online]. Available: http://dlib.net/face_detector.py.html (Date of Access 20 / 04 /2020)
[37] Wierwille, Walter W. et al. "RESEARCH ON VEHICLE-BASED DRIVER STATUS/PERFORMANCE MONITORING; DEVELOPMENT, VALIDATION, AND REFINEMENT OF ALGORITHMS FOR DETECTION OF DRIVER DROWSINESS. FINAL REPORT", 1994.
[38] A. Dasgupta, A. George, S. L. Happy and A. Routray, "A Vision-Based System for Monitoring the Loss of Attention in Automotive Drivers", in: *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1825-1838, Dec. 2013.
[39] A. Liu, Z. Li, L. Wang and Y. Zhao, "A practical driver fatigue detection algorithm based on eye state," *2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, Shanghai, 2010, pp. 235-238.
[40] Q. Cheng, W. Wang, X. Jiang, S. Hou and Y. Qin, "Assessment of Driver Mental Fatigue Using Facial Landmarks", in *IEEE Access*, vol. 7, pp. 150423-150434, 2019.
[41] Zhong, G., Ying, R., Wang, H., Siddiqui, A., & Choudhary, G., Drowsiness Detection with Machine Learning. [Online]. Available: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a (Date of Access 20 / 04 /2020)
[42] T. Soukupová and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks", 2016.
[43] Ching-Hua Weng, Ying-Hsiu Lai and Shang-Hong Lai, "Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network", In Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video, Taipei, Taiwan, Nov. 2016
[44] I. H. Choi, C. H. Jeong, and Y. G. Kim, "Tracking a driver's face against extreme head poses and inference of drowsiness using a hidden Markov model", Appl. Sci., vol. 6, no. 5, 2016.
[45] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", Proc. fifth Annu. Work. Comput. Learn. theory, pp. 144–152, 1992.
[46] Blue and Green Tomorrow, How Traffic Accidents Harm the Environment. [Online]. Available: https://blueandgreentomorrow.com/environment/traffic-accidents-

harm-environment (Date of Access 20 / 04 /2020)
[47] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review, 65*(6), 386–408, 1958.
[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., pp. 1–9, 2012.
[49] M. Ngxande, J-R. Tapamo, M. Burke, Driver Drowsiness Detection Using Behavioral Measures and Machine Learning Techniques: A Review of State-Of-Art Techniques, in: Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2017.
[50] Hu, Y., Lu, M., Xie, C., & Lu, X, Driver Drowsiness Recognition via 3D Conditional GAN and Two-level Attention Bi-LSTM. IEEE Transactions on Circuits and Systems for Video Technology, (2019)
[51] PyPi, opencv-python. [Online]. Available: https://pypi.org/project/opencv-python/ (Date of Access 20 / 04 /2020)
[52] Dlib, Classes. [Online]. Available: http://dlib.net/python/index.html#dlib.get_frontal_face_detector (Date of Access 20 / 04 /2020)
[53] Dlib, Classes. [Online]. Available: http://dlib.net/python/index.html#dlib.shape_predictor (Date of Access 20 / 04 /2020)
[54] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees", *2014 IEEE Conference on Computer Vision and Pattern Recognition* 1867-1874, 2014.
[55] B. Wilhelm, A. Widmann, W, Durst, C. Heine and G. Otto, "Objective and quantitative analysis of daytime sleepiness in physicians after night duties", International journal of psychophysiology: official journal of the International Organization of Psychophysiology. 72. 307-13. 10.1016/j.ijpsycho.2009.01.008., 2009.
[56] Kaggle, Notebooks. [Online]. Available: https://www.kaggle.com/hakkoz/notebooks (Date of Access 20 / 04 /2020)
[57] Early Drowsiness Detection, Repository. [Online]. Available: https://github.com/rezaghoddoosian/Early-Drowsiness-Detection (Date of Access 20 / 04 /2020)
[58] I2 Tutorials, Deep Dive into Bidirectional LSTM. [Online]. Available: https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/ (Date of Access 10 / 02 /2021)

[59] Scikit Learn, Home Page. [Online]. Available: https://scikit-learn.org/stable/ (Date of Access 20 / 04 /2020)
[60] Keras, Home Page. [Online]. Available: https://keras.io/ (Date of Access 20 / 04 /2020)
[61] Flask, Home Page. [Online]. Available: https://flask.palletsprojects.com/en/1.1.x/ (Date of Access 20 / 04 /2020)
[62] NextJS, The React Framework for Production. [Online]. Available: https://nextjs.org/ (Date of Access 20 / 04 /2020)
[63] React, Home Page. [Online]. Available: https://tr.reactjs.org/ (Date of Access 20 / 04 /2020)
[64] Pyimagesearch, Facial landmarks with dlib, OpenCV, and Python. [Online]. Available: https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/ (Date of Access 20 / 04 /2020
[65] Open Face, OpenFace 2.2.0: a facial behavior analysis toolkit. [Online]. Available: https://github.com/TadasBaltrusaitis/OpenFace (Date of Access 20 / 04 /2020)
[66] Github, Aysenuryilmazz/Driver_Drowsiness_Detection. [Online]. Available: https://github.com/Aysenuryilmazz/Driver_Drowsiness_Detection (Date of Access 12 / 02 /2021)
[67] Kaggle, Notebooks. [Online]. Available:
https://www.kaggle.com/hakkoz/notebooks (Date of Access 20 / 04 /2020).
[68] Trello, Home Page. [Online]. Available: https://trello.com/ (Date of Access 20 / 04 /2020)
[69] Dlib, Home Page. [Online]. Available: http://dlib.net/ (Date of Access 20 / 04 /2020)