Marmara University Faculty of Engineering

Computer Engineering Department

CSE4097 - Engineering Project I

“DRIVER DROWSINESS DETECTION”

Analysis and Design Document

Students:	Ayşenur	YILMAZ	150114002
	Mahmut	AKTAŞ	150115010
	Mustafa Abdullah	HAKKOZ	150117509

Supervisor: Prof. Dr. Ciğdem Eroğlu Erdem

09/07/2020

- 06.07.2020 -

1. Introduction

1.1 Problem Description and Motivation

Driver drowsiness is a continual risk for the drivers and road safety which is one of the significant reasons for the road accidents. According to the statistics, highway road crashes hold 11.09% of the total number of accidents. There are numerous other reasons for the road accidents aside from driver fatigue, such as excessive speed, unsafe lane changes, illegal overtaking, parking, overloading etc. Driver drowsiness can be condemned for a major participant in road crashes, which has a 2%-23% estimated share on the pie. These shares are still conservative estimations since it is a difficult job to estimate the exact number of the fatigue-related accidents [1].

Light and dark cycles have an impact on sleepiness and wakefulness. Hence, humans 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].

In this context, most fatigue-related accidents arise during normal sleeping hours, and the more serious the crash, there is more probability that the driver was drowsy. Drowsiness almost has a one-third share of the causes of the single-vehicle accidents in the rural areas.

It is a frequent assumption that drowsiness is only a problem for the long-distance drivers, nonetheless it is a problem for the short-distance drivers also. Driving usually is not the major reason for the fatigue. Drivers generally are already tired when they get into the car due to long-hours of working, sleep apnea, lack of sleep or shift work.

There are various causes for the 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 know that, people can't fight with sleep [3].

Drowsiness has a huge effect on the drivers' wheel control and driving capability and puts them at risk of their safe travel. 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]. Having zero sleep for 24 hours has the same effect as BAC of 0.1 which is double of the legal limit [5,6].

Driving while tired or drowsy can result in:

- 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 dire consequences

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

- Frequently 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]

While these ways can help avoid driver drowsiness, we need to make sure that drivers are awake at the wheel. In this project, we are aiming to design a system that checks the driver's facial behaviors, mainly eyes and mouth, and to be able to detect the drowsiness status of the driver in real-time. For the sake of achieving this goal, firstly we will detect the driver's face and extract some facial features. After the extraction of these features, we will train a model with drowsy non- drowsy test data. By using this model, the application will decide if the driver is drowsy or not.

Drowsiness detection is a difficult task also. Building a system for such an important and vital subject can be challenging. There are a number of difficulties for drowsiness detection. For example, collecting real-life data is very hard since it should be in a real car in traffic with really drowsy drivers. Generally, driver drowsiness datasets divide into two kinds. First kind consists of subjects that are acting as a drowsy person. The second kind consists of subjects that are really drowsy on the videos. Although the second one gives more accurate results, both of these kinds are captured in laboratory environments rather than a real car in traffic. The other difficulty is individual differences. Everybody has his own characteristics, including an eye and face. For instance, there is a person with round eyes versus slanted eyes, long versus short eye lace, and also expressive versus non-expressive person. These differences make model formulation could be more complicated [8]. Last but not least difficulty for the drowsiness detection is for devices that have to capture the face and eye appearance, any condition that can cover face is undesirable. For instance, when the driver does not face towards the front, when lighting conditions are lacking, when the driver wears accessories such as hats and glasses. The camera will not be able to capture the characteristics of the eyes and face.

1.2 Scope of the Project

The main aim of this project is to build a system that detects the drowsiness of the driver and gives 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 consists of three phases. In the initial phase, the development of the blink detection method and generating blink detection features are included. In the second phase, development for frame-based feature extraction with normalization is residing. In the final phase, determining the appropriate classifier method which includes classical machine learning techniques like **SVM**, **k-NN**, HMM and deep learning brand new techniques such as **CNN**, **LSTM**.

The feature extraction phase is proceeded with using two different techniques; Mustafa will be implementing the Blink-based Model and Ayşenur and Mahmut will be implementing feature extraction by frame-based.

In this project, **UTA Real Life Drowsiness Dataset** [9] and **NTHU-DDD** [10] is used for both training and test data. Also, blinks methods are developed mainly based on **Eyeblink8** and **TalkingFace** datasets from **blinkmatters.com** [11].

Since this work is a research-based project, an implementation on a computer system is a primary objective of ours.

1.2.1 Aims of the Project

There are four main goals defined for our project.

- **Detecting Driver Drowsiness with High Accuracy**

The main aim of this project is to detect driver drowsiness. The system will be able to detect driver drowsiness with high accuracy with only using the facial signs and a low-cost camera. This project would be able to attain at least a 65% accuracy rate which is higher by 3.6% as compared to the state-of-art results on the **UTA-RLDD** dataset [12].

- **Early Detection**

The other essential aim for this project is predicting driver drowsiness before the driver falls asleep. Drivers always have to be warned before they fall asleep while driving. This project will be able to extract some facial features that are evaluated from a certain set of landmarks on the human face. Then these features will be interpreted to be able to say if the driver is going to fall asleep. So reading signals as a time-series and predicting at least a couple of seconds forward to prevent a possible accidents, is one of the main goals of the project. More exactly we will try to predict 3 seconds ahaed and this will allow 1 second reaction time to driver considering 2 seconds delay of real-time performance, in bad scenarios.

- **Real-Time Performance**

Another important and crucial aim for this project is working in real-time. The system will always be working until it is closed manually. During it's working time, this project aims to work fast enough to calculate the necessary features and interpret them in real-time. The maximum delay for this system must be 2 seconds for calculating the drowsiness.

- **Adaptivity to the Subject**

The final aim of this project is to build a system that will be able to detect the doziness of all people from different ethnicities and personal characteristics. Since this project will be detecting drowsiness based on facial features of the human face, it will be able to adjust itself and give accurate results with different skin color, eye shape, etc. The project will be subjective for each person by using their facial features' standards. In order to have a consistent system, accuracy levels should not differ much for different racial characteristics.

1.2.2 Constraints

In this project, there are some constraints depending on both the dataset and the technical methods. The following cases are out of the scope of this project:

If there is

- Insufficient illumination of the face according to time of the day,
- If the driver is wearing sunglasses or a hat and may have facial hair,
- There might be obstacles in front of the driver’s eyes e.g. the driver’s hand.

1.2.3 Assumptions

There are some assumptions that must be taken into consideration in order to make our system to work with a desired result, such as:

- It is assumed that all datasets that are used in this project are correctly annotated.
- The images are collected with sufficient illumination of the face.
- It is assumed that the camera is directly placed in front of the driver and nearly has one arm length distance.
- It is assumed that the driver’s head is up and his face fits in the camera.
- It is assumed that annotations provided by datasets for testing are ground-truth annotations.

1.3 Definitions, Acronyms, and Abbreviations

DDD:	Driver Drowsiness Detection
BAC:	Blood Alcohol Concentration
SVM:	Support Vector Machine
k-NN:	K-Nearest Neighbors
CNN:	Convolutional Neural Network
LSTM:	Long Short-Term Memory
UTA:	The University of Texas at Arlington
RLDD:	Real-Life Drowsiness Dataset
UTA-RLDD:	The University of Texas at Arlington Real-Life Drowsiness Dataset
NTHU:	National Tsing Hua University

NTHU-DDD:	National Tsing Hua University Drowsy Driver Detection
ECG:	Electrocardiogram
EEG:	Electroencephalogram
EOG:	Electrooculogram
LBP:	Local Binary Patterns
SIFT:	Scalar-Invariant Feature Transform
SURF:	Speeded-Up Robust Features
BRIEF:	Binary Robust Independent Elementary Features
ORB:	Oriented Fast and Rotated BRIEF
DoG:	Difference of Gaussian
HOG:	Histogram of Oriented Gradient
PERCLOS:	Percentage of Eye Closure
MAR:	Mouth Aspect Ratio
EAR:	Eye Aspect Ratio
MOE:	Mouth Over Eye
EC:	Eye Circularity
SOP:	Size of Pupil
LEB:	Level of Eyebrows
HMM:	Hidden Markov Model
AdaBoost:	Adaptive Boosting
XGBoost:	Extreme Gradient Boosting
NB:	Naïve Bayes
RNN:	Recurrent Neural Networks
ML:	Machine Learning
LSTM-HM:	Hierarchical Multiscale Recurrent Neural Networks
GAN:	Generative Adversarial Networks

Bi-LSTM:	Bidirectional Long Short-Term Memory
EWMA:	Exponentially Weighted Moving Average
RMSE:	Root Mean Square Error
RGB:	Red Green Blue
ARIMA:	Autoregressive integrated moving average
PACF:	Partial Auto Correlation Function

2. Related Work

In this section, we will explain literature’s position under models and techniques topic, baseline results under state-of-art results topic and the main papers that we follow up in our research under selected works topic.

2.1 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 a proper **DDD** (Driver Drowsiness Detection) system [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 in algebraic and algorithmic processes. Some of the widely used face detection methods are; **Viola-Jones algorithm** [26, 24] which uses **light**

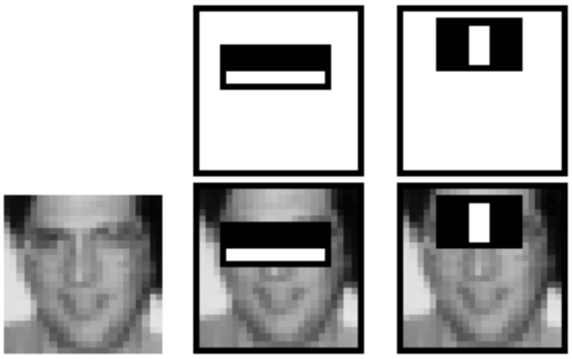


***Figure 1**: An original figure from the 2001 Viola-Jones paper [26]. Detecting eyes and nose with two significant Haar-like features.*

intensity difference on eye-nose regions (see Figure 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), **Gabor Filters** to enhance topological structure of human face, see the changes occurring and detect faces this way [29],

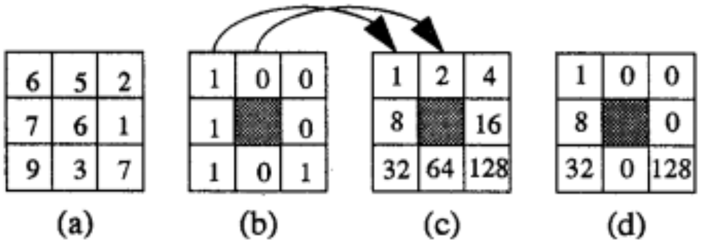


***Figure 2**: An original figure from the 1994 LBP paper [27]. It uses center as threshold, marks neighborhood and converts the whole square to binary code.*

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 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.

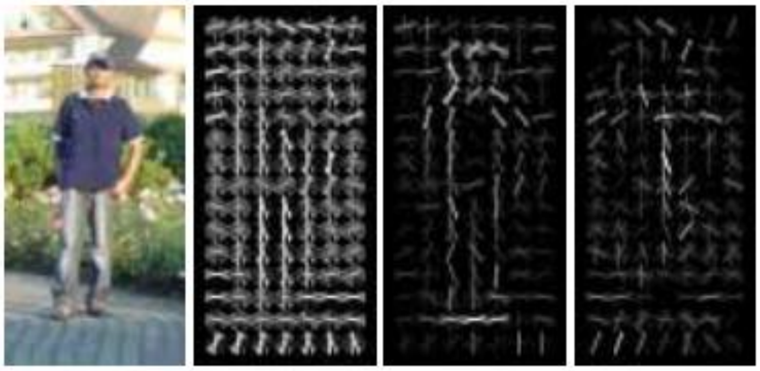


***Figure 3:** An original figure from the 2005 HOG paper [35] showing orientations of gradients.*

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 3.2.2 and 3.2.3 in detail.

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 4 and 5).

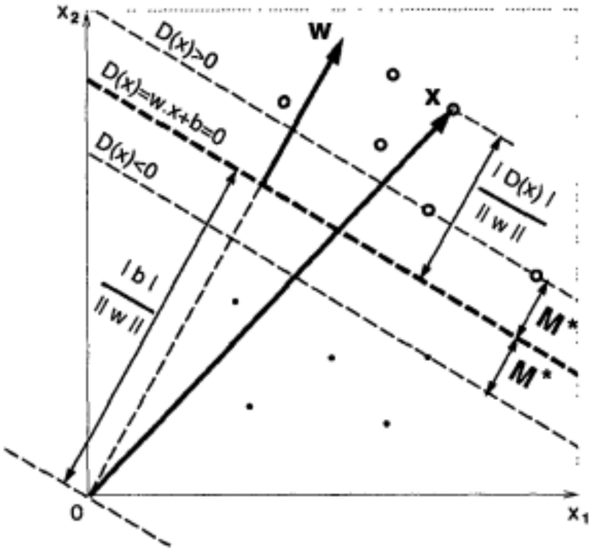


***Figure 4**: An original figure from the 1992 SVM paper [45] showing decision boundary with maximum margins between two classes.*

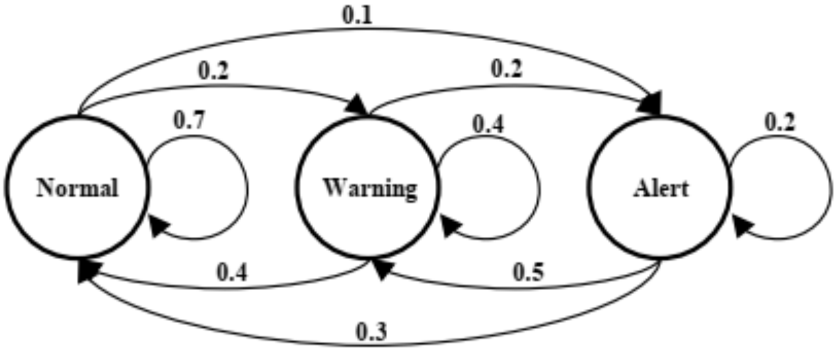


***Figure 5**: An original figure from the 2016 paper [44] showing probabilistic transitions between three possible states: Normal, Warning, Alert.*

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 6). There's also an increasing popularity of **RNN**s (Recurrent Neural 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].

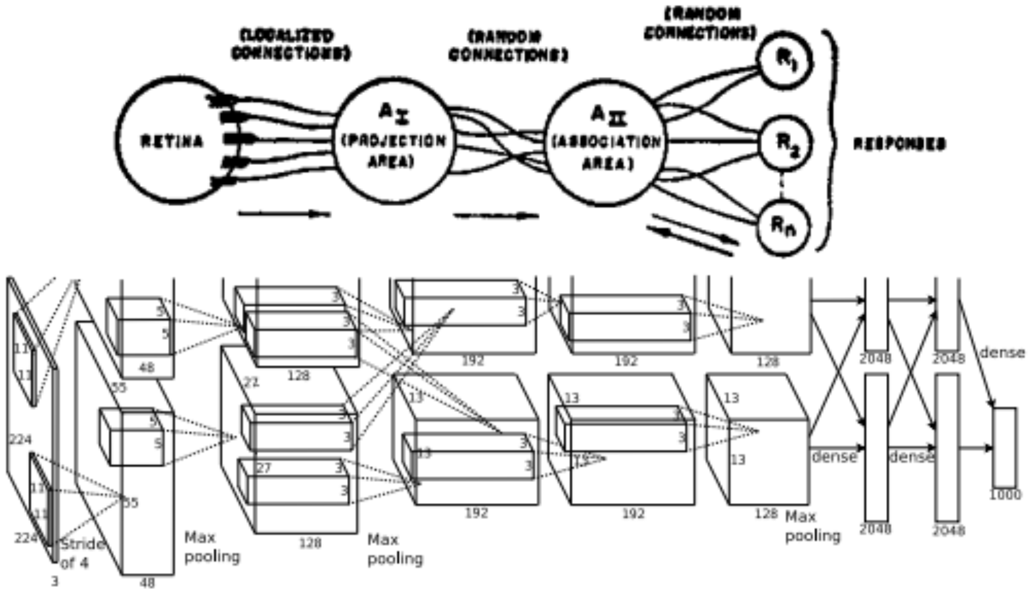


__Figure 6__: First is original figure from the 1958 Perceptron paper [47] and second is ImageNet2010 contest winner AlexNet, a CNN implementation [48].

In our project, we used **simple thresholding** and some conventional machine learning tools such as **KNN**, **Decision Tree**, **Random Forest** and **NB** yet and we are planning to use sequential models like **HMM** and **LSTM**, along with other deep learning concepts such as **CNN** and **Transfer Learning** [41, 46] in second term.

2.2 State-of-art Results

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

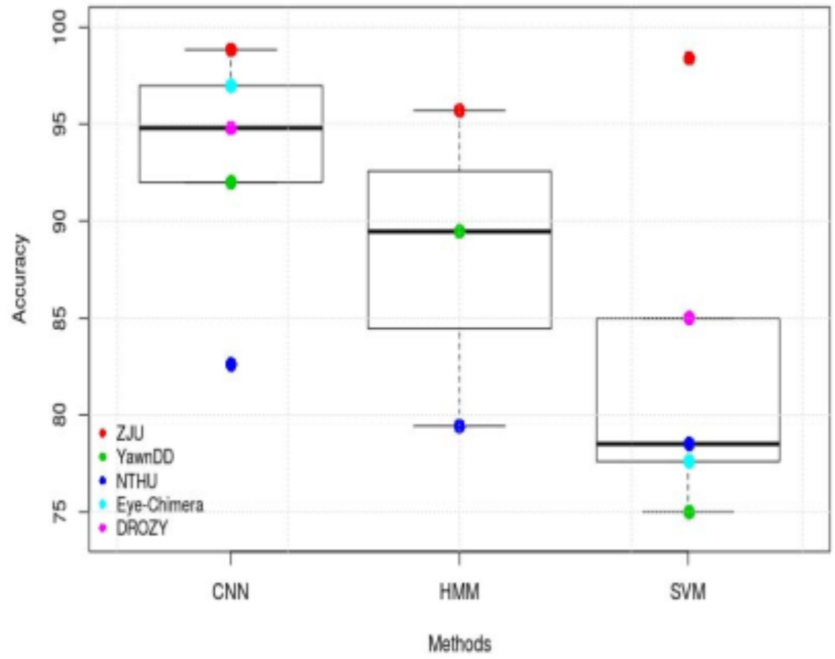
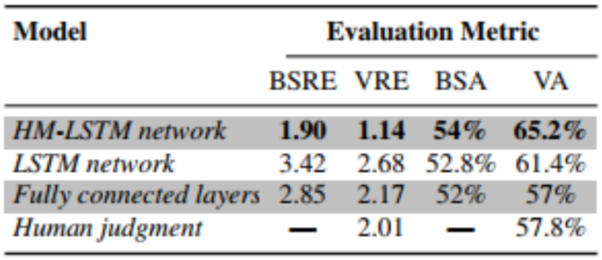


__Figure 7__: An original figure from the 2017 survey paper [49] showing boxplots for comparing ML methods on most common datasets by the date.

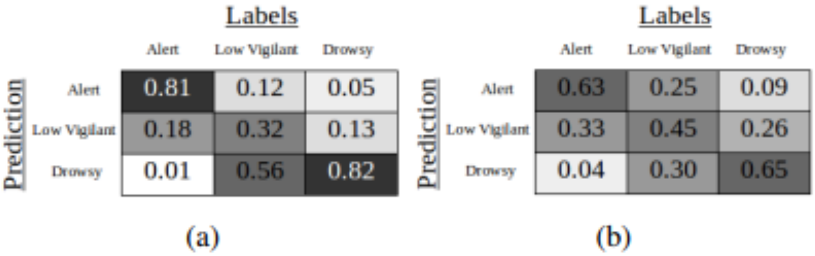
using a meta-analysis approach (see Figure 7) which indicates **CNN** as a most successful classifier against **SVM** and **HMM** [49]. Yet, it's still not enough to predicate state-of-art results because some of the databases are not open to public access and open ones mostly consist of actor subjects or non-realistic environments and these are the main reasons shadowing accountability of works in the literature.

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%** 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 8 and 9.



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



(a)

Prediction \ Labels	Alert	Low Vigilant	Drowsy
Alert	0.81	0.12	0.05
Low Vigilant	0.18	0.32	0.13
Drowsy	0.01	0.56	0.82

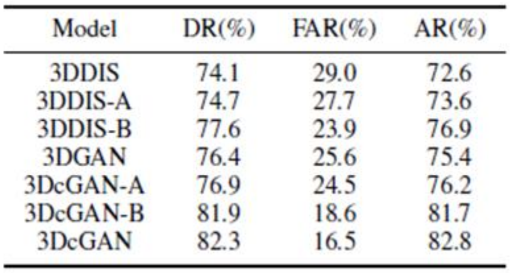
(b)

Prediction \ Labels	Alert	Low Vigilant	Drowsy
Alert	0.63	0.25	0.09
Low Vigilant	0.33	0.45	0.26
Drowsy	0.04	0.30	0.65

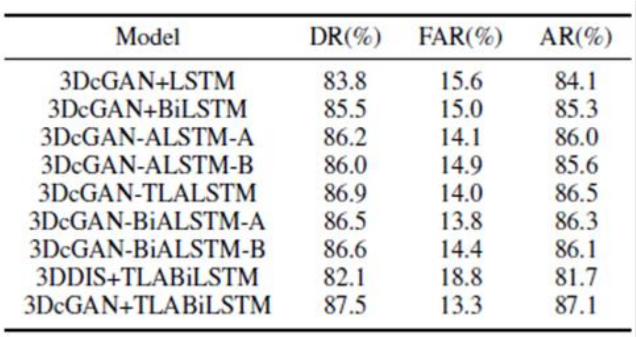
***Figure 9**: An original figure from the 2019 LSTM paper [12] showing confusion matrix of LSTM model (a) and human judgement line (b).*

We are also planning to work 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 Table 1 and Table 2. Hence, these can be used for future researchers to compare their works. Quantitative result in Table 1 shows that the 3DcGAN network achieves the total accuracy rate of 82.8% with detection rate 82.3% and false 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.



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

***Table 1 (left)**: The ablation analysis of the 3d-gan networks on NTHU-DDD testing dataset [50].*
***Table 2 (right)**: The ablation analysis of the several LSTM-based techniques on NTHU-DDD testing dataset [50].*

2.3 Selected Works

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

2.3.1 Driver Drowsiness Detection through HMM based Dynamic Modeling

In the work of Tadesse et al., 2014 [29], researchers investigate the driver drowsiness detection using facial expression recognition for single frame-based analysis and **HMM** based dynamic modeling. To detect driver drowsiness, firstly they applied **Viola-Jones** and **Camshift** algorithms in order to detect the driver's face. After face detection they extracted the facial features by using **Gabor Wavelets**. These parts are exactly the same for both classification systems.

For single frame-based analysis firstly they combined the weak classifiers working on each selected feature to get a strong classifier by using **Adaboost Cascaded Classifier**. After that, since this topic is a two-class problem (drowsy, non-drowsy) they fed the selected features to the **SVM** by using a **kernel method** which proved to have raised in performance over the linear combination of the **Adaboost** weak classifier.

Finally, they implemented **HMM** based dynamic modeling. By using **HMMs**, they captured the temporal information of the facial expressions of the driver which leads to more accurate classification results as compared to single frame-based drowsiness detection.

In conclusion, a dynamic approach gives more performance (97% accuracy) than the single frame-based analysis (90% accuracy). Therefore, it has been stated that facial expressions are better recognized through the analysis of sequence of frames. But one drawback of this paper is the dataset they use. Researchers decided to collect their own dataset instead of using one of the common ones and they end up with a very limited one which consists of only 2 subjects.

We are planning to use this paper's approach of using dynamic and frame-based models at the same time. But instead of just implementing and comparing two models, we will try to ensemble them in order to increase accuracies.

2.3.2 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 failure. 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 in 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 are also planning to use **PERCLOS** feature but with a simpler method based on facial landmarks of Dlib library.

2.3.3 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.

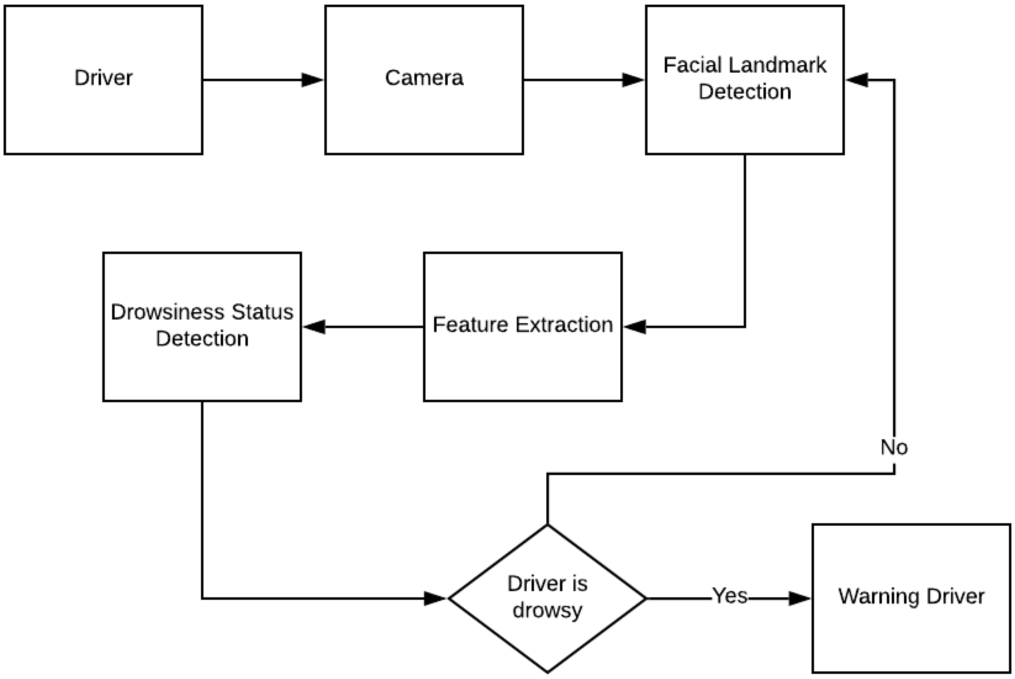
Consequently, the paper represents publicly available, real-life dataset RLDD which is the largest to date (2019) along with end-to-end baseline method using the temporal relationships between blinks to detect early signs before an accident. Overall, the paper hopes that the proposed public dataset will also encourage other researchers to work on drowsiness detection and produce additional and improved results that can be duplicated and compared to each other. We are also planning to use this dataset with similar sequential approach (LSTM) along with blink features, additionally frame-based features from other papers and scalar models as supplements.

3. System Design

In this project, we will study how to implement a real-time driver drowsiness detection system with high accuracy.

3.1 System Model

This project proposes a system in Figure 10. The driver must set up the camera before getting behind the wheel in a proper way that his/her full face can be captured by the camera while driving. Then the camera captures the face and the system detects the facial landmarks. Therefore, the system is able to extract the facial features. The driver should look directly to the camera for 5 seconds before driving for feature normalization. Thereby, the system arranges the threshold values for each feature by the driver's face. This enables the system to become adaptive by the subject. Later, drowsiness status of the driver will be estimated using the pre-trained model. Depending on this status, if the driver is drowsy then the system sends a warning to the user that he/she is sleepy. If the driver is not sleepy then the system continues to evaluate the drowsiness status until it is closed by hand or the driver is drowsy.



***<u>Figure 10</u>**: A real-time Driver Detection System Model. Using predictions of the DDD system to warn drivers in dangerous situations.*

3.2 Flowchart

Our drowsiness detection system starts by reading videos, continues with detecting faces, calculating features and preprocessing. After subject-wise normalization to handle adaptivity problems, the system will produce two kinds of features; **blink-based** and **frame-based**. The first kind of features also require blink detection before calculating features, so there will be an extra module for it. Both of the models will be handed to classifiers to train and generate predictions on test sets. There are two kinds of datasets, while most of them provide **labels (drowsy / not drowsy) for the whole video**, others provide **labels for frames** additionally. For video classification we need to build sequential models such as HMM or LSTMs and for frame classification it can be done by conventional machine learning tools such as CNN, SVM, Decition trees, KNN or Naïve-Bayes. In the final step, the system will produce 2 models based on features and 2 models based on labels. Hence, we will try to merge these models by using **ensemble learning** techniques (see Figure 11).

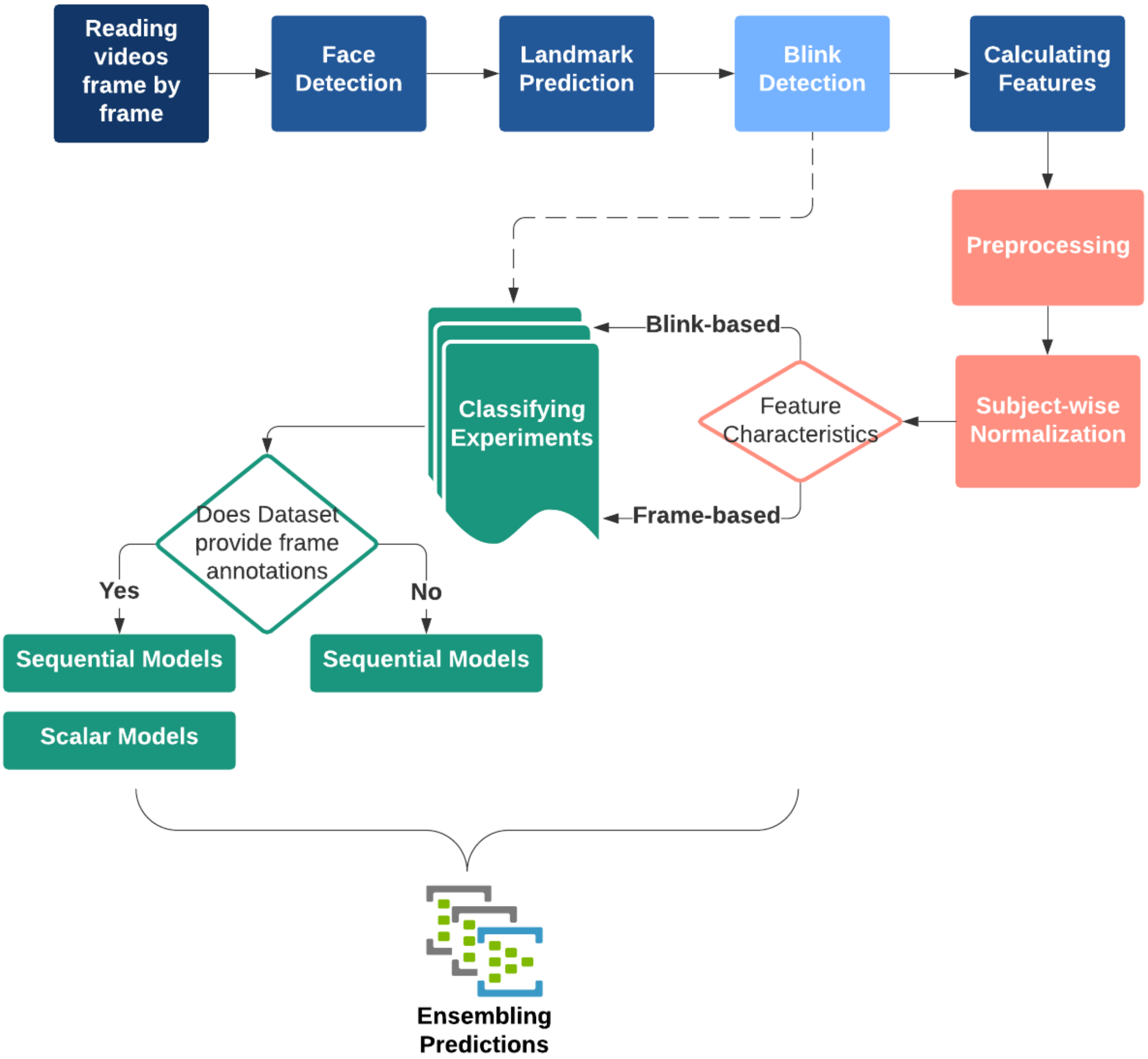


***Figure 11**: Flowchart diagram of the driver drowsiness detection system.*

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 12**)**. **Preprocessing Module** to reading and processing videos, **Feature Extraction Module** to construct ad-hoc features (there will be 2 different paths which are explained in section 3.2.2 and 3.2.3), **Classification Module** to train our input data and finally **Evaluation Module** to make predictions on test data and to calculate accuracy metrics.

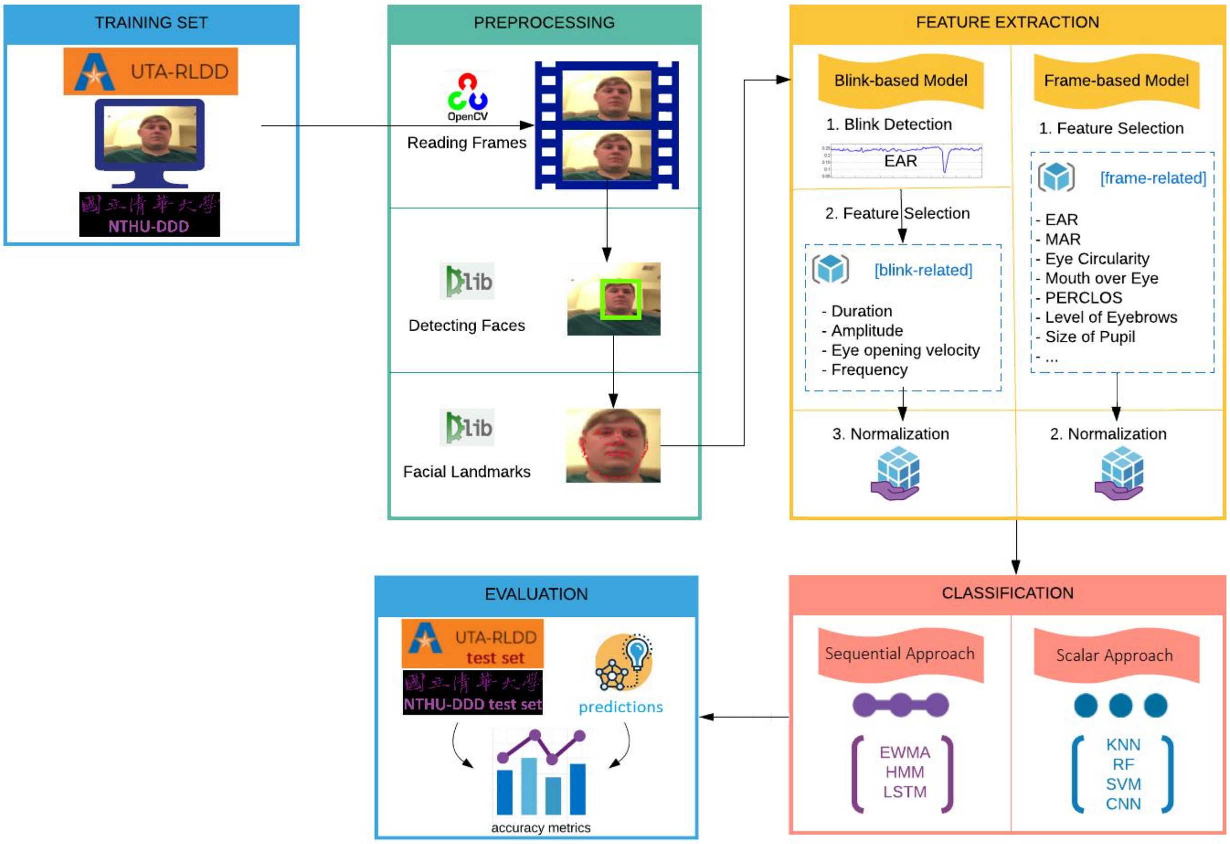


***Figure 12**: Conceptual diagram of the driver drowsiness detection system.*

3.2.1 Preprocessing Module

Implementation of the project starts with **Preprocessing Module** which includes steps;

1. **Reading video frames**, either can be done from a dataset or a camera in a real-time manner. For this step, **opencv-python** [51], python version of OpenCV library, is used.

2. **Detecting faces** with **Dlib's get_frontal_face_detector** [52] method. It uses a pre-trained "Histogram of Oriented Gradients + Linear SVM" pipeline for face detection. There's also another CNN-based method in Dlib but it isn't suitable for real-time purposes.

3. **Predicting facial landmarks** with **Dlib's shape_predictor** [53] method which is an implementation of the paper Kazemi and Sullivan (2014) [54]. This method also uses a pre-trained model of ensemble of regression trees and predicts 68 facial landmarks which can be seen in Figure 13**.** All features will be used in later phases, are extracted from these positional landmarks.

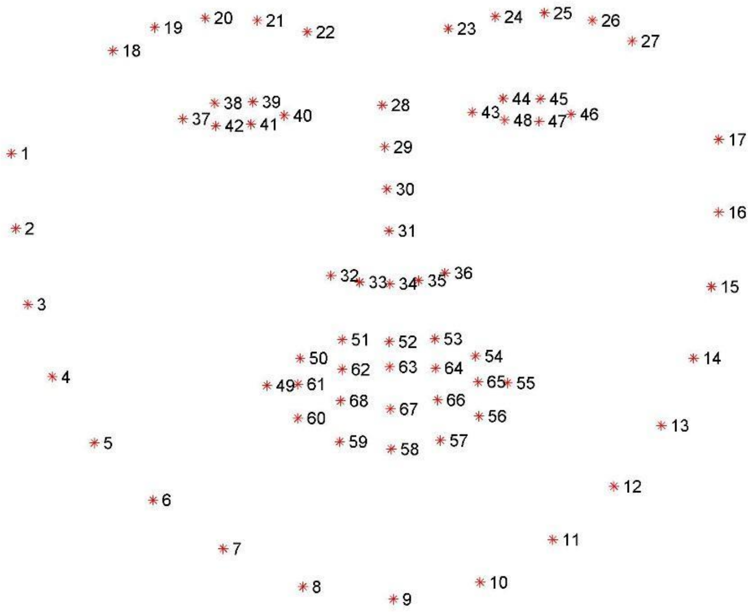


__Figure 13__: 68 facial landmark coordinates of Dlib's shape_predictor method.

After the preprocessing phase, implementation continues with the **Feature Extraction** phase. Since there are two models to experience on, as shown in Figure 12, there will be different features for both.

- **Blink-based Model**: Detects blinks, extracts blink-related features and classifies video sequences based on these features.

- **Frame-based Model:** Without detecting any facial action (blink, yawning etc.), all features are computed for every frame and all of this information will be used for classification.

So, in that context, both models will be explained separately.

3.2.2 Feature Extraction Module of Blink-based Model

For further detail, since the **Blink-based Model** is just an implementation of the work by Ghoddoosian et al (2019) [12], it starts with blink detection, then continues calculating blink-related features.

1. **Blink detection step** is also an implementation of another paper Soukupová and Chech (2016) [42] and investigates of using a simple mathematical formula for real-time purposes which is called "Eye Aspect Ratio (EAR)" (7.1) and can be extracted from eye landmark coordinates in Figure 13.

$$EAR(i) = \frac{\|p_{38} - p_{42}\| + \|p_{39} - p_{41}\|}{2\|p_{37} - p_{40}\|}, \qquad (7.1)$$

In the formula here, p_{37}, …, p_{42} are 2D landmark locations of the left eye depicted in Figure 13 and Figure 14 and i is the frame index. $\|p_a - p_b\|$ represents the Euclidean distance between two landmark positions.

After calculation of EAR for both eyes, average of them is calculated for a window of n = 13 frames (n is determined empirically) and these sequences are used for detecting blinks with SVM classifier.

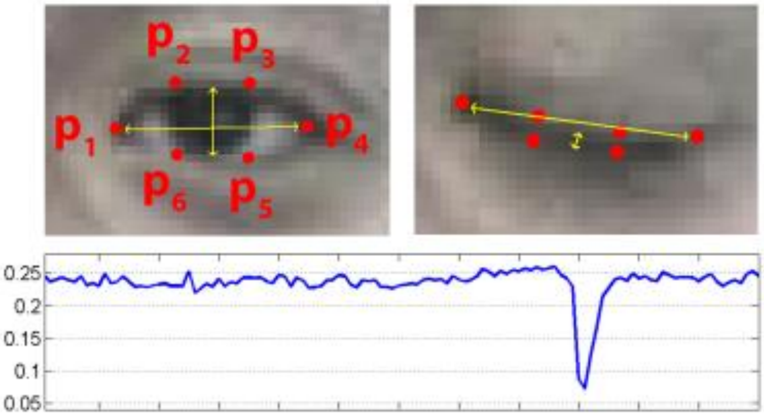


***Figure 14**: An original figure from the 2016 EAR paper [42]. Eye landmarks are used in the calculation of EAR with open/closed eye scenarios.*

2. **Selection of blink features:** Output of blink detection step is a sequence of blink events $\{blink_1, \dots blink_k\}$, each $blink_i$is a 4D-vector consisting relevant EAR(i) value along with $start_i$, $bottom_i$ and end_i values which are timestamps (frame count starting from the beginning of the video) of start, bottom and endpoints of a blink (see Figure 15). As explained in [12] 4 blink-related features are calculated from these 4D vectors which are;

    - **Duration** measures the duration of a blink, which can be formulated as

    $$Duration_i = end_i - start_i + 1, \qquad (7.2)$$

    In the formula (7.2), i represents index, end_i represents timestamp (= frame count starting from the beginning of the video) of ending frame and $start_i$ represents the timestamp of the starting frame of a blink.

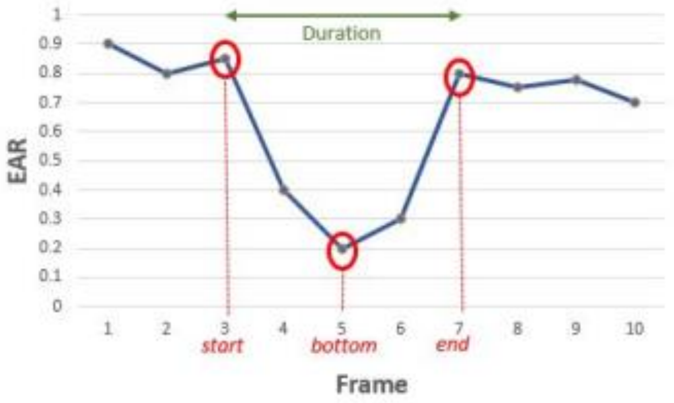


***Figure 15**: An original figure from the RLDD paper [12], shows starting, ending and bottom frames of a detected blink depicting duration.*

- **Amplitude:** Average amplitude of EAR values of starting and ending frames.

$$Amplitude_i = \frac{EAR(start_i) - 2EAR(bottom_i) + EAR(end_i)}{2}, \qquad (7.3)$$

In the formula (7.3), i represents index, end_i represents timestamp (=frame count starting from the beginning of the video) of the ending frame, $start_i$ represents timestamp of the starting frame and $bottom_i$ represents the timestamp of the bottom frame of a blink. Also $EAR(i)$ represents EAR value of ith frame which is calculated with the formula (7.1) (see Figure 16).

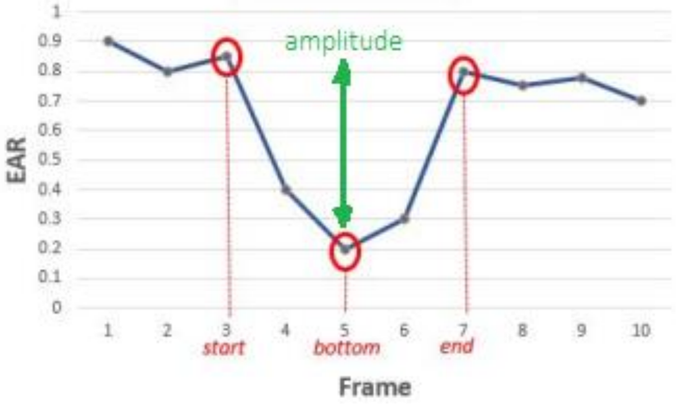


***Figure 16**: An original figure from the RLDD paper [12], shows starting, ending and bottom frames of a detected blink depicting amplitude.*

- **Eye Opening Velocity** measures how fast a subject opens his eyes. This can be a good feature to indicate the drowsiness level.

$$Eye\ Opening\ Velocity_i = \frac{EAR(end_i) - EAR(bottom_i)}{end_i - bottom_i}, \qquad (7.4)$$

In the formula above (7.4), i represents index, end_i represents the timestamp (=frame count starting from the beginning of the video) of the ending frame and $bottom_i$ represents the timestamp of the bottom frame of a blink. Also $EAR(i)$ represents EAR value of ith frame which is calculated with the formula (7.1).

- **Frequency i**ndicates the frequency of previously occurring blinks up to a blink.

$$Frequency_i = 100 \times \frac{number\ of\ blinks\ up\ to\ blink_i}{number\ of\ frames\ up\ to\ end_i}, \qquad (7.5)$$

In the formula above (7.5), i represents index, $blink_i$ represents ith blink, end_i represents timestamp (= frame count starting from the beginning of the video) of the $blink_i$.

After calculation of 4D vectors of each $blink_i$ in the sequence $\{blink_1, \dots\ blink_k\}$, normalization phase starts.

3. **Normalization step:** Each feature calculated above needs to be normalized across subjects in videos because blinking behaviors change person to person. There could be a subject with slant eyes whose EAR values show different characteristics compared to other subjects. So subject-wise normalization is essential to get more accurate results.

   On the other hand, normalization across features also had to be done. We are planning to use the first third of blinks detected in an alert video of a specific subject to normalize features. So, this process can be formulated as,

$$normalized\ feature_{n,m} = \frac{feature_{n,m} - \mu_{n,m}}{\sigma_{n,m}}, \qquad (7.6)$$

   Where, $\mu_{n,m}$ and $\sigma_{n,m}$ are mean and standard deviation of the feature n of the subject m.

   In order to increase success of our models, there will be a standard column-wise normalization also, just like subject-wise normalization. This will be done by scaling each feature with mean and standard deviation.

$$scaled\ feature_n = \frac{feature_n - \mu_n}{\sigma_n}, \qquad (7.7)$$

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

3.2.3 Feature Extraction Module of Frame-based Model

Unlike the Blink-based Model, the Frame-based Model doesn't include a blink detection step. All of the features that are presented below, are calculated for every frame in the video and will be handed into the classifier. So, this way the model doesn't discard any data that can be valuable in the classification phase.

1. **Feature selection step:** This model uses 3 features defined in an online paper by Grant Zhong [49].
   - **Eye aspect ratio (EAR):** Just like the Blink-based Model, the eye aspect ratio of all frames is calculated with the formula (7.1). The average of both eyes will be selected as a feature.

   - **Mouth aspect ratio (MAR):** This formula resembles EAR (7.1), in the context of using 68 facial landmarks (see Figure 11). It uses inner landmarks of the mouth (61, ..., 68) and calculates a ratio just like EAR. Therefore, it can be useful for detecting yawning behavior [25].

$$MAR(i) = \frac{\|p_{63} - p_{67}\|}{\|p_{61} - p_{65}\|}, \qquad (7.8)$$

In the formula here (7.8), p_{61}, …, p_{67} are 2D landmark locations of the inner mouth shape depicted in Figure 17 and i is the frame index. $\|p_a - p_b\|$ represents the Euclidean distance between two landmark positions.

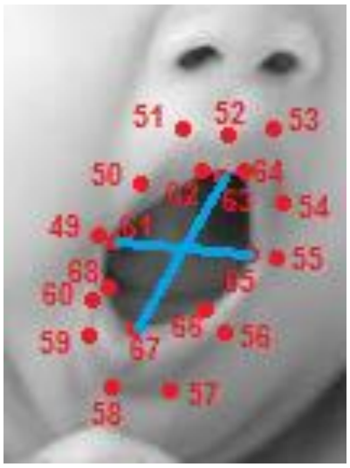


***Figure 17**: Ratio between [p63, p67] and [p61, p65] to measure Mouth Aspect Ratio (MAR).*

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

$$EC(i) = \frac{4 \times \pi \times Pupil\ Area}{(Eye\ Perimeter)^2}, \qquad (7.9)$$

$$Pupil\ Area = \left(\frac{\|p_{38} - p_{41}\|}{2}\right)^2 \times \pi, \qquad (7.10)$$

$$Eye\ Perimeter = \|p_{37} - p_{38}\| + \|p_{38} - p_{39}\| + \|p_{39} - p_{40}\| + \|p_{40} - p_{41}\| + \|p_{41} - p_{42}\| + \|p_{42} - p_{37}\|, \quad (7.11)$$

In the formulas above (7.9, 7.10, 7.11) p_{37}, …, p_{42} are 2D landmark locations of the left eye shape depicted in Figure 18 and i is the frame index. $\|p_a - p_b\|$ represents the Euclidean distance between two landmark positions. The average of both eyes will be selected as a feature.

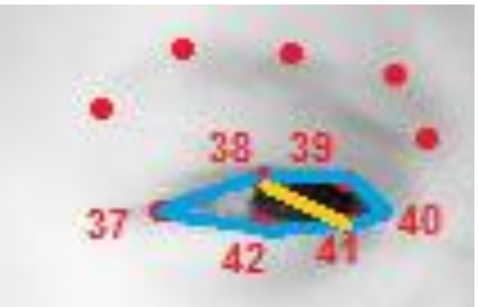


Figure 18: *Blue lines represent Eye Perimeter and the yellow line represents Pupil Diameter which is used in the calculation of Eye Circularity (EC).*

- **Mouth over Eye (MOE):** Basically EAR (7.1) over MAR (7.7). [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.

$$EC(i) = \frac{MAR(i)}{EAR(i)}, \qquad (7.12)$$

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

- **PERCLOS**: Indicates the frequency of closed eyes up until that moment. [45] We are planning to use it with a small-time window like t = 30 seconds.

$$PERCLOS = \frac{count\ of\ frames\ when\ the\ eyes\ are\ closed}{total\ count\ of\ frames\ up\ until\ that\ moment} \times 100\%, \quad (7.13)$$

- **Level of Eyebrows (LEB):** Level of eyebrows can be a good measure to drowsiness so another formula that calculates the average distance between first two of inner points of eyebrows and inner corner of an eye. Other points of eyes are ignored due to their moving nature. Other than two points of eyebrows are ignored also, since they are more stationary.

$$LEB(i) = \frac{\|p_{21} - p_{40}\| + \|p_{22} - p_{40}\|}{2}, \qquad (7.14)$$

In the formula above (7.14) 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 19**.** i is frame index and $||p_a - p_b||$ represents euclidian distance between two landmark positions. The average of both eyes will be selected as a feature.

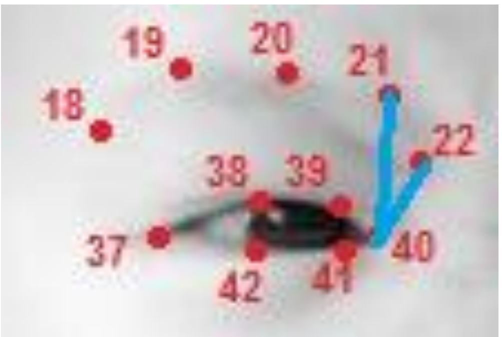


Figure 19: *Blue lines represent distances [p21, p40] and [p22, p40]. Average of them is calculated to measure the Level of Eyebrows (LEB).*

- **Size of Pupil (SOP):** Size of pupil can be a good measure of alertness. It's 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.

$$SOP(i) = \frac{||p_{38} - p_{41}||}{||p_{37} - p_{40}||}, \qquad (7.15)$$

In the formula above (7.15), p_{37}, …, p_{40} are 2D landmark locations of the left eye depicted in Figure 20 and i is the frame index. $||p_a - p_b||$ represents the Euclidean distance between two landmark positions. The average of both eyes will be selected as a feature.

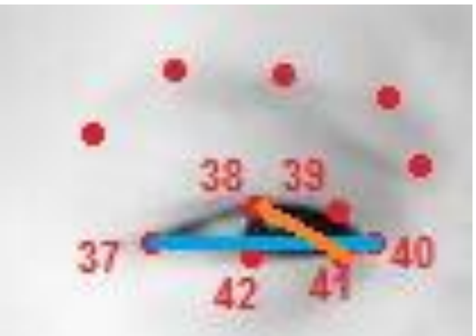


Figure 20: *Blue line represents Eye Width [p37, p40] and orange line represents Pupil Diameter [p38, p41]. The ratio of them is called Size of Pupil (SOP).*

2. **Normalization step:** Just like the normalization step in the Blink-based Model, all of the features will be normalized across subjects and features itself by using the formula-7.6 and 7.7. But unlike Blink-based Model, Frame-based Model can use just a few frames of an alert subject instead of a huge number of blinks. So, this method is more suitable in real-time scenarios.

3.2.4 Classification Module

After preprocessing and feature extraction, we are planning to try some classification techniques starting from conventional ones (Logistic Regression, Naïve-Bayes, K-Nearest Neighbor, Decision tree, Random Forest, SVM) to novel approaches (XGBoost, CNN, LSTM) with increasing complexity.

One of the important aims of the project is introducing an early warning mechanism to the system so sequential/temporal relations play a key role to achieve it. When conventional machine learning tools investigate each frame, they don't take sequential relation between frames into consideration, unlike some deep learning approaches like LSTM and HMM. Therefore, using EWMA (exponentially weighted moving average) on a large-time window like t = 3 minutes is one of the solutions which can be experimented on conventional machine learning tools.

$$EWMA(i) = w \times x(i) + (1 - w) \times EWMA(i - 1), \qquad (7.16)$$

For the formula above (7.16), w is weight, i is frame index and $x(i)$ is a feature we want to soften like EAR, MAR, etc. when $x(i)$ represents the current value of a feature, $EWMA(i - 1)$ represents past values. Thus $EWMA(i)$ calculates the recursive mean of a feature. $1/w$ approximately gives a number of past values to be considered and in that manner w will be chosen empirically.

3.2.5 Prediction and Evaluation Module

By using the models trained in the previous phase and getting prediction results on the test set, it's possible to compare them with truth labels in two different manners:

- Predictions on the drowsiness level of the subject in a frame or
- Predictions 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. As explained in section 3.2.4, time-series tools like LSTM and HMM can be used on classifying whole videos. In addition to them, basic Machine Learning tools also can be used in the same way after softening the values of frames with EWMA (Formula-7.16).

Therefore, we are planning to use sequential tools (LSTM, HMM, EWMA) when video labels available (on UTA-RLDD and NTHU-DDD) and static/granular tools (other ML techniques) when frame labels available (only NTHU-DDD).

Officially recommended evaluation method for UTA-RLDD is "*Use one-fold of the UTA-RLDD dataset as your test set and the remaining four folds for training. After repeating this process for each fold, the results would be averaged across the five folds.*" [9]. With this way, **RMSE** (Root Mean Square Error) will be used as an evaluation metric as recommended.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y_i} - y_i)^2}{n}}, \qquad (7.17)$$

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

There's also an evaluation script provided for NTHU-DDD which uses accuracy as an evaluation metric [10]. We will stick to official evaluation methods to compare the results of papers as baselines to ours.

3.3 Comparison Metrics

In addition to the recommended metrics defined in Section 3.2.5**,** we are planning to use standard comparison metrics defined in Table 3 and 4.

	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*: Symbol definitions 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

Table 4*: Formulas for comparison metrics*

3.4 Datasets 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

4. System Architecture

This section explains our drowsiness detection system architecture. Figure 21 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 two separate operations for different types of features. The first operation is extracting blink-based features. Blink-based model uses sequential data and frame-based model evaluates the features in each frame individually. The system executes feature selection algorithms for both models and normalizes the features by both subject-wise and column-wise. At the end of this stage, we have two separate dataframes for the blink-based and frame-based models. These dataframes flow through the **(8)** 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 annotated then we use both sequential and scalar classification methods. If it is video-labeled, then we use only sequential methods. After the classification stage is over, trained models flow through the **(9)** and tested with the test set.

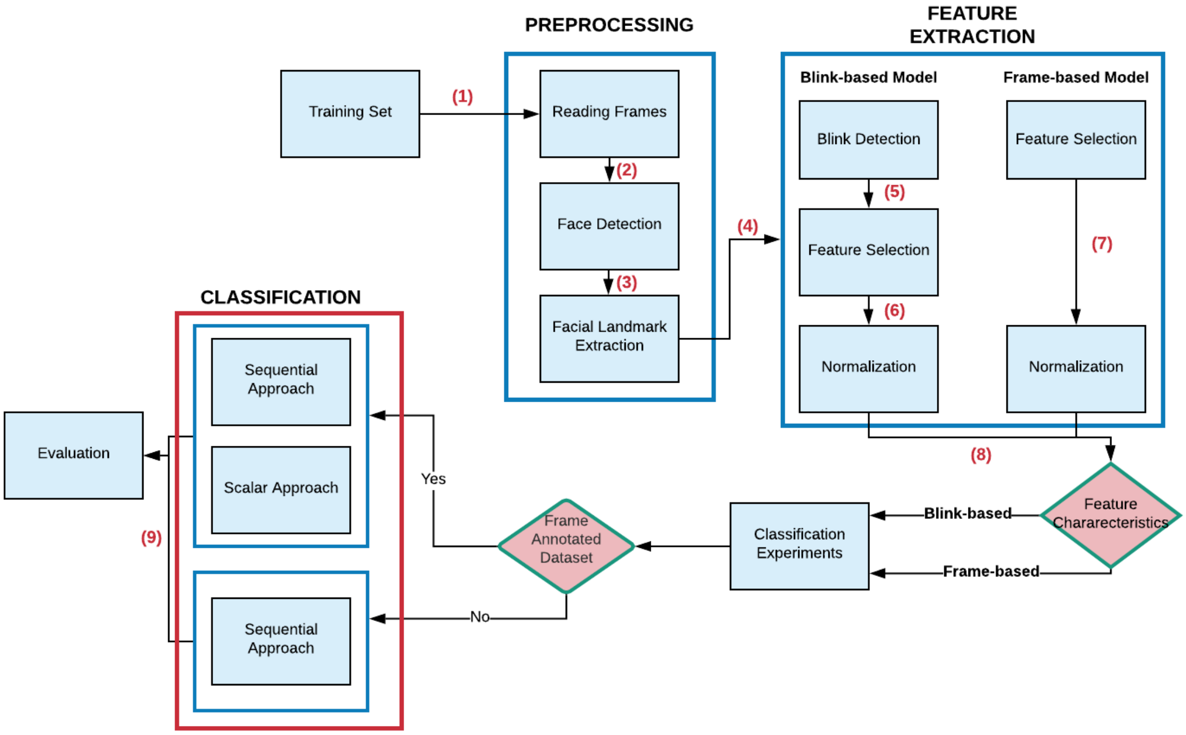


__Figure 21__: Data / Control Flow of the Driver Drowsiness Detection System. Flow of the data through the system is; videos of the dataset (1), gray scales frames of each video (2), face region of the frames (3), facial landmarks location of the face (4), blink features (5), selected blink features (6), selected frame-based features (7), dataframes with normalized features (8), trained model of frame-based dataframe and blink-based dataframe (9).

5. Experimental Study

In this section we will explain our experiment environment and preliminary results.

5.1 Experimental Setup

Preliminary experiments are completed in online **Kaggle** notebook environment (CPU-only) [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 models: blink-based model and frame-based model as explained in Section 3.2. There will be preprocessing, feature extraction, classification and evaluation steps for both of models. One difference for blink-based model is **blink detection** step. We completed blink detection experiments for blink-based model and some simple classification experiments for frame-based model by the end of the first term.

- **Results for Frame-based Model:** Out of our two datasets, only NTHU-DDD provides frame related labels, so the experiments for this model are run only for it. After reading frames by OpenCV and Dlib and extracting features from landmark positions, just like explained in Section 3.2.1, we measured impurity-based feature importances by training a Random Forest Classifier (see Figure 22**)** along with **permutation importances.** Permutation importance method randomly shuffles values of a feature over rows and measures the decrease in the model score. This procedure breaks the relationship between the feature and the target, and in that way, importance of a feature can be calculated. After examining results of feature elimination step (see Figure 22 and 23**)**, a feature named **CLOSENESS** is discarded due to its low importance and other features are handed to three classifiers: **Decision Tree**, **Naïve-Bayes** and **K-nearest Neighbor**.

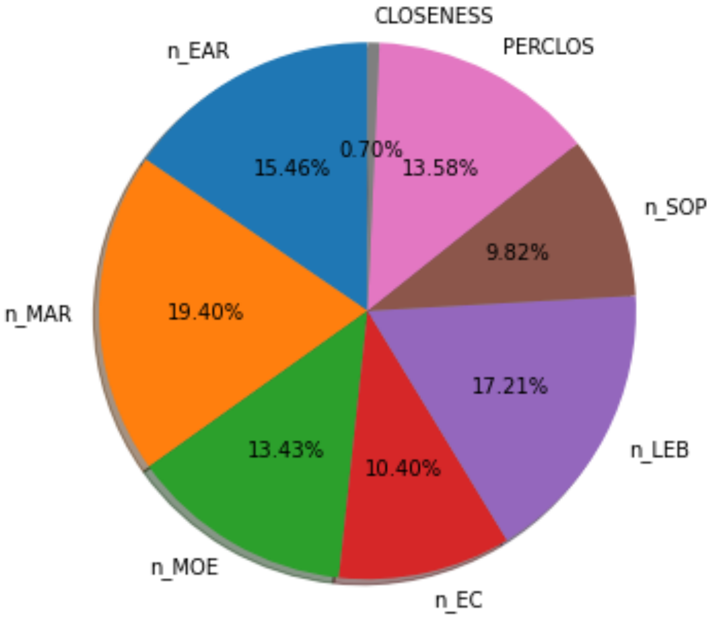


Figure 22: *Importances of features provided by Random Forest model.*

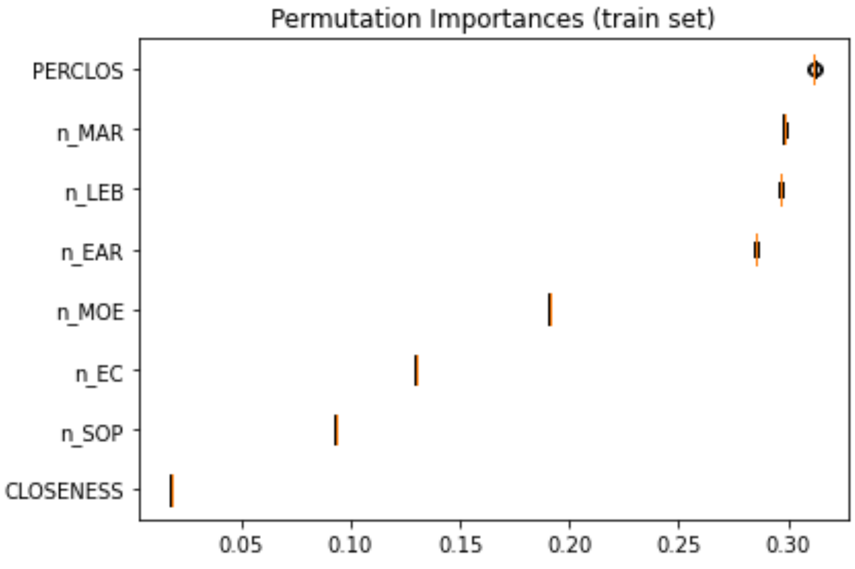


Figure 23: *Permutation importances of features*

For classification, we used 80-20 train-test splitting and 10-fold stratified cross-validation. After determining best parameters and testing our models, we obtained the results in Table 5. We have also done additional column-wise normalization to increase scores. Among 4 classifiers, Random Forest provides best scores (F1: 82%).

		Non-Scaled (only subject-wise)	Scaled (subject-wise + column-wise)
Decision Tree	Accuracy	0.7276	0.7298
	AUC	0.7194	0.7214
	F-1	0.77	0.77
Naive-Bayes	Accuracy	0.5598	0.5595
	AUC	0.5947	0.5595
	F-1	0.51	0.51
K-nearest Neighbor	Accuracy	0.7079	0.7374
	AUC	0.6961	0.7258
	F-1	0.75	0.78
Random Forest	Accuracy	0.7894	0.7903
	AUC	0.7824	0.7829
	F-1	0.82	0.82

Table 5*: Scores of four classifiers, scaled and non-scaled versions.*

These were just preliminary experiments; in the next term, we are planning to add more classifiers to frame-based model such as SVM, XG-Boost and CNN along with extra preprocessing techniques.

- **Results for Blink-based Model:** Blink-based model differs from frame-based model only by feature calculation. Before calculating blink features, we need to build a proper blink detection system. In order to achieve this, we run several experiments [56] on training set (**eyeblink8** consists of 8 videos, 71260 frames [11]) and test set (**talkingFace** consists of only 1 video, 5000 frames [11]):
  1. **Simple Thresholding:** We defined 3 simple thresholds when detecting blinks:
  - EAR_THRESHOLD = 0.21; eye aspect ratio to indicate blink.
  - EAR_CONSEC_FRAMES = 3; number of consecutive frames the eye must be below the threshold.
  - SKIP_FIRST_FRAMES = 0; how many frames we should skip at the beginning for calibration phase.

  2. **Adaptive Thresholding:** We tried to make threshold defined above adaptive with some time-series approaches:
  - Adaptive EAR_THRESHOLD: EWMA + outlier detection (see Figure 24).
  - Adaptive EAR_CONSEC_FRAMES: Predicting significant values of PACF plot and

testing it with gridsearch + ARIMA.

- Adaptive SKIP_FIRST_FRAMES: To capture the first complete blink, we keep track of slope of linear fitting.

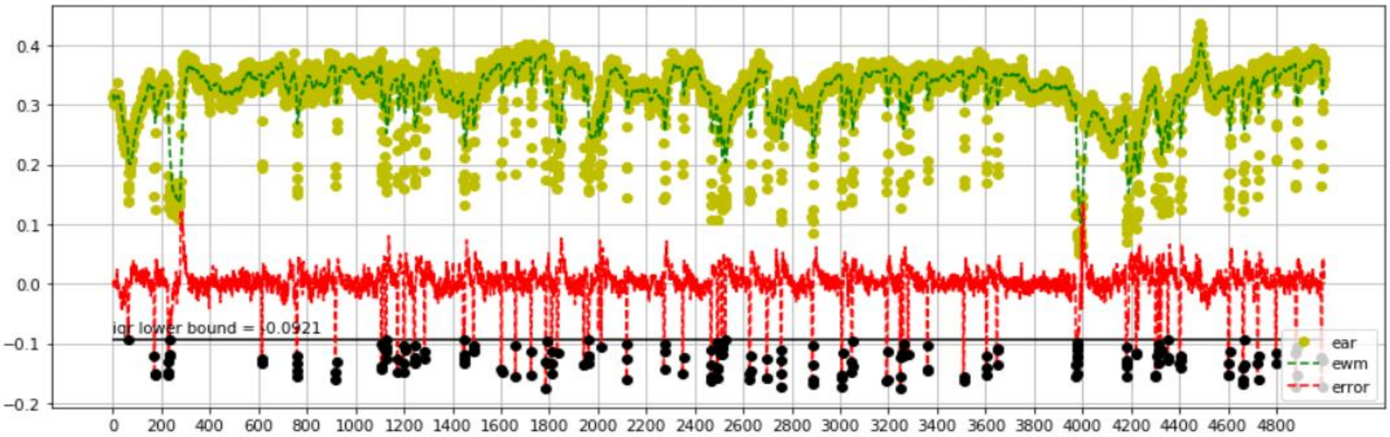


***Figure 24**: Adaptive thresholding on talkingFace dataset. Yellow dots represent EAR values, green dashed line is EWMA, red line is the difference between EAR-EWMA and black dots are outliers that can be inferred as blinks.*

3. **Machine Learning Model:** We constructed a 13-D dataframe from EAR values with 13 frame-window, 6 for previous frames, 6 for next frames and the current frame. We used this dataframe to train an SVM model (C: 10, 'class_weight': None, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': 5000) with 10-fold CV + gridsearch to predict blinks.

We compare these 3 models (see Figure 25) and it's concluded that ML model produces the best scores (F1: 86%). With using this model, we will calculate blink-related features and implement blink-based model in the next term.

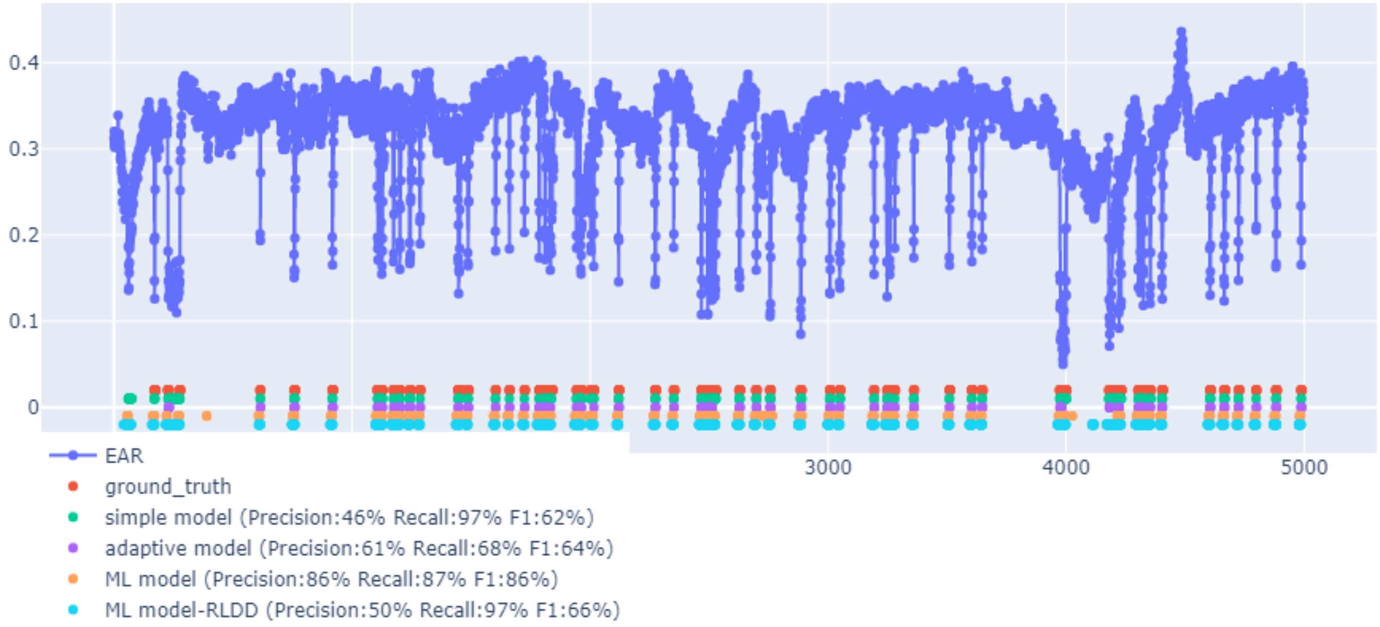


***Figure 25**: Comparison of 3 blink detection models along with pre-trained SVM model provided by RLDD dataset [57].*

6. Tasks Accomplished

In this section, we will explain our task planning, completed task and meeting schedule.

6.1 Current State of The Project

As explained in section 5.2 Experimental Results, we work through 2 main paths in our project. Some group members work on frame-based model and complete all of the steps defined in section 3.2 and Figure 12 for couple of classifiers (DT, NB, RF, KNN). There will be other classifiers to be added like CNN, XGBoost and SVM later in the project.

On the other hand, for blink-based model, implementing a successful blink detector was an essential part so we focused on that. We tried several experiments can be seen in [56] and we end up with a Machine Learning model on this part. In continuation of our project, we will extract blink features based on this model and implement blink-based model for driver drowsiness detection system.

Preliminary results for both of models can be seen in section 5.2 also. After implementing both of models with scalar classifiers, we will try dynamic ones also, with sequential approaches like LSTM, HMM and EWMA, to see if we can get better results in the manner of evaluation metrics and project goals.

6.2 Task Log

During this semester, between 20.01.2020 and 26.05.2020 we had weekly meetings with our advisor on Tuesdays. In these meetings, we present our progress and discussed our roadmap. In first two months of meetings, until pandemic break, we did literature search on our project topic with the guidance of the advisor.

Until the end of March, we have found enough time to start coding and implemented some simple tasks like processing videos. When remote classes started in April, we continued our weekly meetings on online platforms and developed important parts of the project such as frame-based model and blink detection modules.

We are planning to continue to work on the project in summer and complete the implementation of scalar models just by leaving the sequential models for second semester.

6.3 Task Planning and Milestones

For this topic, we will present our Task Calendar/Schedule on Table 6. Milestones are determined as deadlines of project documents and presentations.

Task No	Task Description	Expected Output	Months										
			2	3	4	5	6	7	8	9	10	11	12
1	**Literature Survey:** Previous researches and publications are going to be scanned. The similarities and differences between previous projects are going to be investigated and the future path of the project is going to be determined.	Presentations for weekly meetings.	■										
2	**Establishing Environments:** Necessary tools and technologies will be established to work with properly.	Personal computers and online notebooks.	■										
3	**Implementing Blink Detection for Blink-based Model:** Building and evaluating an adaptive blink detector on available blink datasets in the literature.	A successful classification model for blink detection.		■	■								
4	**MILESTONE 1:** Preparing Project Specification Document**.**	A .pdf document.		■	■								
5	**Feature Extraction & Normalization for Frame-based Model:** The extraction of features is going to be available and after that be normalized to get better accuracy. In the feature extraction two different methods will be used. Blink based and frame-based detections are going to be held.	A pandas dataframe for classification purposes.		■	■								
6	**MILESTONE 2:** Preparing Analysis and Design Document**.**	A .pdf document.			■								
7	**Comparing and Deciding which features will be used for classification:** Eliminating unnecessary features and generating fresh features. Searching for if there is a possibility to implement features of blink-based model frame-by-frame and if it does, then is it possible to make the system as a hybrid model.	Analysis reports with plots and tables.				■							
8	**Implementing Classification:** Both using classical Machine Learning techniques (SVM, k-NN, HMM) and deep learning techniques (CNN, LSTM). Searching for possibility of ensemble two models (blink-based and frame-based) in classification phase.	Pre-trained classification models with using SKLearn and Keras libraries.					■			■			
9	**Testing the Project:** Testing the final project in order to see whether it gives the warning correctly and determining the evaluation metrics for comparing state-of-art results.	Scores of evaluation metrics on test datasets of RLDD and NTHU-DDD.									■		
10	**Real-Time Demonstration:** Constructing the real-time demonstration.	Simple GUI on a computer connected to a webcam.									■	■	
11	**MILESTONE 3:** Preparing Project Report & Poster.	A .pdf document and a poster.											■

Table 6: *Task Calendar/Schedule*

6.4 Timeline

Time planning of tasks can be seen in Figure 26 as GANTT chart. Dark green rows are for completed tasks, light green ones are for tasks under process and red ones are for incoming tasks.

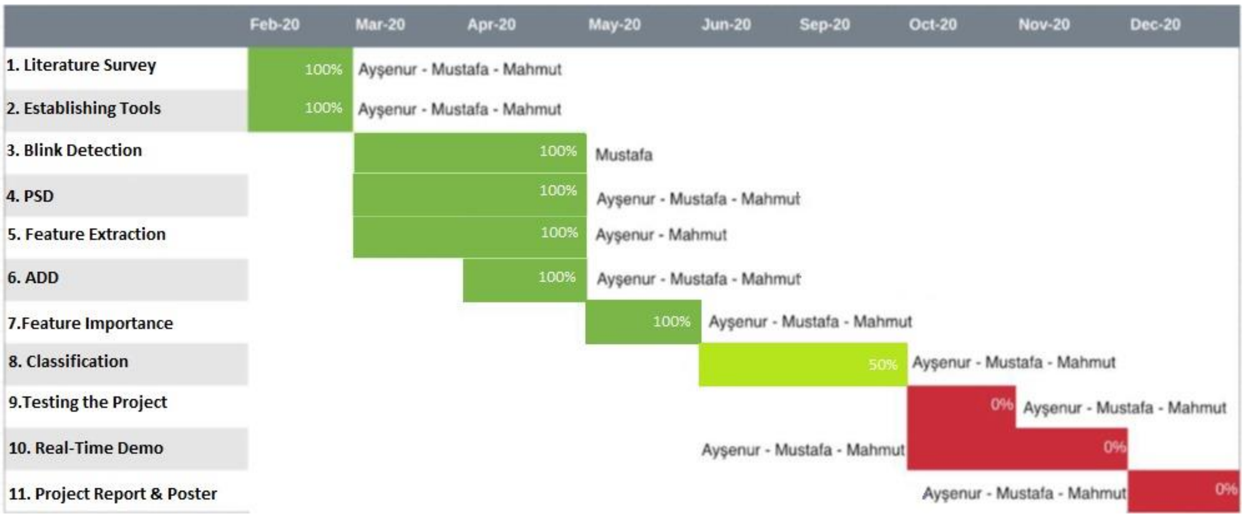


Figure 26: *GANTT Chart for work management, timeline and milestones*

7. 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] Blink Matters, Research. [Online]. Available: https://www.blinkingmatters.com/research (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] M. Hashemi, A. Mirrashid, and A.B. Shirazi, "Deep learning based Driver Distraction and Drowsiness Detection. ArXiv, abs/2001.05137, 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)