A Project Report on

Driver Drowsiness And Alert System

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Saylee Patne (16104021)
Riya Sangal (16104014)
Yashashree Gore (17204009)

Under the Guidance of

Prof. Anagha Aher

Prof. Geetanjali Kalme

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled "***DRIVER DROWSINESS AND ALERT SYSTEM"*** Submitted by ***"Saylee Patne" (16104021), "Riya Sangal" (16104014), "Yashashree Gore" (17204009),*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai.***

(Prof. Geetanjali Kalme)
Co-Guide

(Prof. Anagha Aher)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Driver Drowsiness and alert System"*** submitted by ***"Saylee Patne" (16104021),"Riya Sangal" (16104014),"Yashashree Gore" (17204009),*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology.***, to the University of Mumbai, is a bonafide work carried out during academic year 2017-2018.

(Prof.Geetanjali Kalme)
Co-Guide

(Prof.Anagha Aher)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s) 1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Saylee Patne (16104021))

(Riya Sangal (16104014))

(Yashashree Gore(17204009))

Date:

Abstract

Drowsy driving is one of the major causes for road accidents. Hence, detection of driver’s fatigue and its indication is an active research area. Therefore, in this study, a low cost, real time driver’s drowsiness detection system is developed with acceptable accuracy. In the developed system, a front camera of driver’s cell phone records the video and drivers face is detected. For this purpose, eyes, nose and mouth are detected to improve the area of interest using Google vision API. It is the best choice when it comes to face detection for better performance of Real Time Image processing. B4A is used as it is a simple and powerful tool for developing the Android applications without time-consuming. When the drowsiness is detected, then the driver is alerted by audio.

Content

1.Introduction

Drowsy driving is one of the major causes of deaths occurring in road accidents. The truck drivers who drive for continuous long hours (especially at night), bus drivers of long-distance route or overnight buses are more susceptible to this problem. Driver drowsiness is a nightmare to passengers in every country. Every year, a large number of injuries and deaths occur due to fatigue related road accidents. Hence, detection of driver's fatigues and its indication is an active area of research due to its immense practical applicability. The basic drowsiness detection system has three blocks/modules; acquisition system, processing system and warning system. Here, the video of the driver's frontal face is captured in acquisition system and transferred to the processing block where it is processed online to detect drowsiness. If drowsiness is detected, a warning or alarm is sent to the driver from the warning system. Generally, the methods to detect drowsy drivers are classified in three types; vehicle based, behavioral based and physiological based. In vehicle-based method, a number of metrics like steering wheel movement, accelerator or brake pattern, vehicle speed, lateral acceleration, deviations from lane position etc. are monitored continuously. Detection of any abnormal change in these values is considered as driver drowsiness. This is intrusive measurement as the sensors are attached on the driver which will distract the driver. Depending on the sensors used in the system, system cost as well as size will increase. However, inclusion of more parameters/features will increase the accuracy of the system to a certain extent. These factors motivate us to develop a low-cost, real time driver's drowsiness detection system is developed with acceptable accuracy. In the developed system, a front camera of driver's cell phone records the video and drivers face is detected. For this, eyes, nose and mouth are detected to improve the area of interest using Google Vision API. For creating an app android studio alternative b4a is used. When the drowsiness is detected, then the driver is alerted by audio.

1.1 Objective

- It can be used for the safety of driver while driving which reduces accidents and detects drowsiness of the driver.
- To generate alerts to the driver by playing loud music.
- To capture video and detect drowsiness.
- To enhance the safety of the driver.
- To monitor driver fatigue and alert him when drowsiness situation is detected.
- To monitor driver attentiveness by ensuring hes keeping his eyes on the road.

1.2 Problem Definition

Today drowsy driving is a serious problem that leads to thousands of accidents each year. Motor vehicle collisions lead to significant death and disability as well as significant financial cost to both security and individual due to the driver impairments. Drowsiness is one of the factors for collisions.
In India, no monitoring device is used to measure the drowsiness of driver. Some kind of systems like driver fatigue monitor, real time vision based on driver state monitoring system, seeing driver assisting system, user center drowsiness driver detection and working system are implemented in foreign countries. All the systems focus either changes in eye movement, physiological measures or driver performance measure.
The main purpose of this project is to develop an application for Drowsiness detection of the driver. That's why this system is proposed for reducing accidents. Instead of using hard- ware components like webcam, we will be implementing on mobile app. We are implementing a system where the user has to first register in the app and then the app will capture the video of drivers face in frames and Google Vision API will be applied on these frames. When the new frames are captured calculation is done on the region of interest to determine the ratio of height and width of the region and on that basis detection is done and if the probability is less than 50% then alerts are send. We will be implementing this on B4A platform.

1.3 Technology Stack

- Google Vision API
  It allows developers to easily integrate vision detection features within applications, including image labelling, face and landmark detection.

- B4A
  Alternative of Android Studio. It describes as Rapid development tool. Used for application development.

2. Literature Review

Paper Title 1: Driver Drowsiness Monitoring System using Visual Behaviour and Machine Learning.

Authors: Ashish Kumar and Rusha Patra

Publication: 2018 IEEE

Description: In the developed system, a webcam records the video and drivers face is detected in each frame employing image processing techniques. The video is recorded using webcam (Sony CMU-BR300) and the frames are extracted and processed in a laptop. After extracting the frames, image processing techniques are applied on these 2D images. Presently, synthetic driver data has been generated. The volunteers are asked to look at the webcam with intermittent eye blinking, eye closing, yawning and head bending. The video is captured for 30 minutes duration. After extracting the frames, first the human faces are detected. Numerous online face detection algorithms are there. In this study, histogram of oriented gradients (HOG) and linear SVM method [10] is used. In this method, positive samples of fixed window size are taken from the images and HOG descriptors are computed on them. Subsequently, negative samples (samples that do not contain the required object to be detected i.e., human face here) of same size are taken and HOG descriptors are calculated. Usually the number of negative samples is very greater than number of positive samples. After obtaining the features for both the classes, a linear SVM is trained for the classification task. To improve the accuracy of SVM, hard negative mining is used. In this method, after training, the classifier is tested on the labeled data and the false positive sample feature values are used again for training purpose. For the test image, the fixed size window is translated over the image and the classifier computes the output for each window location. Finally, the maximum value output is considered as the detected face and a bounding box is drawn around the face. After detecting the face, the next task is to find the locations of different facial features like the corners of the eyes and mouth, the tip of the nose and so on. Prior to that, the face images should be normalized in order to reduce the effect of distance from the camera, non-uniform illumination and varying image resolution. Therefore, the face image is resized to a width of 500 pixels and converted to grayscale image. After image normalization, ensemble of regression trees is used to estimate the landmark positions on face from a sparse subset of pixel intensities. In this this method, the sum of square error loss is optimized using gradient boosting learning. Different priors are used to find different structures. Using this method, the boundary points of eyes, mouth and the central line of the nose are marked. After detecting the facial landmarks, the features are computed. After computing all the three features, the next task is to detect drowsiness in the extracted frames. In the beginning, adaptive thresholding is considered for classification. Later, ma- chine learning algorithms are used to classify the data. For computing the threshold values for each feature, it is assumed that initially the driver is in complete awake state. This is called setup phase. In the setup phase, the EAR values for first three hundred (for 10s at 30 fps) frames are recorded. Out of these three hundred initial frames containing face, average of 150 maximum values is considered as the hard threshold for EAR. The higher values are considered so that no eye closing instances will be present. If the test value is less than this threshold, then eye closing (i.e., drowsiness) is detected. As the size of eye can vary from person to person, this initial setup for each person will reduce this effect. Similarly, for calculating threshold of MOR, since the mouth may not be open to its maximum in initial frames (setup phase) so the threshold is taken experimentally from the observations. If the test value is greater than this threshold then yawn (i.e., drowsiness) is detected. Head bending feature is used to find the angle made by head with respect to vertical axis in terms of ratio of projected nose lengths. Normally, NLR has values from 0.9 to 1.1 for normal upright position of head and it increases or decreases when head bends down or up in the state of drowsiness. The average nose length is computed as the average of the nose lengths in the setup phase assuming that no head bending is there. After computing the threshold values, the system is used for testing. The system detects the drowsiness if in a test frame drowsiness is detected for at least one feature. To make this thresholding more realistic, the decision for each frame depends on the last 75 frames. If at least 70 frames (out of those 75) satisfy drowsiness conditions for at least one feature, then the system gives drowsiness detection indication and the alarm is given.

Drawbacks: Uses webcam which is external hardware and can be costlier and as well as requires high level image processing to focus the visuals

Paper Title 2: Eye Gaze Tracking Based Driver Monitoring System

Authors: Annu George Mavely, J.E Judith, Sahal P A, Steffy Ann Kuruvilla

Publication: 2017 IEEE

Description: In this paper a driver monitoring system using eye gaze technique is introduced. In this method, the first step is the tracking of human face from a real time video sequence to locate the eye region and to determine the number of times the user blinks his/her eye. The main focus is to use this eye tracking system to detect the drowsiness of the driver while driving thereby reducing the number of accidents happening day by day. In the proposed method Raspberry Pi 2 is soul of the framework, which control the total framework operation. Fig.1 describes the proposed methodology of driver monitoring system using eye gaze tracking. It is used as the image and general processing unit. In raspberry Pi Broadcom 700MHz Chip is used as main signal processing chip unit in which CPU is a 32 bit ARM11-RISC processor. This project uses a Logitech c270 HD webcam and LED IR illuminator for image capturing. A version of LINUX called Raspbian wheezy operating system is installed. It is a free operating system optimized specially for the hardware of Raspberry Pi. A 32GB MicroSD card is used to install OpenCV 3.2 on Pi board. Vibration motor is used to generate steering vibrations based on the algorithm. With the help of speaker or headphone or with any sound producing devices alarms are produced. For image capturing process uses a low-cost web camera. Camera is connected to the Pi board using USB port. In automobile vehicles camera is placed above the steering wheel pointing straight to the user. By placing camera in this manner helps to capture drivers face easily. An IR source is used to capture the image during night without causing any discomfort to the driver. For automatic on off purpose the LEDs are fitted with LDR. For monitoring the driver not only the eyes were detected but also eye gaze also calculated. The eye gaze is tracked from a real time video sequence. CAMSHIFT (Continuously Adaptive Mean Shift) algorithm is used for extracting the ROI (Region of Interest). The user's drowsiness is detected by determining whether the eyes were opened or not. The eye detection algorithm detects whether the eyes are opened this helps as in determining the user's drowsiness. In the frame when the users face is detected and eyes are not detected, assume that the user is falling asleep. When the eyes of the user is not detected more than a particular time period the user is alerted to wake up.

Drawbacks: This project uses Raspberry Pie which needs more power supply and uses sensors which increases the cost of the product.

Paper Title 3: An Accurate ECG Based Transportation Safety Drowsiness Detection Scheme

Authors: Kwok Tai Chui, Kim Fung Tsang, Hao Ran Chi, Bingo Wing Kuen Ling, and Chung Kit Wu

Published in 2016 IEEE

Description: In this paper, a driver drowsiness classifier (DDC) based on the ECG signals is developed. Also, a self-defined kernel is designed and implemented based on the optimal correlation analysis. The convolution kernel is fused with the cross-correlation kernel by a genetic algorithm. In the performance evaluation, the DDC obtains an overall accuracy of 97.01 percent, the sensitivity of 97.16 percent and the specificity of 96.86 percent. If either the convolution or the cross-correlation kernel is employed, then the performance of the classifier will be degraded by more than 10percent on average. The comparison results reveal that the performance is downgraded by 17 percent to 63 percent if the feature selection is performed by the typical kernels. It can be seen that the driver drowsiness detection via the ECG signals outperforms those based on the existing methods. Due to the high measurement stability and the high immunity to noise of the ECG signals compared to the image-based EEG and EOG signals, the ECG signals find more important applications in the market. Also, our proposed kernel improves the performance by 48.4 percent to 87.2 percent compared to related image-based methods and the biometric signal-based methods. The cross correlation based kernel matrix (Kxcorr,ij) captures the symmetric information among the ECG signals from different classes whereas convolution based kernel (Kconv,ij) captures the anti-symmetric information among the ECG signals from the same class. Hence, the KDDC can achieve a better performance than the prevailing kernels. This is because both the symmetric and anti-symmetric information of the ECG signals are highly related to the three driver states, namely the awake stage, the sleep stage 1 and the sleep stage 2. Also, the KDDC is adaptively designed and customized to the driver drowsiness detection application.

Drawbacks: It uses ECG module for heart rate which can be costlier.

Paper Title 4: A Multimodal System for Assessing Alertness Levels due to Cognitive Loading

Authors: Anwesha Sengupta, Anirban Dasgupta, Aritra Chaudhuri, Anjith George, Aurobinda Routray, Rajlakshmi Guha

Published in 2016 IEEE

Description: This paper proposes a scheme for assessing the alertness levels of an individual using simultaneous acquisition of multimodal physiological signals and fusing the information into a single metric for quantification of alertness. The system takes Electroencephalogram (EEG), high-speed image sequence and speech data as inputs. Certain parameters are computed from each of these measures as indicators of alertness and a metric by a fusion of the parameters for indicating alertness level of an individual at an instant is proposed. The scheme has been validated experimentally using standard neuropsychological tests such as the Visual Response Test (VRT), Auditory Response Test (ART), a Letter Counting (LC), and the Stroop Test. The tests are used both as cognitive tasks to induce mental fatigue as well as a tool to gauge the present degree of alertness of the subject. Correlation between the measures has been studied and the experimental variables have been statistically analyzed using measures such as multivariate linear regression and Analysis of Variance (ANOVA). Correspondence of trends obtained from biomarkers and neuropsychological measures validate the usability of the proposed metric. The work describes a system that combines information from multimodal data to determine the state of alertness of an individual. The system receives EEG, speech signals and high-speed image sequences through the Serial, Audio and Ethernet ports respectively, and calculates the alpha energy of EEG, voiced-unvoiced ratio of speech signals and peak saccadic velocity. The system has been cross-validated using neuropsychological measures of alertness in an experiment on sustained attention. The experiment makes use of a unique design of the auditory response test and uses the stimuli as both task and tool. To the best of knowledge of the authors, the present study is the first of its kind that uses a varied combination of biomarkers and neuropsychological parameters for analysis of alertness levels. Significant correspondence has been observed between the results of the biomarkers and results of neuropsychological parameters. The results of the study would hence be instrumental in developing a multi-modal engine to determine the state of alertness of an individual.

3. Proposed System Working

The Driver Monitoring is a new system based on a camera to monitor driver alertness, that not only recognizes the driver, but also checks his or her level of vigilance. The purpose of the Driver Monitoring system is to alert the driver when signs of drowsiness or distraction are detected. These developments will contribute to heightened safety and more intuitive use of the new generation of driver assistance functions.

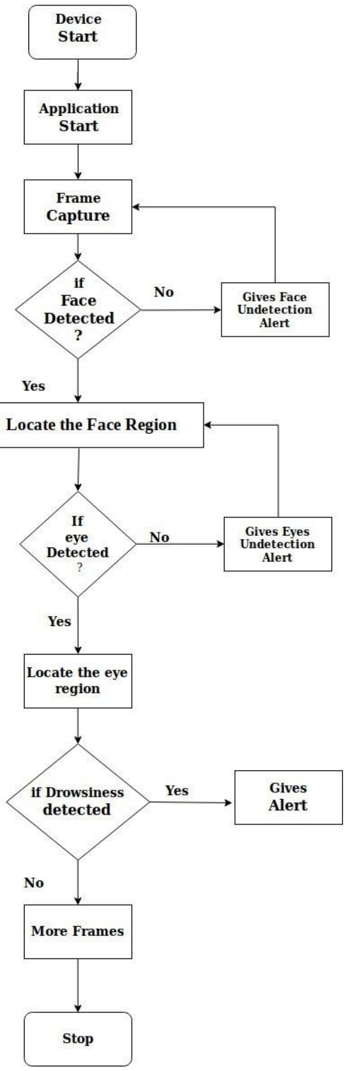


Figure 3.1: Driver Monitoring System Working

Initially, the driver device needs must be switched ON. The driver then starts the application. The proposed app on starting checks for the Face Dimension of Driver and if the driver face is not visible or detected will alert the driver about non-recognized face. Similar procedure takes for eye recognition and on successful recognition the algorithm starts processing the eye pattern and face pattern for DROWSINESS symptoms.

If the system suspects DROWSINESS symptoms will immediately alert the Driver by ringing the phone at full volume. If the symptoms are not detected will continue to monitor the frame unless the app is stopped by the driver.

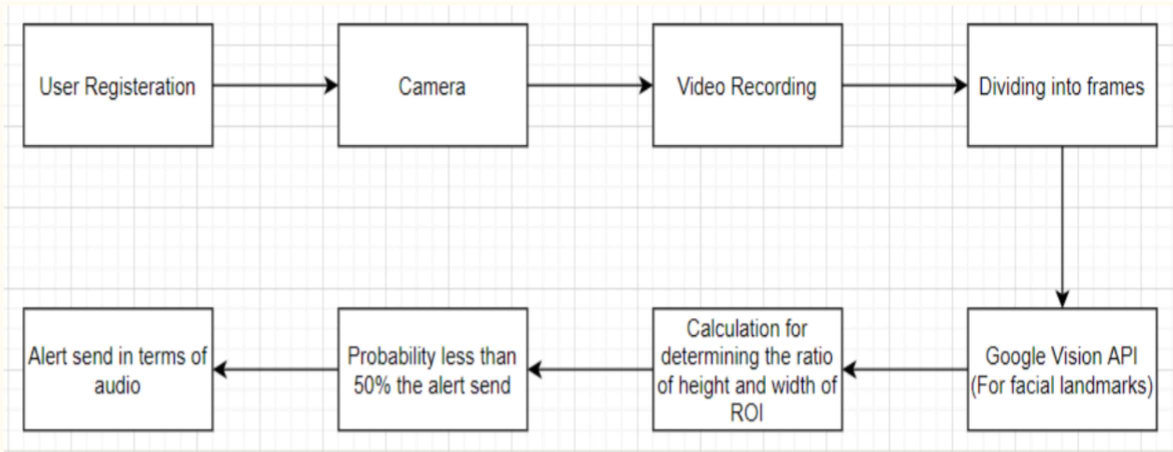


Figure 3.2: Flow of modules

Module 1: User Registration

In this, the user is able to login into the app. If the user visits for the first time then he/she has to register. After that he/she can login into the app and start the process. It has a feature, if the user forgets its password then an OTP will be sent to the registered mobile number and then the user is able to reset the password.

Module 2: Google Vision API

Google Vision API allows developers to easily integrate vision detection features within applications, including image labeling, face and landmark detection, optical character recognition (OCR), and tagging of explicit content.
Basically, this has been used for face detection and the facial points of the region of interest like eyes. The frames are captured every 5 seconds. Then the consecutive 2 frames are compared and if the two frames match then drowsiness detection alert is sent to the user.

4. Project Design

4.1 Use case Diagram

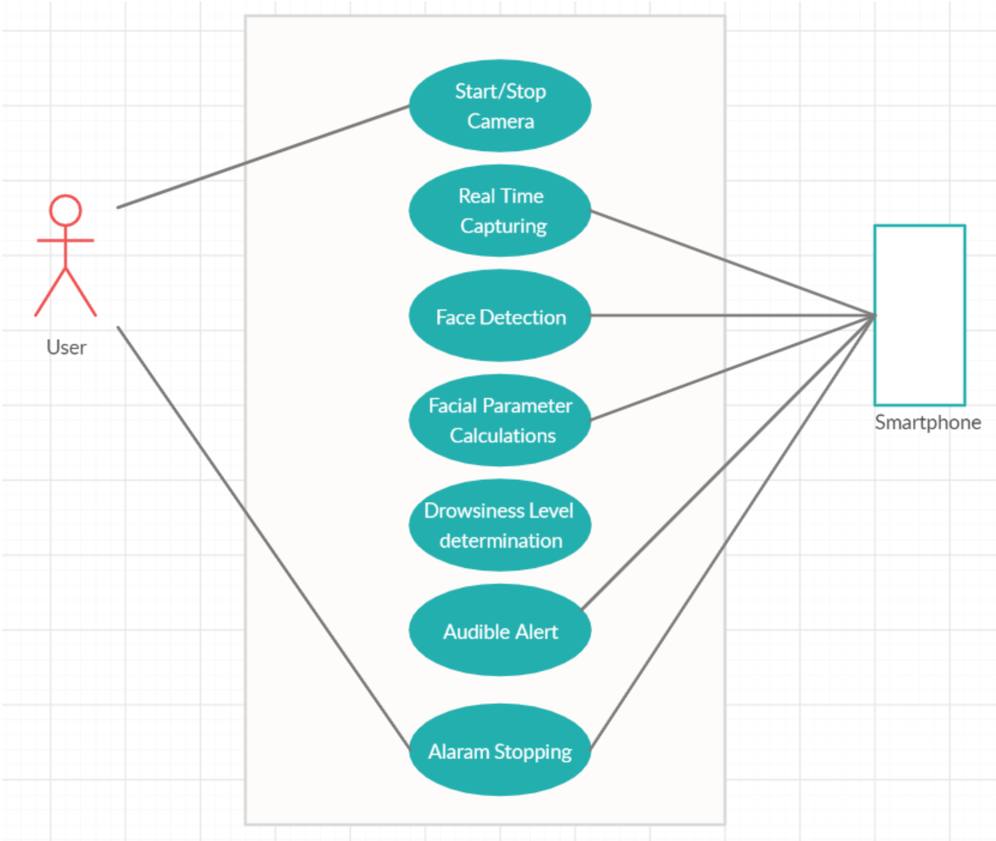


Figure 4.1: Use Case Diagram

In this use case diagram the functional requirements of the system are provided. User and smartphone are the actors, the one who interact with the use cases. Use cases are linked to the actors with communication link. Here the use case are inside the boundary of the system.

4.2 Activity Diagram

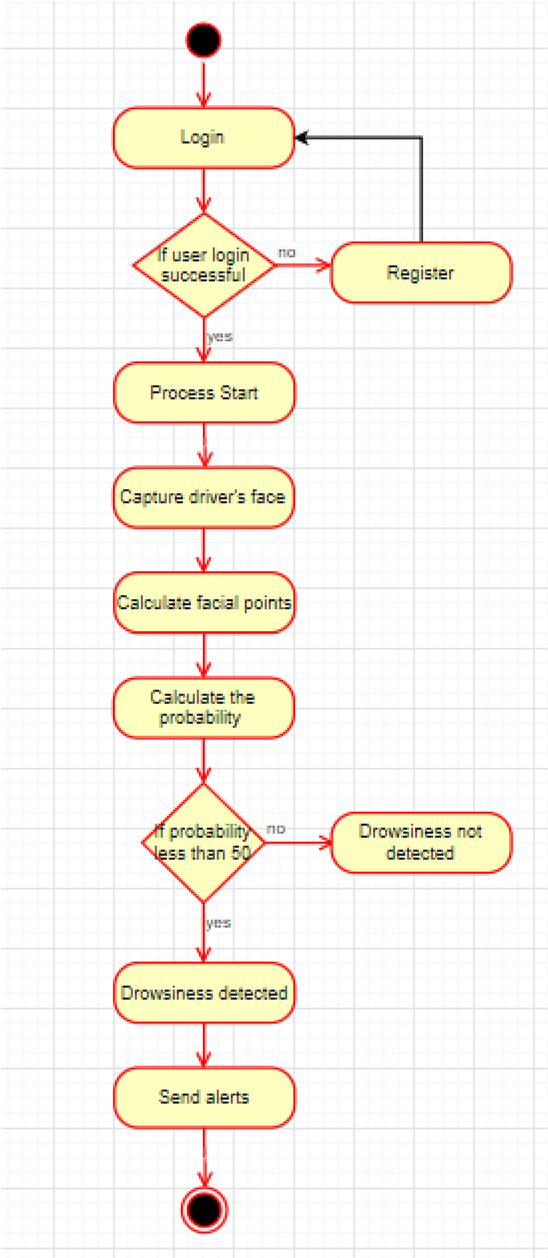


Figure 4.2: Activity Diagram

In this activity diagram we can easily understand the complete flow of the working. First, the user logs in the app then the process starts. Then frames are captured and calculation is done on facial landmarks. The probability is calculated and if it is less than 50% drowsiness is detected and alerts are sent.

4.3 Class Diagram

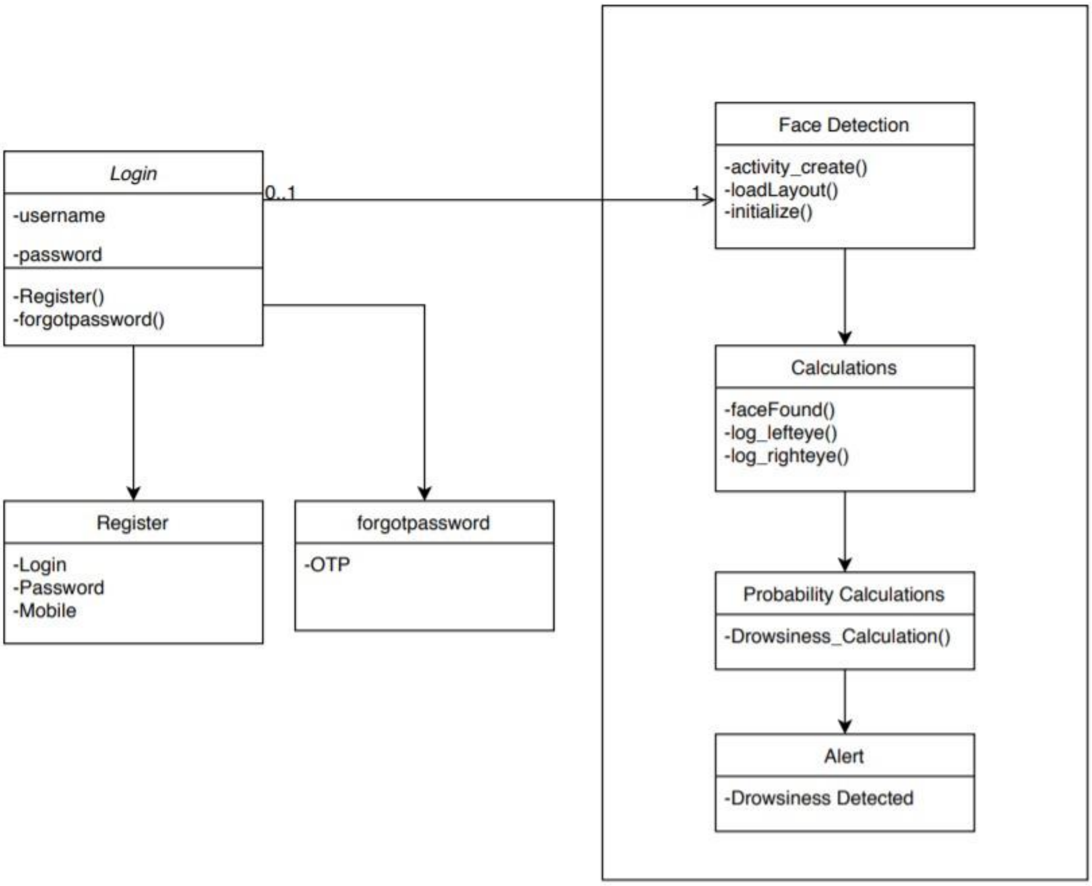


Figure 4.3: Class Diagram

In this class diagram we can see features provided to user. User logs in the app. If the user is new the user registers and if the users forgets the password via OTP user is able to change the password. Then the user is able to start the process. Frame is captured and drowsiness is detected on the basis of the probability of the facial points.

5. Project Implementation

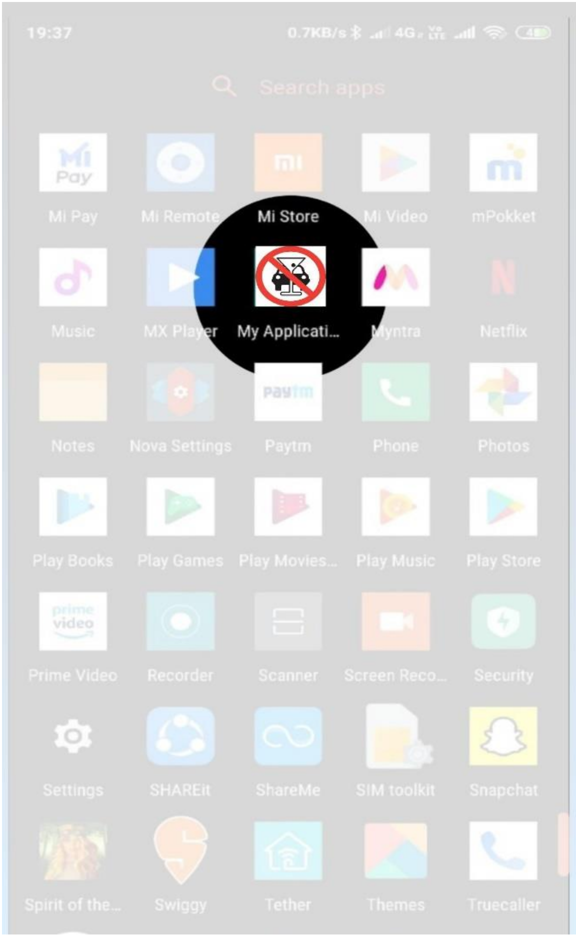


Figure 5.1: Application Icon

This is how the application will look like when the user will view it. This will help them to identify the app properly. Then the user can start the app manually and it will be guided through the whole process of drowsiness detection. The alerts given by the speech will signal the driver to drive carefully and not to feel fatigue.

```
#Region  Project Attributes
    #ApplicationLabel: Drowsiness Detection
    #VersionCode: 1
    #VersionName:
    'SupportedOrientations possible values: unspecified, landscape or portrait.
    #SupportedOrientations: portrait
    #CanInstallToExternalStorage: True
#End Region
#AdditionalRes: ..\resource
#BridgeLogger: True
#Region  Activity Attributes
    #FullScreen: False
    #IncludeTitle: False
#End Region

Sub Process_Globals
    'These global variables will be declared once when the application starts.
    'These variables can be accessed from all modules.
    Dim API_KEY1 As String
    Dim SQL1 As SQL
    Dim d_login As String
    Dim d_pass As String
    Dim d_mobile As String
    Dim d_mode As Int
    Dim Sms1 As PhoneSms
    Dim phone1 As PhoneCalls
    Dim d_use As Int
    Dim ran_no As Int
End Sub
```

Figure 5.2: Code snippet of main

The code snippet let us know about the orientation of the camera capture frame which is portrait and other permission are set. All the variables required for each activity and flow of the application is initiated here.

```
Sub cmdregister_Click
    If d_mode = 0 Then
        txtlogin.Text = ""
        txtpassword.Text = ""
        txtmobile.Text = ""
        d_mode = 1
        Label4.Visible = True
        Label4.Text = "Mobile:"
        txtmobile.Visible = True
        cmdlogin.Text = "Register"
        cmdcancel.Text = "Back"
        cmdforgot.Visible = False
        cmdregister.Visible = False
    Else
        d_mode = 0
        txtlogin.Text = ""
        txtpassword.Text = ""
        txtmobile.Text = ""
        Label4.Visible = False
        txtmobile.Visible = False
        cmdlogin.Text = "Login"
        cmdcancel.Text = "Cancel"
        cmdforgot.Visible = True
        cmdregister.Visible = True
    End If
End Sub
```

Figure 5.3: Code snippet for register

This function will be called when the register button is clicked.

```
Sub cmdlogin_Click
    If d_mode = 0 Then
        LogTable1
        If d_use = 0 Then
        If txtlogin.Text = d_login Then
            If txtpassword.Text = d_pass Then
                maps.u_name = d_login
                StartActivity(maps)
            Else
                Msgbox  ("Password Incorrect","Alert!")
            End If
        Else
            Msgbox  ("Login ID is Incorrect","Alert!")
        End If
        End If
    Else If d_mode = 1 Then
        If txtlogin.Text = "" Or txtpassword.Text = "" Or txtmobile.Text = "" Then
            Msgbox("Enter Proper Details","Alert!!!")
        Else
            SQL1.ExecNonQuery("INSERT INTO table3 VALUES('" & txtlogin.Text & "')")
            SQL1.ExecNonQuery("INSERT INTO table4 VALUES('" & txtpassword.Text & "')")
            SQL1.ExecNonQuery("INSERT INTO table5 VALUES('" & txtmobile.Text & "')")
            ToastMessageShow("Registration Completed.",False)
            d_mode = 0
            Label4.Visible = False
            txtmobile.Visible = False
            cmdlogin.Text = "Login"
            cmdcancel.Text = "Cancel"
            cmdforgot.Visible = True
            cmdregister.Visible = True
            txtlogin.Text = ""
            txtpassword.Text = ""
            txtmobile.Text = ""
```

Figure 5.4: Code snipped of Login function

```
Sub cmdforgot_Click
    If txtlogin.Text = ""  Then
        Msgbox("Enter Valid Username","Alert!!!")
        'Exit
    Else
        LogTable1
        If d_use = 0 Then
            If txtlogin.Text = d_login Then
                ran_no = Rnd(1000,9999)
                Log(ran_no)
                Sms1.Send(d_mobile,"OTP: " & ran_no)
                ToastMessageShow("OTP sent.",False)
                txtlogin.Text = ""
                txtpassword.Text = ""
                txtmobile.Text = ""
                Label2.Visible = False
                txtlogin.Visible = False
                Label3.Visible = False
                txtpassword.Visible = False
                cmdregister.Visible = False
                cmdforgot.Visible = False
                Label4.Visible = True
                txtmobile.Visible = True
                Label4.Text = "OTP:"
                ToastMessageShow("Enter OTP",False)
                d_mode = 2
                cmdlogin.Text = "Enter"
            Else
                Msgbox  ("Login ID is Incorrect","Alert!")
                'Exit
            End If
        End If
    End If

End Sub
```

Figure 5.5: Code snippet of forget password

This function will be called when the user forgets the password. The user is able to change the password via OTP.

```
Sub Activity_Create(FirstTime As Boolean)
    'Do not forget to load the layout file created with the visual designer. For example:
    'Activity.LoadLayout("Layout1")
    Activity.LoadLayout("1")
    p_status = 0
    p_drowsiness = 0
    VideoFileDir = rp.GetSafeDirDefaultExternal("")
    VideoFileName = "1.mp4"
    Activity.LoadLayout("1")
    Activity.LoadLayout("StillPicture")
    cam.Initialize(pnlCamera)
    Log(cam.SupportedHardwareLevel)
    buttons = Array(btnScene, btnAutoExposure, btnEffects, btnFocus, btnMode)
    SetState(False, False, VideoMode)
    sf.Initialize
    lv1.SingleLineLayout.Label.TextSize = 15
    lv1.SingleLineLayout.Label.TextColor = Colors.White
    lv1.TwoLinesLayout.Label.TextSize = 15
    lv1.TwoLinesLayout.Label.TextColor = Colors.white
    Label14.Text = "Welcome " & u_name
    If FirstTime Then

        SQL1.Initialize(File.DirDefaultExternal, "test1.db", True)
        GPS1.Initialize("GPS")

        SQL1.Initialize(File.DirDefaultExternal, "test1.db", True)


        Timer1.Initialize("Timer1", 5000)
        Timer2.Initialize("Timer2", 500)
        Timer1.Enabled = False
```

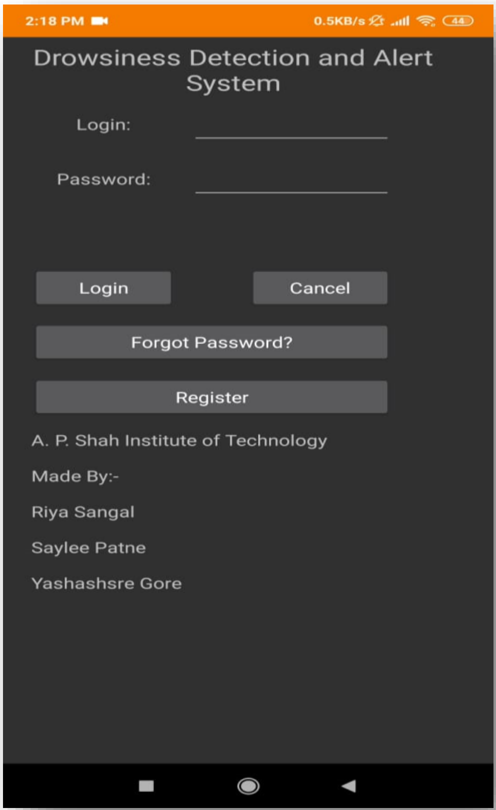
Figure 5.6: Code Snippet of the function when the app starts capturing the frames after every 5 seconds.

```
Sub fv1_face_found(LandMarks As Object)

    Dim mylist As List
    mylist = LandMarks
    Dim i As Int
    Dim left_eye As Int
    Dim right_eye As Int
    Dim happy As Int
    Dim lm As Int = sf.Val(mylist.Get(mylist.Size - 1))
    lv1.AddSingleLine("face ID = " & mylist.Get(mylist.Size - 2))
    For i = 0 To lm - 1

    Next
    Log (mylist.Get(i+2))
    Log (mylist.Get(i+3))
    Log (mylist.Get(i+4))
    Log (mylist.Get(i+6))
    left_eye = mylist.Get(i+2) * 100
    right_eye = mylist.Get(i+3) * 100
    happy = mylist.Get(i+4) * 100
    Log (left_eye)
    Log (right_eye)
    Log (happy)
    Label2.Text = left_eye & " %"
    Label3.Text = right_eye & " %"
    Label9.Text = mylist.Get(i+6)
    Label12.Text = happy & " %"
    If (left_eye < 50) Or (right_eye < 50) Then
        p_drowsiness = p_drowsiness + 1
        If p_drowsiness = 2 Then
            Label1.Text = "Drowsiness Status: Detected"
            TTS1.Speak("Alert! Drowsiness Detected!!!",False)
        Else
            Label1.Text = "Drowsiness Status: Not Detected"
        End If
    Else
        p_drowsiness = 0
        Label1.Text = "Drowsiness Status: Not Detected"
    End If

End Sub
```

Figure 5.7: Code snippet of calculation part

This code snippet let us know about the calculation of the facial landmarks of the left eye and right eye. The probability of both the eyes individually are compared to 50%. If the probability is less than 50% then the drowsiness is detected or less drowsiness is not detected.

Result



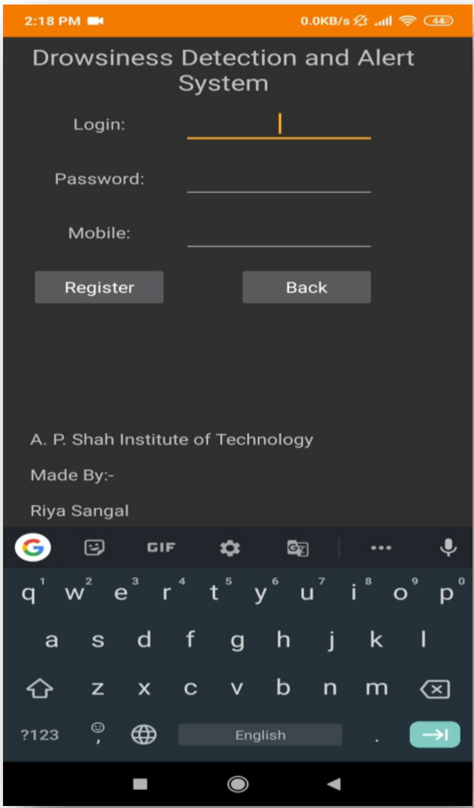


Figure 5.8: Register

Figure 5.9: Login

The first thing after opening the app is, user must login to its account. If user is new then he/she should register first. To register we have to enter the name and password and provide mobile number which is required if anyone forgets the name and password, one OTP will be send to the registered number and the person can login again.

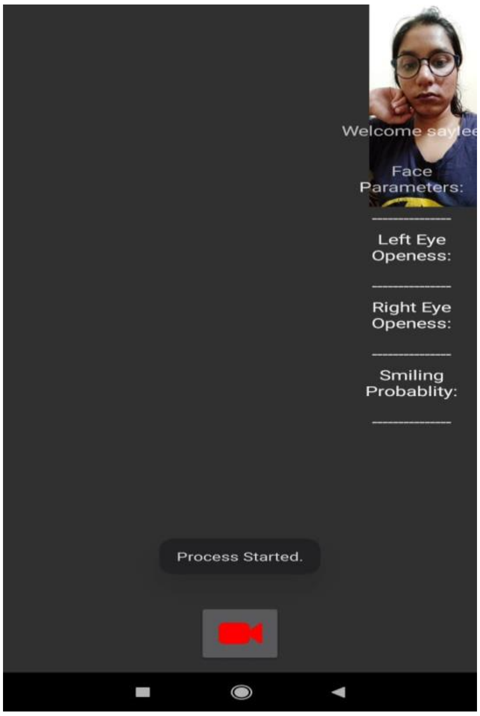


Figure 5.10: Process Started

Once the app opens, we need to click on the camera for allowing camera permission for the process to start. After giving permissions the process will get start after clicking the record icon on the bottom.

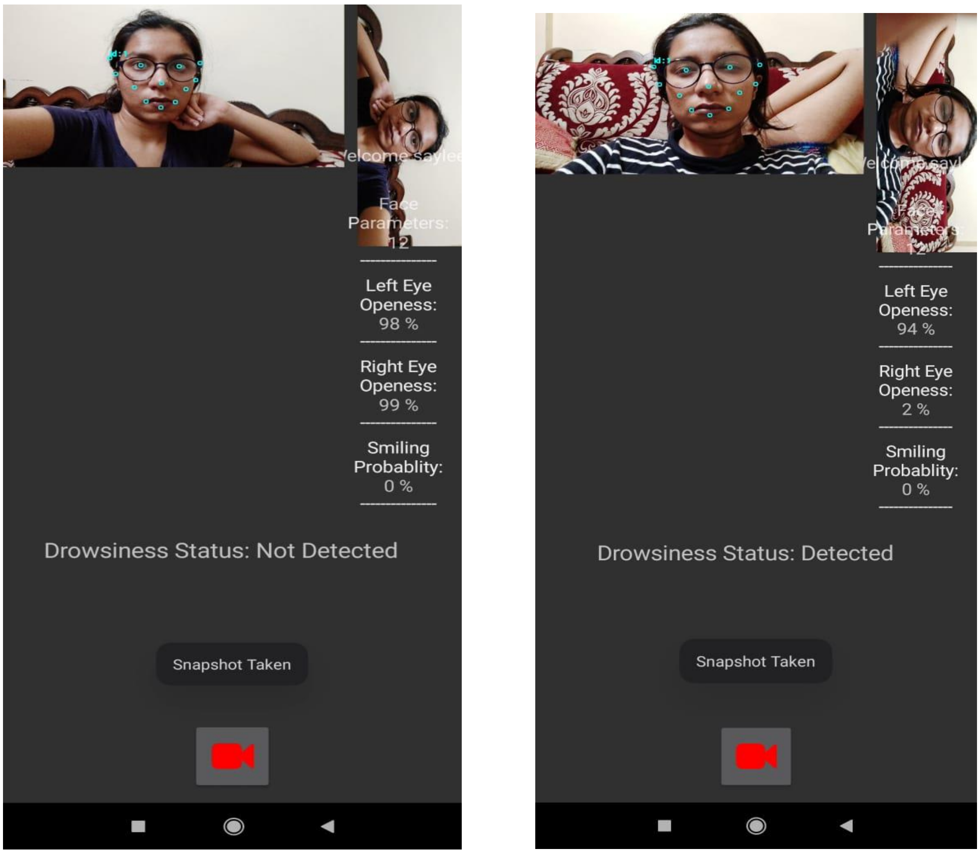


Figure 5.11: Output whether the drowsiness is detected or not detected using facial landmarks

6.Conclusion and Future Scope

6.1 Conclusion

In this we have proposed a mobile app which will be helpful in drowsiness detection. We have proposed a mobile app which will be helpful in drowsiness detection. The app captures the video of drivers face in frames and Google Vision API will be applied on these frames. No use of sensors.

Application will be free of cost; hence cost is very much effective. When the frames are captured and calculation is done on the region of interest (eg. eyes) to determine the ratio of height and width of the region and on that basis Detection is done and if the probability is less than 50% then alerts are sent.

6.2 Future Scope

- Using Pressure sensor on the steering alarm, Automatic braking System can be set in case of drowsiness.
- By using wireless Technology if the driver gets drowsy an alert message can be sent to a selected person mobile by using GSM module along with the alarm in vehicle.
- Development of a hybrid microcontroller for a vehicle which also consists of a alcohol detector which will sense is the driver is drunk and would not start the vehicle.

Bibliography

[1]Ashish Kumar and Rusha Patra, Driver Drowsiness Monitoring System using Visual Behaviour and Machine Learning, IEEE, 2018

[2] Annu George Mavely, J.E Judith, Sahal P A, Steffy Ann Kuruvilla, Eye Gaze Tracking Based Driver Monitoring System, IEEE, 2017

[3] Kwok Tai Chui, Kim Fung Tsang, Hao Ran Chi,Bingo Wing Kuen Ling, and Chung Kit Wu, An Accurate ECG Based Transportation Safety Drowsiness Detection Scheme, IEEE, 2016

[4] Anwesha Sengupta, Anirban Dasgupta, Aritra Chaudhuri, Anjith George, Member, IEEE, Aurobinda Routray, Rajlakshmi Guha, A Multimodal System for Assessing Alertness Levels due to Cognitive Loading, IEEE 2016

Acknowledgement

We have great pleasure in presenting the report on **DRIVER DROWSINESS AND ALERT SYSTEM.** We take this opportunity to express our sincere thanks towards our Guide Prof. **Anagha Aher** **&** Co. Guide Prof. **Geetanjali Kalme** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department, IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank Prof. **Anagha Aher** BE project coordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Saylee Patne: 16104021:

Riya Sangal: 16104014:

Yashashree Gore: 17204009: