**DRIVER ASSISTENT**

**M.R.P LANKATHILAKA**

**132028**

Major I: Computing and Information Systems

Major II: Electronics

Faculty of Applied sciences

Wayamba University of Sri Lanka

February, 2018

**DRIVER ASSISTENT**

**M.R.P LANKATHILAKA**

**132028**

Major I: Computing and Information Systems

Major II: Electronics

Faculty of Applied sciences

Wayamba University of Sri Lanka

February, 2018

Declaration

The dissertation is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student’s name …………………………… Date …………………………..

Signature …………………………………..

This is to certify that this dissertation is based on the work of Mr. /Mrs. /Ms. ………………….. Under my/our supervision. The dissertation has been prepared according to the format stipulated and is of acceptable and is of acceptable standard.

Certified by

Supervisor 1 Name ………………………. Date ……………………..

Signature ……………………

Supervisor 2 Name ………………………. Date ……………………...

Signature……………………

# 

# Abstract

Lack of concentration of the driver is one of the main reason of traffic accidents. This is a complex problem that involves due to phone calls, tiredness, drowsiness, etc. Therefore driver in alertness is a very important. In recent years, driver drowsiness has been one of the major causes of inattention. Many surveys indicate the reliable driver inattention detection system which could alert the driver before a mishap happens. Researchers have attempted to determine driver inattention using behavioral of driver face.

Image processing techniques to monitor the state of the driver is essential to identify drowsiness and tiredness. Driver state monitoring systems generally work based on driving patterns, driver’s video or physiological signals. Driver’s video or driving patterns are convenient to acquire, but to assess driver state accurately is difficult because these methods assess the driver state indirectly. Drowsy driver alert system can form the basis of the system to reduce the accidents. The purpose of such a system is to detect driver fatigue by placing the camera inside the car.

This work describes a system called “Driver Assistant” which has been developed to keep the driver vigilant always.

**ACKNOWLEDGMENT**

The special thanks go to my supervisor Mr. T.Arudchelvam, Senior Lecturer of the Department of Computing & Information Systems. The supervision and support that he gave truly assisted the progression and smoothness of the project development. The cooperation is indeed appreciated.

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my coordinator Mrs. T.Hansika Gunasekara, Department of Computing & Information Systems, Faculty of Applied Sciences, Wayamba University of Sri Lanka.

Table of Contents

[Abstract 4](#_Toc509090741)

[1. Introduction 8](#_Toc509090742)

[1.2 YML 12](#_Toc509090743)

[2. RELATED WORKS 12](#_Toc509090744)

[3. OVERVIEW OF THE SYSTEM 13](#_Toc509090745)

[4. METHOD 14](#_Toc509090746)

[4.1 Haar-Cascade development process 15](#_Toc509090747)

[4.2 Samples 16](#_Toc509090748)

[4.3 Creating the description file of positive samples 16](#_Toc509090749)

[4.4 Creating the description file of negative samples 17](#_Toc509090750)

[4.5 Packing the positive samples into a vec file 18](#_Toc509090751)

[4.6 Training the classifier 19](#_Toc509090752)

[4.7 Converting the trained cascade into a xml file 20](#_Toc509090753)

[4.8 Driver identification security system 20](#_Toc509090754)

[4.9 Micro-sleep detection system 21](#_Toc509090755)

[4.10 Assistant 23](#_Toc509090756)

[4.11 Map service 23](#_Toc509090757)

[4.12 Phone call system 26](#_Toc509090758)

[4.13 SMS service system 26](#_Toc509090759)

[4.14 MP3 player 28](#_Toc509090760)

[4.15 Radio system 29](#_Toc509090761)

[4.16 Calendar 30](#_Toc509090762)

[4.17 Settings 30](#_Toc509090763)

[5. Implementation 31](#_Toc509090764)

[5.1 Driver’s Assistant home page 31](#_Toc509090765)

[5.2 Assistant 39](#_Toc509090766)

[5.3 Anti Sleep 46](#_Toc509090767)

[5.4 Security 48](#_Toc509090768)

[5.4 Map 52](#_Toc509090769)

[5.4 SMS 55](#_Toc509090770)

[5.5 MP3 player 66](#_Toc509090771)

[5.6 Radio 68](#_Toc509090772)

[5.7 Calendar 71](#_Toc509090773)

[5.8 Settings 72](#_Toc509090774)

[6.Evaluation 79](#_Toc509090775)

[7.Result and Conclusion 81](#_Toc509090776)

[8.References 82](#_Toc509090777)

[[2]. Autonomous Robots Research Advances 82](#_Toc509090778)

**List of Figures**

Figure 1.0.1: micro-sleep detection…………………………………………………..10

Figure 1.1.1: Haar features…………………………………………………………...11

Figure 1.1.2: Training images………………………………………………………..12

Figure 3.0.1: System overview………………………………………………………14

Figure 4.0.1: System main menu…………………………………………………….16

Figure 4.3.1: Positive images for faces and eyes…………………………………….18

Figure 4.4.1: Negative images……………………………………………………….19

Figure 4.5.1: clipping positive images……………………………………………….19

Figure 4.6.1: While running the haartraning.bat……………………………………..21

Figure 4.8.1: driver identification……………………………………………………22

Figure 4.9.1: eye tracking to detect micro sleep……………………………………..23

Figure 4.9.2: micro sleep detection…………………………………………………..23

Figure 4.11.1: Map home page………………………………………………………25

Figure 4.11.2: Road map…………………………………………………………….25

Figure 4.11.3: Traffic jam map……………………………………………………...26

Figure 4.11.4: Satellite map…………………………………………………………26

Figure 4.13.1: Message service interface…………………………………………....29

Figure 4.14.1: MP3 player interface………………………………………………...29

Figure 4.15.1: Radio main interface………………………………………………...30

Figure 4.16.1: Calendar……………………………………………………………...31

Figure No 6.0.1: Functional Testing Process………………………………………..81

**List of Tables**

Table 4.9.0.1 – Accurate level between XML and YML……………………………22

Table 4.9.0.2 – Detection level between XML and YML……………………..……76

Table 6.0.2:Functional testing for Driver assistant………………………………… 76

# 1. Introduction

Drivers who cause fatal road traffic accidents when they fall asleep at the wheel .These type of sleep is called as a micro sleep. Micro sleep is a short burst of sleep that can be experienced by anyone. It can even occur without the person being aware that it happened. During a micro sleep, a person can lose attention and fall asleep. This can be just for a few seconds or even up to half a minute. Micro sleep usually occurs when a person is tired or doing monotonous work, such as driving on a motorway. It usually happens during the times that our biological clock wants us to rest, such as at night and mid-afternoon. Scientists are still unsure what happens in our brains during micro sleep. However, some studies have shown that parts of the brain fall asleep, while others stay awake. Therefore, when we are tired our brain wants to sleep, so it starts to shut down, which causes micro sleep. [6]



Figure 1.0.1: micro-sleep detection

The majority of vehicles involved in accidents due to micro sleep were dual purpose vehicles, lorries and cars, while the most vulnerable times are 1-5 a.m., 1-2 p.m., 5-6p.m. and 11-12 midnight. It has been estimated that 16.5 to 17.8 percent of traffic accidents are fatigue related adding that driving under sleep deprivation [2] There were over 2600 deaths on roads in 2010.This trend is rising since it is 300 more than the deaths recorded in 2009[8].

Micro sleep is very common, the driver continues with his previous activity without interruption and usually drives along without any incident. However, if an oncoming vehicle cuts into opposite lane, or if the driver is negotiating a curve, or if the vehicle in front suddenly slows or stops, the normal reflexes are absent to react fast enough and an accident is the result. It is noteworthy that such fatigue induced accidents have some common features. [8]

To prevent these types of asleep, image processing technique is very useful. Because of Scientists are still unsure what happens in our brains during micro-sleep. So that efficient way to detect micro sleep is tracking eyes.

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps:

• Importing the image via image acquisition tools;

• Analyzing and manipulating the image;

• Output in which result can be altered image or report that is based on image analysis.

**1.1 Knowledge Base**

Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.[3]

Here I work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.[3]

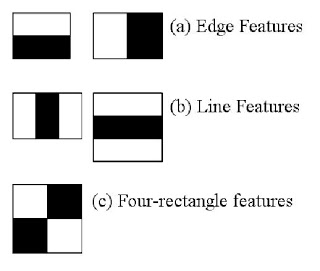


Figure 1.1.1: Haar features

Now all possible sizes and locations of each kernel is used to calculate plenty of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. [3]

But among all these features calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. It is achieved by Adaboost. [3]

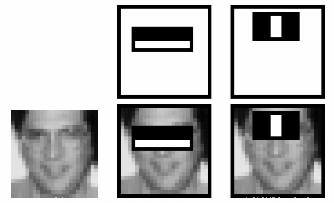


Figure 1.1.2: Training images

For this, apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. Then select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).[4]

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier.

## 1.2 YML

YML is a human-readable data serialization language. It is commonly used for configuration files, but could be used in many applications where data is being stored. YML targets many of the same communications applications as XML. Custom data types are allowed, but YML natively encodes scalars . These data types are based on the Perl programming language, though all commonly used high-level programming languages share very similar concepts. YML supports both Python-style indentation to indicate nesting.[9]

# 2. RELATED WORKS

There have been several studies about motor vehicle driver fatigue. There are even some commercial systems in the market that detect and signal driver unawareness. Most of these systems monitor driver's steering patterns. When these patterns are unusual, a warning signal is issued. Facial feature extraction is an active research area in computer vision. Yoo proposed an approach that takes into account facial symmetry in order to find the placement of the face in an image. Sophisticated methods such as Eigen space matching, contour tracking using snakes, deformable template, and ellipse fitting techniques are computationally very expensive. Skin color based detection is a widely used technique to extract face region from background images. Eye tracking is also being widely studied due to its potential applications in multi-modal user interfaces.

# 3. OVERVIEW OF THE SYSTEM

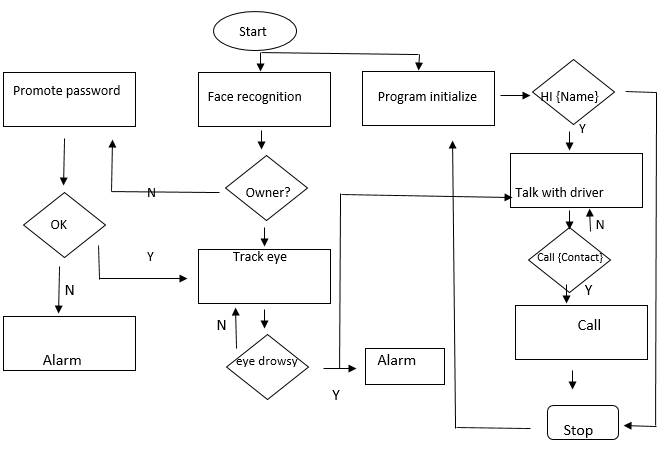


Figure 3.0.1: System overview

This system is an AI bot. It observes driver by tracking his face before start driving. If it fails to identify, then promotes a password. Driver should say the correct password within a specific time .The password does not correct, then grant an alarm. If it identifies driver correctly, then it will be tracking driver’s eye to detect micro sleep while driving. If all conditions satisfied, driver can proceed the driving safely. If the system able to detect micro sleep then automatically grant an alarm. Driver can preserve concentration by interacting (chatting) with the system to prevent feel sleepy. The system also provides telephone call and play music facilities according to the voice command of the driver while driving.

# 4. METHOD

In this section, the detail configurations of the system are described. The system evaluates the trend of driver’s drowsiness using image processing, and determines the micro sleep based on detecting eye state.

There are various function in Driver’s assistant system. These function can be generally divided into the following categories:

* Assistant
* Anti-sleep
* Security
* Map
* Voice call
* SMS
* MP3 player
* Radio
* Calendar
* Settings

This system mainly focus on preventing accident due to driver’s low concentration. For that Anti sleep, Voice call and SMS function are used. Assistant, MP3 player and Radio function can be used to provide entertainment for the driver. Using Map button, driver can get direction to the destination, information of traffic jam and near places. Calendar button can be used to view dates.

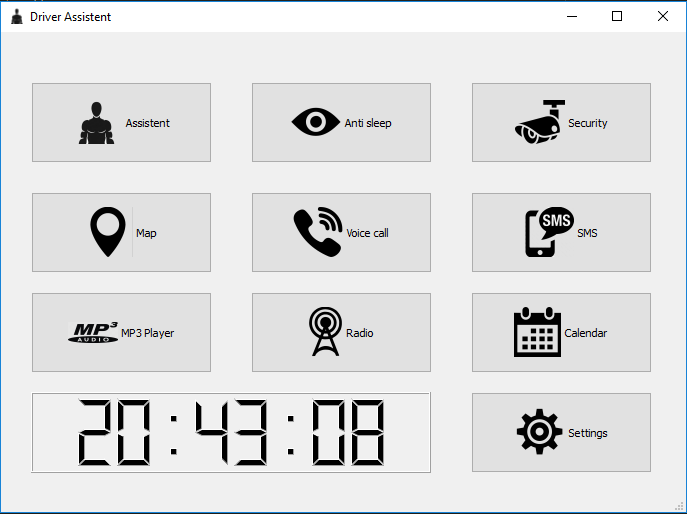


Figure 4.0.1: System main menu

## 4.1 Haar-Cascade development process

1. Creating the description file of positive samples

2. Creating the description file of negative samples

3. Packing the positive samples into a vec file

4. Training the classifier

5. Converting the trained cascade into a xml file

## 4.2 Samples

In order to train own classifier needs samples, which means need a lot of images that show the object we want to detect (positive sample) and even more images without the object (negative sample). The numbers depend on a variety of factors, including the quality of the images, the object want to recognize, the method to generate the samples, the CPU power is most effective factor for this. Training a highly accurate classifier takes a lot of time and a huge number of samples.

## 4.3 Creating the description file of positive samples

This is where I gathered about 20 images of human’s faces. I download around 20 images and cropped face images were available at there. I thought training with such images would generate a face detector which is robust to facial pose.

I needed around 20 images which I can use to generate positive samples OpenCV can work with. It’s also important that they should differ in lighting and background. **Once we have the pictures, we need to crop them so that only our desired object is visible**. Keep an eye on the ratios of the cropped images, they shouldn’t differ that much. The best results come from positive images that look exactly like the ones want to detect the object in, except that they are cropped so only the object is visible.

****

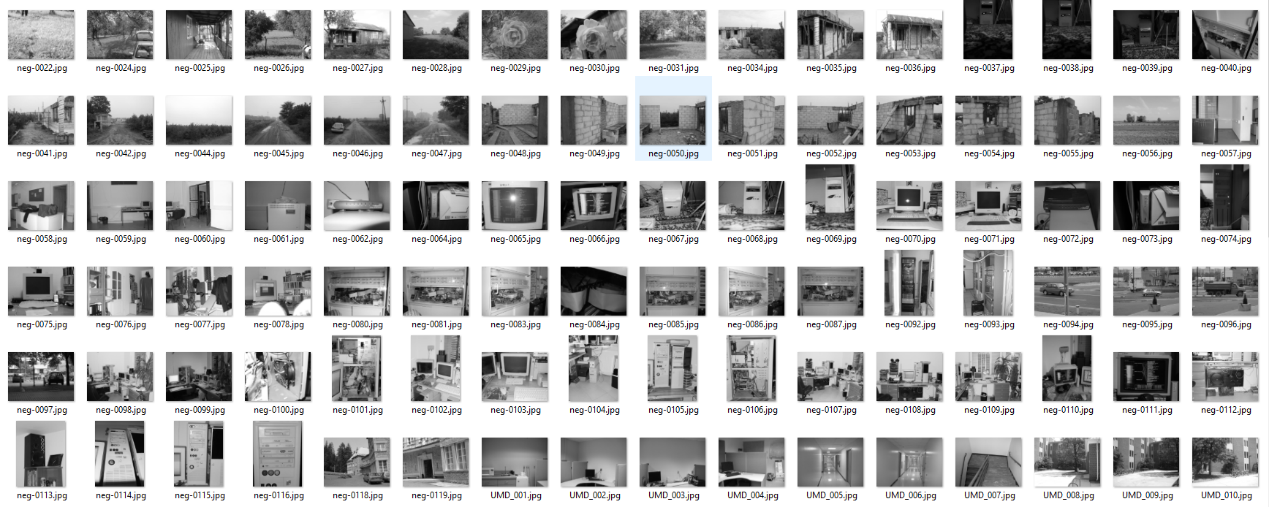
****

Figure 4.3.1: Positive images for faces and eyes

## 4.4 Creating the description file of negative samples

Then I needed the negative images, the ones that don’t show a faces and eyes. In the best case, if I were to train a highly accurate classifier, I had a lot of negative images that look exactly like the positive ones, except that they don’t contain the object I want to recognize.

For getting better result needs at least **200** of them.

****

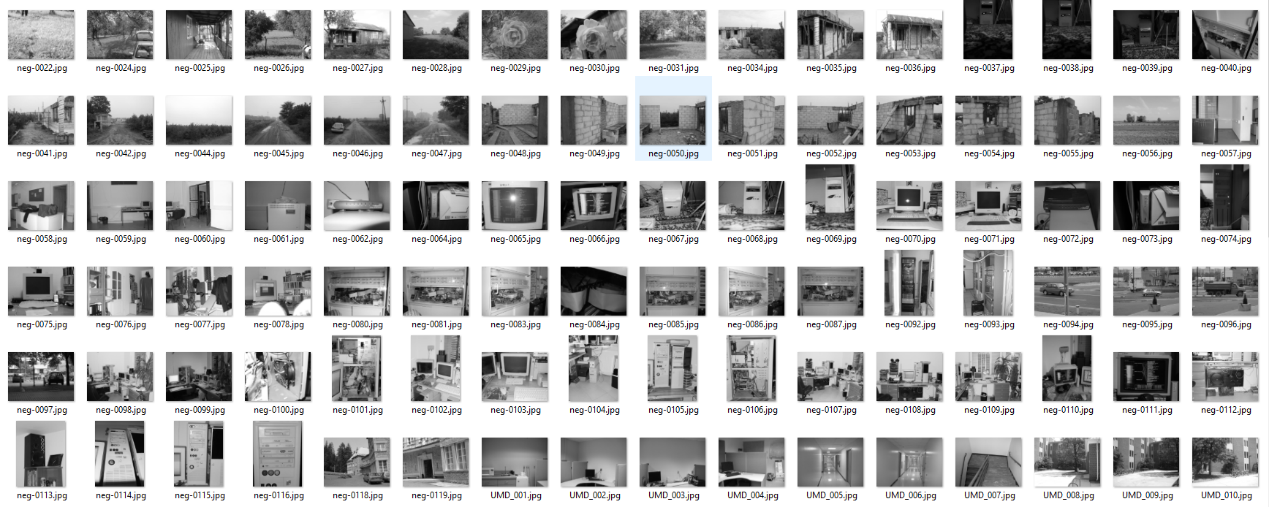
****

Figure 4.4.1: Negative images

## 4.5 Packing the positive samples into a vec file

In this step needs to create a data file (vector file) that contains the names of positive images as well as the location of the objects in each image. We can create this file via two utilities: Objectmarker or Image Clipper. [11]

The first one is simpler and faster, and the second one is a bit more versatile but more time consuming to work. I tried with Objectmaker.

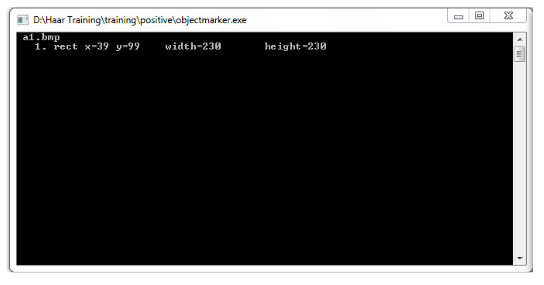
****

Figure 4.5.1: clipping positive images

Finally merge all backups into a final info.txt

Within the info.txtthere would be some information like below:

rawdata\image1200.bmp 1 34 12 74 24

rawdata\image1201.bmp 3 35 25 70 39 40 95 80 92 120 40 45 36

rawdata\image1202.bmp 2 10 24 90 90 45 68 99 82

The first number in each line defines the number of existing objects in the given image. For example, in second line, the number 3 means that already selected three objects (e.g. face) within image1201.bmp. The next four numbers (35 25 70 39) defines the location of first object in the image (top left vertex: x=35, y=

24, width=70 and height=39). The next numbers (40 95 80 92) identifies the data for the second object; Next ones (120 40 45 36) are for the third object.[11]

The content of the bath file is:

createsamples.exe -info positive/info.txt -vec vector/facevector.vec -num

200 -w 24 -h 24

Main Parameters:

-info positive/info.txt Path for positive info file

-vec vector/facevector.vec Path for the output vector file

-num 200 Number of positive files to be packed in a vector file

-w 24 Width of objects

-h 24 Height of objects

Then batch file loaded info.txt and packed the object images into a vector file. After running the batch file, got facevector.vec file.[11]

## 4.6 Training the classifier

haartraining.exe -data cascades -vec vector/vector.vec -bg negative/bg.txt

-npos 200 -nneg 200 -nstages 15 -mem 1024 -mode ALL -w 24 -h 24 –nonsym

-data cascades Path and for storing the cascade of classifiers

-vec data/vector.vec Path which points the location of vector file

-bg negative/bg.txt Path which points to background file

-npos 200 Number of positive samples ≤ no. positive bmp files

-nneg 200 Number of negative samples (patches) ≥ npos

-nstages 15 Number of intended stages for training

-mem 1024 Quantity of memory assigned in MB

-mode ALL Look literatures for more info about this parameter

-w 24 -h 24 Sample size

-nonsym Use this if subject is not horizontally symmetrical

Harrtraining.exe collects a new set of negative samples for each stage, and –nneg sets the limit for the size of the set. It uses the previous stages’ information to determine which of the "candidate samples" are misclassified. [11]

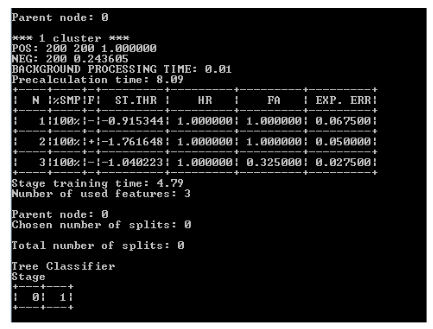


Figure 4.6.1: While running the haartraning.bat

## 4.7 Converting the trained cascade into a xml file

After finishing Haar-training step, catalogues were named from “0” up to “N-1”in which Nis the number of stages already defined in haartraining.bat. In each of those catalogues there should be AdaBoostCARTHaarClassifier.txtfile. Copy all the folders 0..N-1into the data folder.

Then combined all created stages (classifiers) into a single XML file which final file a “cascade of Haar-like classifiers”. Then run the batch file convert.bat

haarconv.exe data myfacedetector.xml 24 24

myfacedetecor.xml is the output file name and 24 24 are W and H respectively. [11]

## 4.8 Driver identification security system

****

Figure 4.8.1: driver identification

This tracks the driver face to detect whether the driver is an unauthorized person or not. If it detects the person as authorized then, proceeds to next process (eye tracking), otherwise system grants an alarm. When system unable identify the owner, promote a voice password. If it is correct, then proceeds to eye tracking process otherwise grant an alarm as mentioned above.

## 

## 4.9 Micro-sleep detection system

This function able to detect the iris of the driver by using image processing technique. If it identifies the driver is in micro sleep, then redraw the driver’s attention.

It is not the right way of detecting the driver’s eyes when close. There is a considerable time gap between eye open and close. Therefore, the detection of micro sleep can reduce the risk of accident rather than detection of anti-sleep.

Use of the height of open eye is commonly used to detect the micro-sleep. But the height of the open eye is differed from person to person. Therefore, height of the micro sleep is not a constant value. According to that, this methodology is not efficient. Using set of micro sleep image samples can overcome the above drawback. These samples also differ from one to another. Therefore the system can clearly identify different micro sleeps using YML.

YML is much less verbose. This makes it subjectively easier to read and edit for many people. On the flip side, it's slightly harder to parse.

The biggest difference, though, is that XML is meant to be a markup language and YML is really more of a data format. Representing simple, hierarchical data tends to be more gracefully done in YML, but actual marked-up text is awkward to represent. [10]

Table 4.9.0.1 – Accurate level between XML and YML

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | XML(60s) | XML(60s) |  | YML(60s) | YML(60s) |  |
|  | Correct | False | Recall | Correct | False | Recall |
| Open eye detection | 15 | 8 | 65.21% | 20 | 6 | 76.92% |
| Close eye detection | 12 | 7 | 63.16% | 18 | 7 | 66.66% |
| Face detection | 22 | 5 | 81.48% | 24 | 4 | 85.71% |

Results are shown in Table 01. We can see that the accuracy and the efficiency of YML is better than the XML, when specified time limit (60s) is used. The accuracy of both XML and YML have been analyzed by giving the random times. Within 60 seconds there are several detections that differ from XML to YML.

Table 4.9.0.2 – Detection level between XML and YML

|  |  |  |
| --- | --- | --- |
|  | XML | YML |
| Open eye detection | 23 | 26 |
| Close eye detection | 19 | 25 |
| Face detection | 27 | 28 |

 Results are shown in Table 02. We can determine that detection level of YML also higher than XML.

Figure 4.9.1: eye tracking to detect micro sleep



Figure 4.9.2: micro sleep detection

## 4.10 Assistant

This function helps to keep concentration of the driver by interact with the system. This system reacts with the driver’s voice commands. Driver can play music, check date and time etc.

Assistant conducts a [conversation](https://en.wikipedia.org/wiki/Conversation) via auditory methods. This system designed to convincingly simulate how a driver would behave as a conversational partner, Assistant use in [dialog systems](https://en.wikipedia.org/wiki/Dialog_system) for keep concentration. In here use [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords.

## 4.11 Map service

Driver can use this function to find a way to a destination with little effort. Map service provides the layout of roads, the locations of cities and towns, state boundaries, geographical features, restaurant reviews and satellite images. This function provides the Street View perspective, allowing driver to see houses, storefronts and points of interest from a driver's point of view.

This Service mainly divided into three section. There are Map, Traffic and Satellite. Map function can be used to get direction to the destination. Satellite function can be used to see real view of the road. Traffic function can be used to check traffic jam.

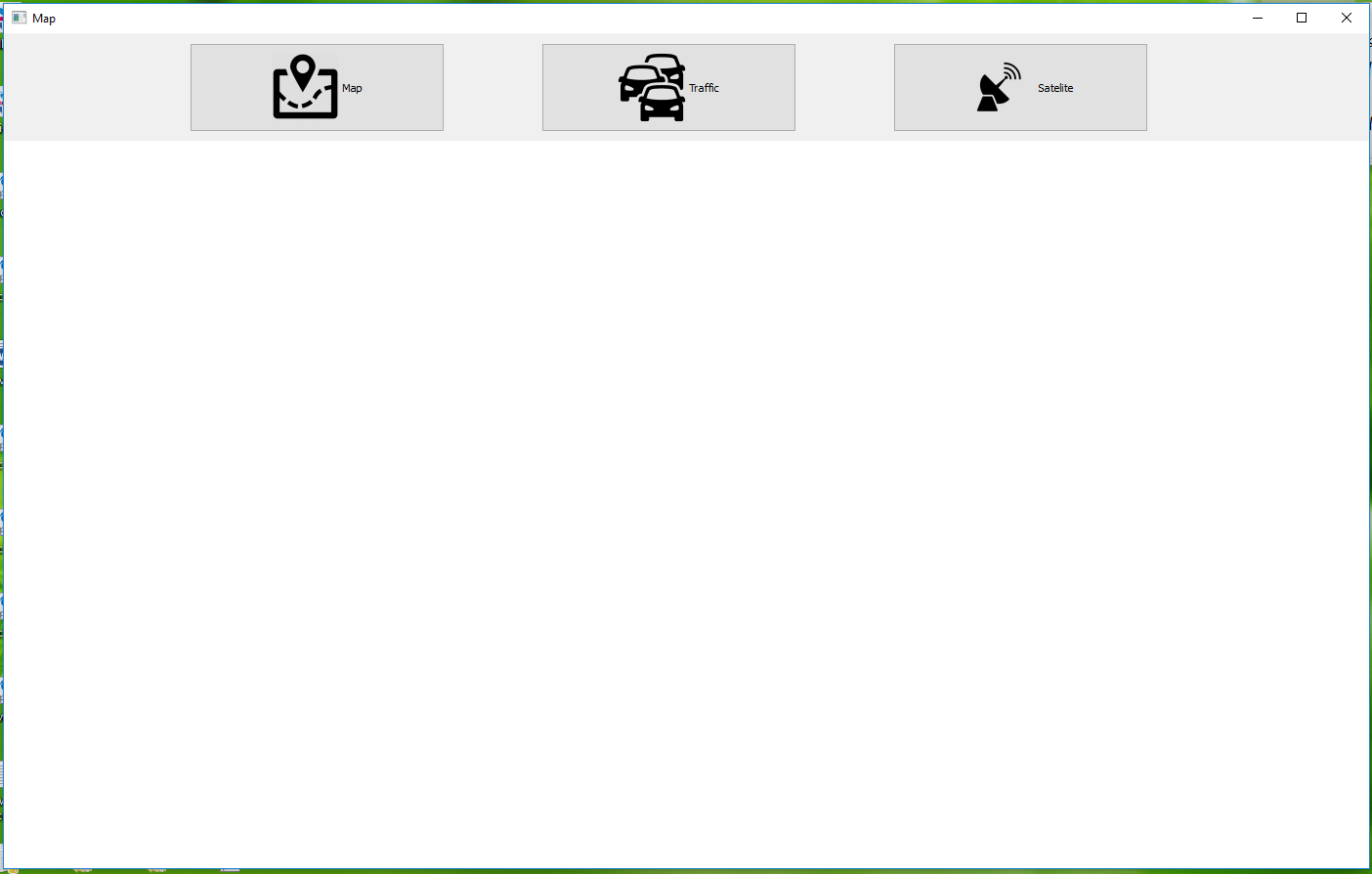


Figure 4.11.1: Map home page

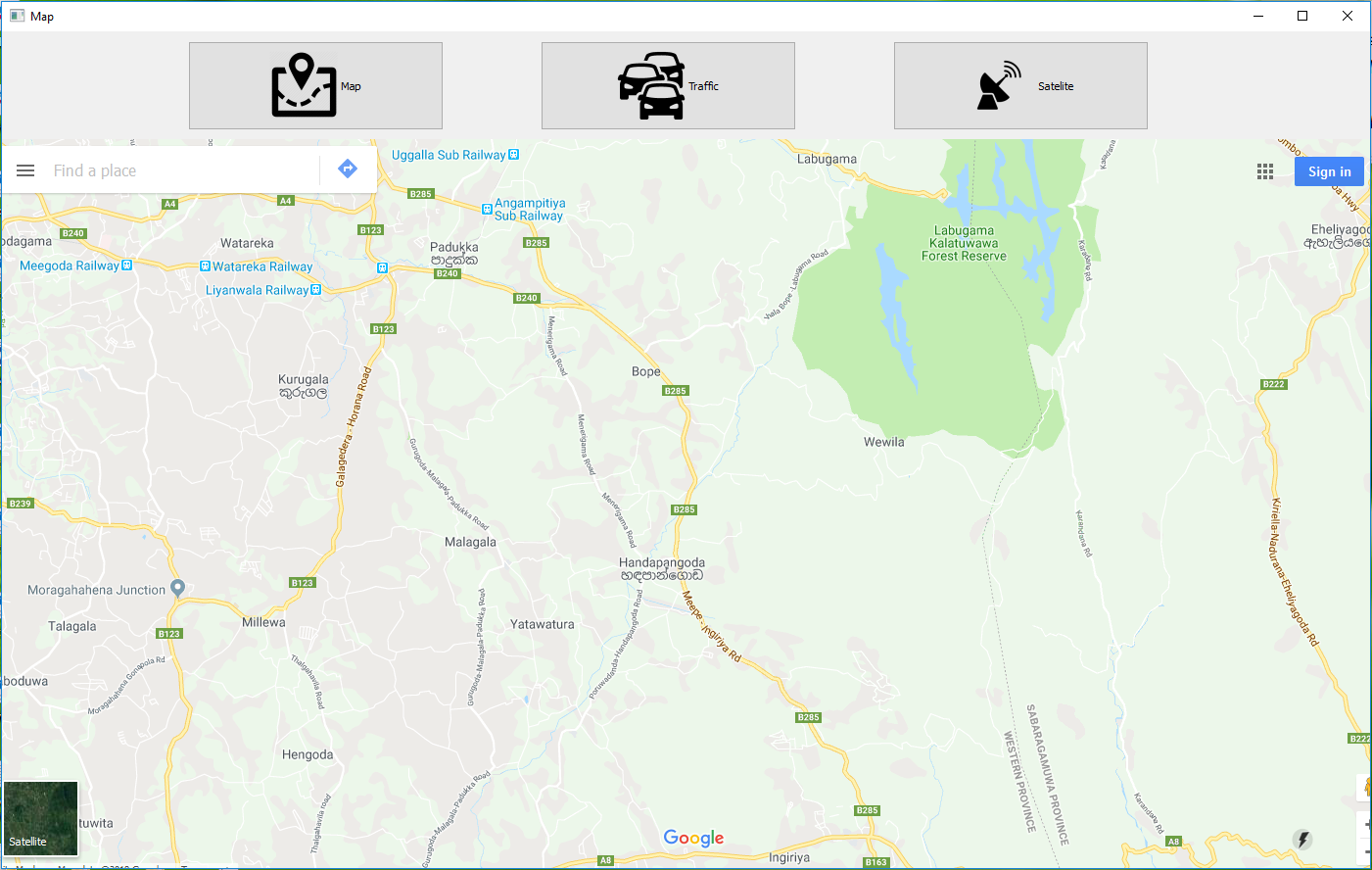


Figure 4.11.2: Road map

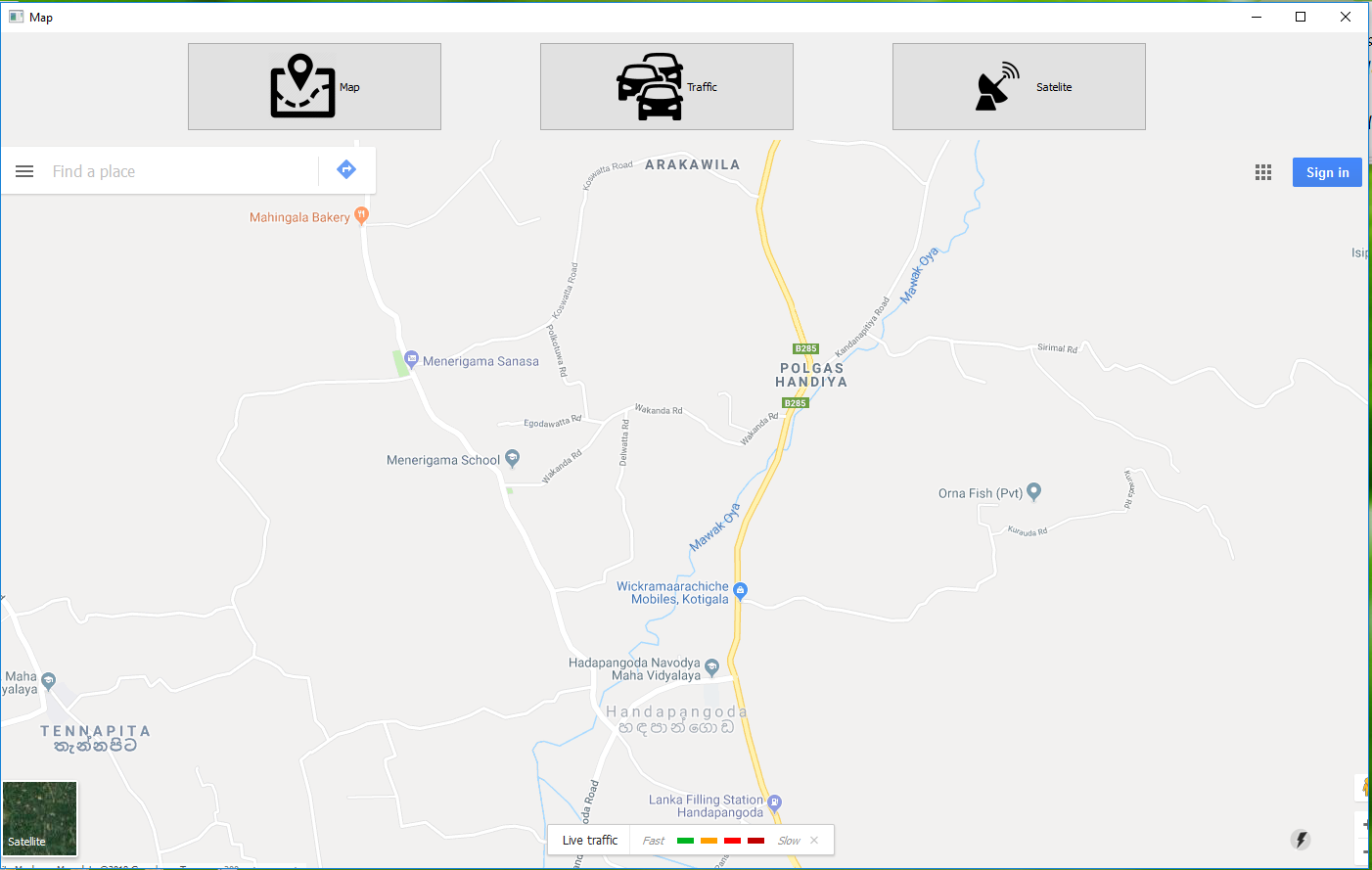


Figure 4.11.3: Traffic jam map

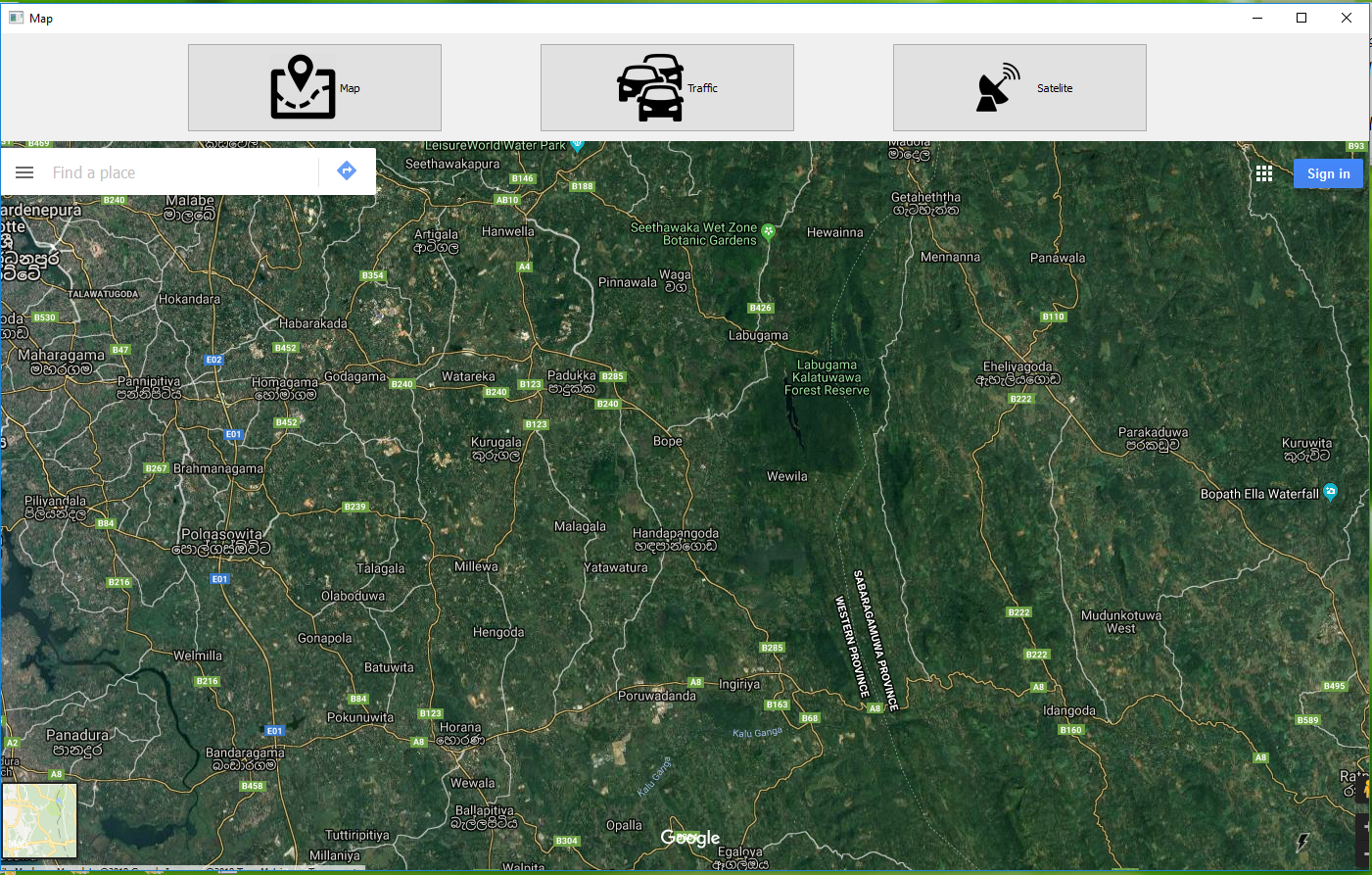


Figure 4.11.4: Satellite map

## 4.12 Phone call system

Driver can get phone call using voice commands. This system grants hand free calling. There are six buttons in the interface. Out of these six buttons five buttons have contacts. Using these buttons driver can directly call to the related contacts since the contacts are saved in those buttons. After one of those five buttons has clicked, driver can hear a beep sound. Then the driver can respond with the call.

When “other” button (which doesn’t have a contact number) is clicked, system ask for the phone number, promoting a voice message “Please tell the phone number”. After that, driver can hear a beep sound. Then the driver can respond with the call.

## 4.13 SMS service system

This SMS service can be used to send messages using voice commands. There are six buttons in the interface. Out of these six buttons five buttons have contacts. Using these buttons driver can directly send messages to the related contacts since the contacts are saved in those buttons. After one of those five buttons has clicked, it promotes a voice message “Please tell the message, you want to send”. After that, driver can hear a beep sound. Then the driver can respond with the message that he wants to send. Then the system converts the vocal message in to a text and send the text to the related contact number.

When “other” button (which doesn’t have a contact number) is clicked, system ask for the phone number, promoting a voice message “Please tell the phone number”. After that, driver can hear a beep sound. Then the driver respond with the phone number to the system. After that system promotes a voice message “Please tell the message, you want to send”. After that, driver can hear a beep sound. Then the driver can respond with the message that he wants to send. Then the system converts the vocal message in to a text and send the text to the related contact number.

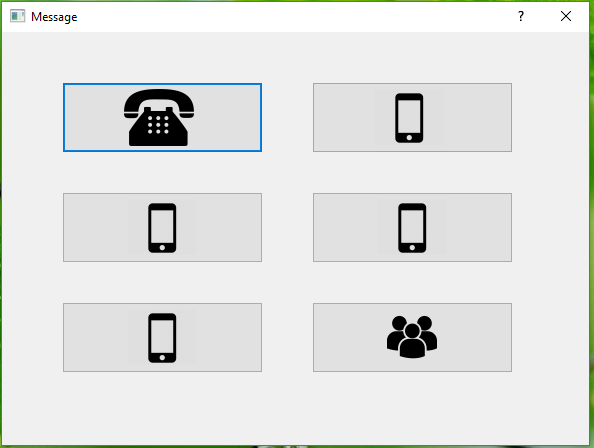


Figure 4.13.1: Message service interface

## 4.14 MP3 player

This function can be used to play music. This function also provide in Assistant mode. The different is, in here user can choose desired music files.  This music player also acts according to the voice commands. When driver speak to the system as “play music” music player will be launched.

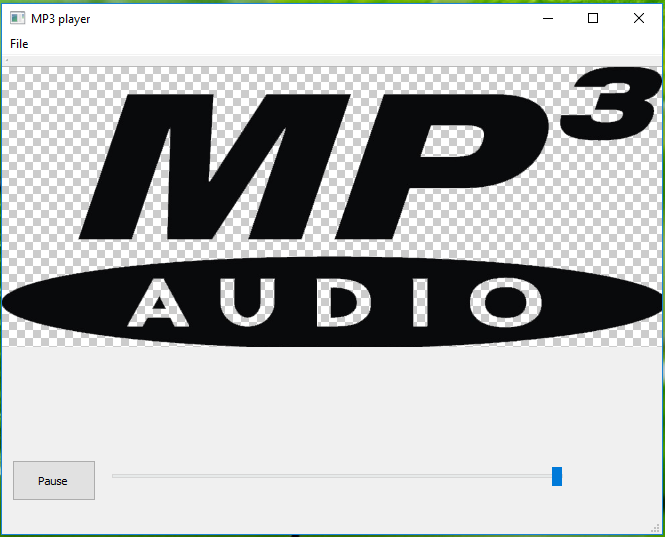
****

Figure 4.14.1: MP3 player interface

## 4.15 Radio system

Usually there is a radio player in vehicle, but driver must adjust the FM frequency manually to listen to the radio properly. During the bad weather cannot listen radio properly. Often unclear and is affected by whether .To overcome these type of obstacles use this radio system. When driver click on the button which he want to adjust, automatically connect to that FM frequency and play.

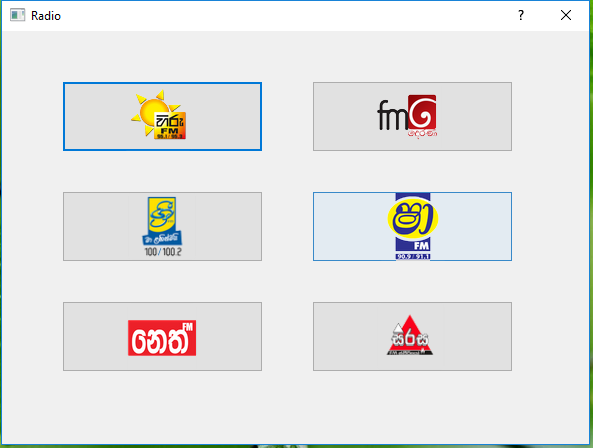
****

Figure 4.15.1: Radio main interface

## 4.16 Calendar

When driver want check current date he can use both assistant mode and calendar mode. In here additionally he can check even previous dates

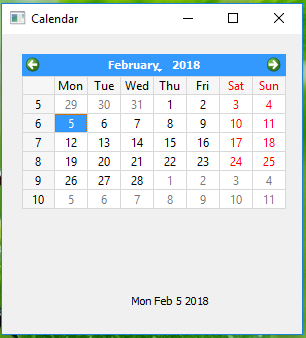
****

Figure 4.16.1: Calendar

## 4.17 Settings

This function use for change the driver’s identification. Before navigate to that part user must say correct word for promoted password. Then user can change driver’s authorization.

# 5. Implementation

System is captured live stream with the web cam. For this need to create a video capture object. Normally camera argument should be passed as 1 or 0.As in my case one camera will be connected. Primary camera is selected by passing 0. Then can capture frame by frame.

cap=cv2.VideioCapture(0) use for pass this argument and launch the camera. After getting video steam must release the capture. cap.release() method use for close capturing of frames. cap.read() returns True or False values.If frame is read correctly,it will be True. Then system is converted frame into gray scale. cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) is used for that. Converting to gray scale is not necessary for image processing.But is usually done for simplicity and data reduction.When we have a RGBA image (red-green-blue-alpha).If we converted this image to gray scale we need to process ¼ of the data compared to the color image.

import numpy as np

import cv2

cap = cv2.VideoCapture(0)

while(cap.isOpened()):

ret, frame = cap.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

cv2.imshow('frame',gray)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

# 

Now can capture a video, after capturing video cv2.imwrite() method use to save that video.Using that process we can capture a video, Then needs to analyze the video and identify the faces. The cascadeclassifier( ) is used the identify the object. The XML file is used to create the neural network. The XML file has the values of nodes and weights.

<feature>

<rects>

<\_>

8 10 9 10 -1.<\_>

<\_>

8 15 9 5 2.<\_></rects>

<tilted>0</tilted></feature>

<threshold>1.2923140311613679e-003</threshold>

<left\_val>-0.9412388205528259</left\_val>

<right\_node>1</right\_node><\_>

<\_>

This XML file create using positive and negative images. The objectmaker software of haartraining\_stuff is used to crop the images.

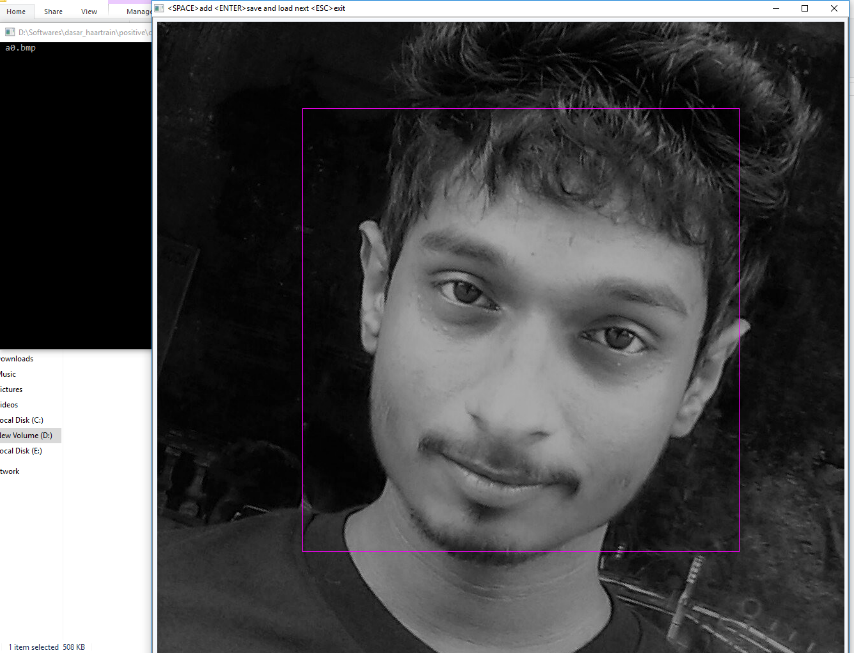


Figure No 5.0.1: Crop the faces with objectmaker software

# After crop all images of .bmp format, that file’s information load into the text file.

rawdata/face1.bmp 1 321 166 587 681

rawdata/face10.bmp 1 11 7 143 142

rawdata/face11.bmp 1 8 9 136 154

rawdata/face12.bmp 1 34 9 103 133

rawdata/face13.bmp 1 8 7 134 151

rawdata/face14.bmp 1 15 11 76 106

rawdata/face15.bmp 1 19 9 163 155

rawdata/face16.bmp 1 12 11 104 126

rawdata/face17.bmp 1 3 6 78 97

rawdata/face18.bmp 1 3 6 67 101

rawdata/face19.bmp 1 2 7 66 89

rawdata/face2.bmp 1 25 11 118 141

rawdata/face20.bmp 1 9 5 102 150

rawdata/face21.bmp 1 6 8 119 155

rawdata/face22.bmp 1 5 5 112 123

rawdata/face3.bmp 1 37 9 118 108

rawdata/face4.bmp 1 18 4 100 121

rawdata/face5.bmp 1 19 5 131 145

rawdata/face6.bmp 1 24 6 115 139

rawdata/face7.bmp 1 23 9 94 104

rawdata/face8.bmp 1 14 5 111 144

rawdata/face9.bmp 1 7 8 123 129

Face1.bmp image’s height and width represents in pixels as 587 and 681.face cropped rectangle coordinator represent as x=321 and y=166.

# Then negative images must create in .bmp file format. After adding negative file samples run create\_list.bat .This creates the negative images’ information file.

# 

UMD\_077.jpg

UMD\_078.jpg

UMD\_079.jpg

UMD\_080.jpg

UMD\_081.jpg

UMD\_082.jpg

UMD\_083.jpg

Run the ‘samples\_creation.bat’ this creates a positive vector file. These system call to the Opencv\_createsample.exe file and pass the text file which is created. The .vec file is created in the project. When these script is run.



Figure No 5.0.2: vector file for the positive images

The next process is creating XML file using this vector file. The XML creator windows batch file is used for this. These all images are processed and converts it in to numerical value. Then these values are bind together and create XML file using this vector file. Number of nodes and number of stages of the neural network is depend on these number of positive and negative images. This process time is approximately 15 hours. But this depend on power of the computer.

This procedure is used for create face cascade and eye cascade. For this process different XML file should be created.

Then this XML file can be used for the eye and face detection.

import cv2

import time

import winsound

cap = cv2.VideoCapture(0) # 640,480

w = 640h = 480

id=raw\_input('enter id')

samplenum=0;

while (cap.isOpened()):

ret, frame = cap.read()

if ret == True:

eye = cv2.CascadeClassifier('haareye.xml')

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

detected =eye.detectMultiScale(frame, 1.3, 5)

CascadeClassifier('haareye.xml' ) –

CascadeClassifier( ) is used to read the XML file. The Cascade Classifier is identify the images. The classifier outputs a “1” if the show the eye, and “0” otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier.

eye.detectMultiScale(frame, 1.3, 5)

This is use for the control the detection process. Normally 1.3 is use for the real time detection.

Rectangle ( ) –

Rectangle ( ) is used to display the border line in the selected area. Normally detectMultiScale() identified image using 4 digits. There are X and Y coordinators, width(w) and height(h). Rectangle( ) function use this values to mark the object.

cv2.rectangle(frame, (x, y), ((x + w), (y + h)), (0, 0, 255), 1)

(0,0,255) is represent BGR value. And 1 is represent thickness of the line

cv2.line(frame, (x, y), ((x + w, y + h)), (0, 0, 255 ), 1)

cv2.line(frame, (x + w, y), ((x, y + h)), (0, 0, 255), 1) cv2.circle(frame, (x+w/2, y+h/2),3, (0, 0, 255), 5)

This also use for the represent to selected area.

import numpy as np

import cv2

import time

import winsound

cap = cv2.VideoCapture(0) # 640,480

w = 640

h = 480

id=raw\_input('enter id')

samplenum=0;

while (cap.isOpened()):

ret, frame = cap.read()

if ret == True:

faces = cv2.CascadeClassifier('haarcascade\_eye.xml')

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

detected = faces.detectMultiScale(frame, 1.3, 5)

for (x, y, w, h) in detected:

samplenum = samplenum + 1;

cv2.imwrite("eyecaptured/eyeid." + str(id) + "." + str(samplenum) + ".jpg", gray[y:y + h, x:x + w])

cv2.rectangle(frame, (x, y), ((x + w), (y + h)), (0, 0, 255), 1)

cv2.line(frame, (x, y), ((x + w, y + h)), (0, 0, 255 ), 1)

cv2.line(frame, (x + w, y), ((x, y + h)), (0, 0, 255), 1)

cv2.circle(frame, (x+w/2, y+h/2),3, (0, 0, 255), 5)

pupilO = frame # show picture

cv2.waitKey(100);

cv2.imshow("eye", frame);

cv2.waitKey(1);

if (samplenum > 1000):

break

# else:

# break

# Release everything if job is finished

cap.release()

cv2.destroyAllWindows()

# When camera argument pass as 0,System is promoted user input of ID. Here ID==1 use for the identify open eye and ID==0 is use for identify close eye and microsleep.

Using ‘cv2.imwrite( )’ capture the image that is selected as gray[y:y + h, x:x + w].



Figure No 5.0.3: open eyes named as ID==1

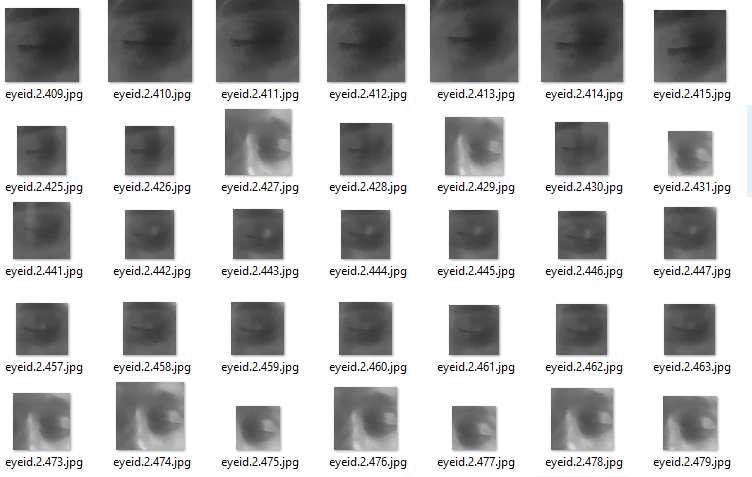


Figure No 5.0.4: close and microsleep eyes named as ID==0

import os

import cv2

import numpy as np

from PIL import Image

recognizer=cv2.createLBPHFaceRecognizer();

path='eyecaptured'

def getImagesWithID(path):

imagePaths=[os.path.join(path,f) for f in os.listdir(path)]

faces=[]

IDs=[]

for imagePath in imagePaths:

faceImg=Image.open(imagePath).convert('L');

faceNp=np.array(faceImg,'uint8')

ID=int(os.path.split(imagePath)[-1].split('.')[1])

faces.append(faceNp)

print ID

IDs.append(ID)

cv2.imshow("training",faceNp)

cv2.waitKey(10)

return IDs,faces

Ids,faces=getImagesWithID(path)

recognizer.train(faces,np.array(Ids))

recognizer.save('eyecascade/trainningData.yml')

cv2.destroyAllWindows()

# imagePaths=[os.path.join(path,f) for f in os.listdir(path)]

# This use to fetch the directory each and every images and putting into f. then join method appending f to the path with the slash.this helps to creating list of all the images.

Now we can loop all image and stored that user id and face.

faceNp=np.array(faceImg,'uint8')

Open the image and convert it into numpy array.because opencv only work with numpy array

All face recognition models in OpenCV are derived from the abstract base class [FaceRecognizer](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html#FaceRecognizer : public Algorithm), which provides a unified access to all face recongition algorithms in OpenCV.Here I use createLBPHFaceRecognizer() to train my new data set.

import numpy as np

import cv2

import time

from playsound import playsound

import pyttsx

from datetime import datetime

#import winsound

cv2.destroyAllWindows()

engine = pyttsx.init()

engine.say('!!!!anti sleep mode is activated!')

engine.runAndWait()

playsound('beep.mp3')

cap = cv2.VideoCapture(0) # 640,480

w = 640

h = 480

rec=cv2.createLBPHFaceRecognizer();

rec.load("eyecascade\\trainningData.yml")

id=0

font=cv2.cv.InitFont(cv2.cv.CV\_FONT\_HERSHEY\_COMPLEX\_SMALL,2,1,0,2)

while (cap.isOpened()):

ret, frame = cap.read()

if ret == True:

faces = cv2.CascadeClassifier('haarcascade\_eye.xml')

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

detected = faces.detectMultiScale(frame, 1.3, 5)

for (x, y, w, h) in detected:

cv2.rectangle(frame, (x, y), ((x + w), (y + h)), (0, 0, 255), 1)

cv2.line(frame, (x, y), ((x + w, y + h)), (0, 0, 255 ), 1)

cv2.line(frame, (x + w, y), ((x, y + h)), (0, 0, 255), 1)

cv2.circle(frame, (x+w/2, y+h/2),3, (0, 0, 255), 5)

id, conf = rec.predict(gray[y:y + h, x:x + w])

if (id == 2):

id = "close"

print "eye close";

playsound('sd.mp3')

elif(id==1):

id="open"

cv2.cv.PutText(cv2.cv.fromarray(frame), str(id), (x, y + h), font, 255);

# show picture

cv2.imshow('frame', frame)

# cv2.imshow('frame2', pupilFrame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# else:

# break

# Release everything if job is finished

cap.release()

cv2.destroyAllWindows()

Then created data set of .yml format is loaded into cv2.createLBPHFaceRecognizer().Now system can be identified open eye and microsleep eye.if(id==1) then it is open eye.else it is close eye.

This process also use for the identify driver’s face

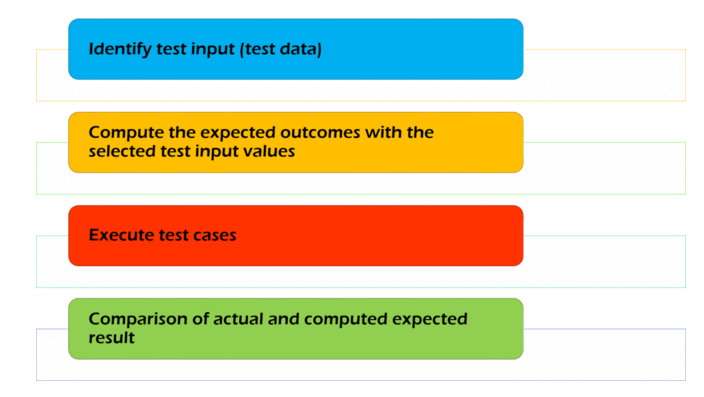
# 6. Evaluation

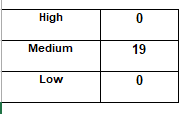
This application was tested using functional testing methods. Functional testing is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application.

Each and every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results. This testing involves checking of User Interface, APIs, Database, security, client/ server applications and functionality of the Application Under Test. This testing was done manually.

The prime objective of Functional testing is   checking the functionalities of the software system. It mainly concentrates on -

* Mainline functions:  Testing the main functions of an application
* Basic Usability: It involves basic usability testing of the system. It checks whether an user can freely navigate through the screens without any difficulties.
* Accessibility:  Checks the accessibility of the system for the user
* Error Conditions: Usage of testing techniques to check for error conditions.  It checks whether suitable error messages are displayed.

 Figure No 6.0.1: Functional Testing Process



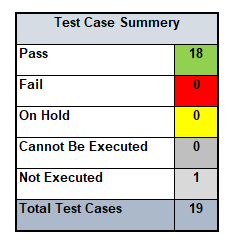
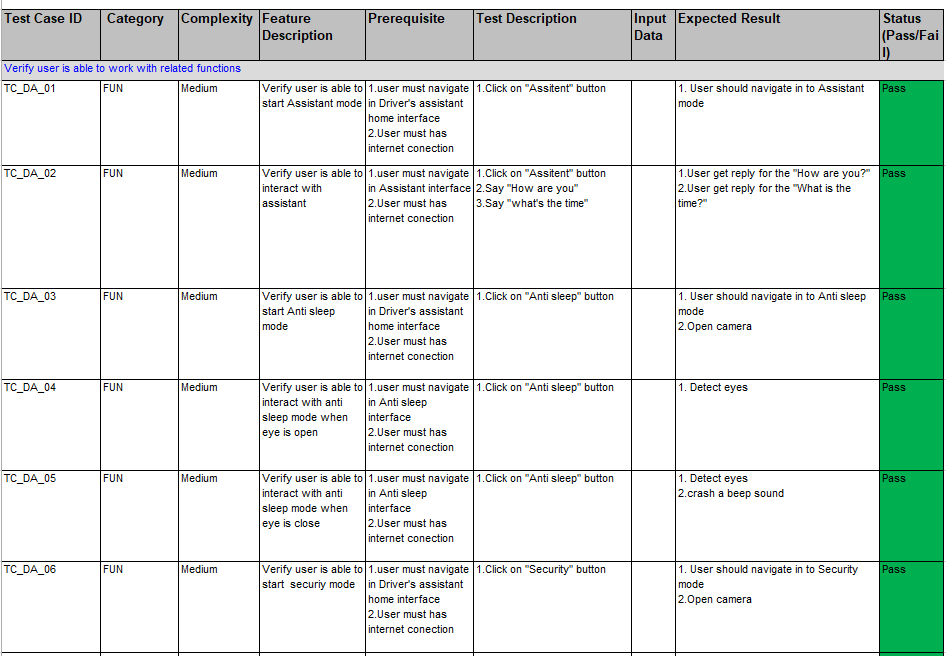
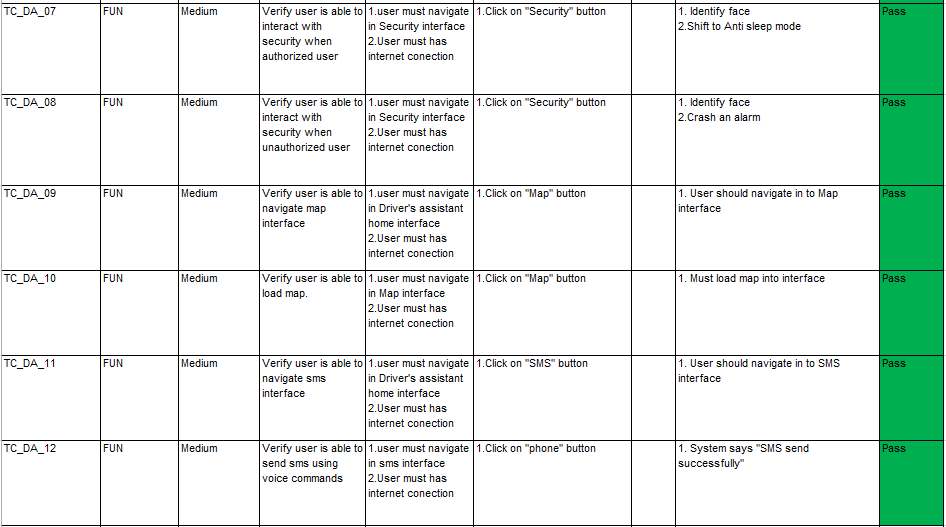
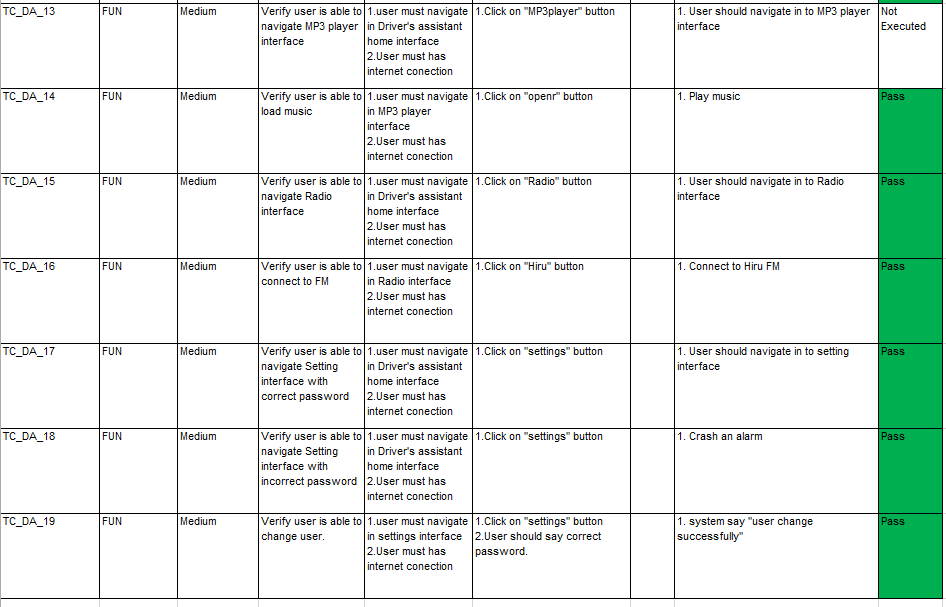


Table 6.0.2:Functional testing for Driver assistent



****

****

# 

# 7. Result and Conclusion

The design of driver drowsiness detection and alert system is presented. This system is used to avoid various road accidents caused by drowsy driving. And also this system can be used for security purpose and driver concentration. This involves avoiding accident to unconsciousness through micro-sleep.

# 8. References

[1]. Aleksandar Colic,Driver Drowsiness Detection: Systems and Solutions,

2010,P.54.

[2]. Weihua Yang ,Autonomous Robots Research Advances, 2012,P.32,33,34

[3].http://www.onlineclassnotes.com/2011/10/describe-fundamental- steps-of-     digital.html-on 14th October 2017

[4].http://www.onlineclassnotes.com/2011/10/describe-fundamental-steps-of-digital.html-on 14th October 2017

[5].https://sisu.ut.ee/imageprocessing/book/1-on 14th October 2017

[6]. [https://easstacademy.org/blog/microsleepdrowsiness- driving](https://easstacademy.org/blog/microsleepdrowsiness-%20%20driving)-on 19th October 2017

[7]. <http://www.sundaytimes.lk/081102/News/sundaytimesnews_21.html>- on 19th October 2017

[8].[http://island.lk/index.php?page\_cat=article-details&page=article-](http://island.lk/index.php?page_cat=article-details&page=article-details&code_title=30453)details&code\_title=30453 -on 19th October 2017

[9].https://en.wikipedia.org/wiki/YAML –on 5th December 2017

[10] https://stackoverflow.com/questions/130853-compared-to-xml –on 5th December 2017

[11]. Mahdi Rezaei, Creating a Cascade of Haar-Like Classifiers: Step by Step

P. 1-7

[12]. https://docs.opencv.org/3.4.1/d7/d8b/tutorial\_py\_face\_detection.html–on 31st March 2018.

[13].<https://stackoverflow.com/questions/20473352/is-conversion-to-gray-scale-a-necessary-step-in-image-preprocessing> –on 31st March 2018.