INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 05 ISSUE: 04 | APRIL - 2021 ISSN: 2582-3930

REAL-TIME DRIVER DROWSINESS DETECTION SYSTEM

Sangeetha B[1], Ashwin R[2], Rithik R Jain[3], Vijay Krishnaa S[4]

Abstract— Driver fatigue is a chief cause of traffic accidents. For this reason, it is essential to develop a monitoring system for drivers' level of fatigue. In recent years, driver fatigue monitoring technology based on machine vision has become a research hotspot, but most research focuses on driver fatigue detection during the day. A lot of road accidents nowadays occur due to reckless driving. A driver driving for long hours is prone to fatigue and drowsiness. There are also other possible situations such as, the driver could be drunk. Such abnormalities can lead to even fatalities. On road driver's fatigue and drowsiness is contributing to more than 30% of reported road accidents. Considering this fact the system designed will detect if the driver is drowsy or not based on his eye and mouth movements. This is done by visual assessment of the driver using a night vision camera. This project presents a day and night monitoring system for real-time fatigue driving detection, which makes up for the deficiencies of fatigue driving detection technology at night. First, we use infrared imaging to capture a driver's image at night, and then we design an algorithm to detect the driver's face .Second, we use eye-detection algorithm that locates the position of the corners of the eye and calculate eye aspect ratio as well as mouth aspect ratio to detect if the person is yawning. We use eye blinking parameters to evaluate fatigue. It involves live monitoring of EAR (Eye aspect Ratio) and MAR (Mouth Aspect Ratio – detects yawn) by application of Image processing. HD live video is disintegrated in continuous frames and facial landmarks have been detected using pre trained Neural Network based Dlib functions. Hence a computer vision based idea has been utilized for road safety. A camera system installed towards the face of the driver, checks the driver's eyes and mouth with a specific end goal to recognize fatigue. A warning sign is issued to caution the driver, in such a situation when fatigue is recognized.

I. INTRODUCTION

1.1 IMAGE PROCESSING

Digital Image Processing means processing digital images by means of a digital computer. We can also say that it is a use of computer algorithms, in order to get enhanced images either to extract some useful information.

An image is defined as a two-dimensional function, F(x,y), where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x,y) is called the intensity of that image at that point. When x,y, and amplitude values of F are finite, we call it a digital image.

In other words, an image can be defined by a two-dimensional array specifically arranged in rows and columns.

Digital Image is composed of a finite number of elements, each of which elements have a particular value at a particular location. These elements are referred to as picture elements, image elements, and pixels. A Pixel is most widely used to denote the elements of a Digital Image.

In particular, digital image processing is the only practical technology for:

- Classification
- Feature extraction
- Multi-scale signal analysis
- Pattern recognition
- Projection

1.2 TOOLS AND IMAGE PROCESSING LIBRARIES

Following optimized tools and image processing libraries are used by the author for implementation of the presented algorithm.

Open CV: OpenCV (Open-source Computer Vision) is the Swiss Army knife of computer vision. It has a wide range of modules that can help us with a lot of computer vision problems. But perhaps the most useful part of OpenCV is its architecture and memory management. It provides you with a framework in which you can work with images and video in any way you want, using OpenCV's algorithms or your own, without worrying about allocating and deallocating memory for your images. Open CV libraries and functions are highly optimized and can be used for real time image and video processing. OPENCV's highly optimized image processing functions are used by the author for real time image processing of live video feed from camera. Some face recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features.

DLib: Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. Open Source Dib library is used by the author for implementation of CNN(Neural Networks). Highly optimized Pre-learned facial shape predictor and detectors functions are used by the author for detection of facial landmarks. Facial landmarks were further used for extracting eye coordinates.

COMPILING/INSTALLING THE DLIB PYTHON INTERFACE

- You can install dlib using the command:
    - ➔ pip install dlib

Python: Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language including NASA, Google, YouTube, BitTorrent, etc. Python is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had a deep focus on code readability. Python language is used by the author due to his cross platform compatibility as the main coding language for algorithms. Open CV and Dlib libraries are integrated in python interpreter for using readymade optimized functions.

pyttsx 3 2.7: An OFFLINE Python Text to Speech library (TTS) which works for both python3 and python2. This library is very useful especially if you don't want any delay in the speech produced and don't want to depend only on the internet for TTS conversion. It also supports multiple TTS engines like Sapi5 , nsss , espeak .

imutils : A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.

Playsound: Play sound on Python plays an audio file of a given format. There are several modules that can play a sound file (.wav). These solutions are cross platform (Windows, Mac, Linux). The main difference is in the ease of use and supported file formats. All of them should work with Python 3. The audio file should be in the same directory as your python program, unless you specify a path.

Anaconda framework: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Anaconda command prompt is just like any command prompt, but it makes sure that you are able to use anaconda and conda commands from the prompt, without having to change directories or your path.

1.3. FACIAL LANDMARKS WITH DLIB, OPENCV, AND PYTHON

Facial landmarks are used to localize and represent salient regions of the face, such as:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more.

What are facial landmarks?

Detecting facial landmarks is a subset of the shape prediction problem. Given an input image, a shape predictor attempts to localize key points of interest along the shape.

In the context of facial landmarks, our goal is to detect important facial structures on the face using shape prediction methods. Detecting facial landmarks is therefore a two-step process:

Step 1: Localize the face in the image.

Step 2: Detect the key facial structures on the face ROI.

Step #1 (face detection) :

- We could use OpenCV's built-in Haar cascades.
- We might apply a pre-trained HOG + Linear SVM object detector specifically for the task of face detection.
- Or we might even use deep learning-based algorithms for face localization.

In either case, the actual algorithm used to detect the face in the image doesn't matter. Instead, what's important is that through some method we obtain the face bounding box (i.e., the (x, y)-coordinates of the face in the image)

Step #2: detecting key facial structures in the face region.

There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

The facial landmark detector included in the dlib library is an implementation of the One Millisecond Face Alignment with an Ensemble of Regression Trees paper by Kazemi and Sullivan (2014).

This method starts by using:

A training set of labeled facial landmarks on an image. These images are manually labeled, specifying specific (x, y)-coordinates of regions surrounding each facial structure.

Priors, or more specifically, the probability of distance between pairs of input pixels.

Given this training data, an ensemble of regression trees are trained to estimate the facial landmark positions directly from the pixel intensities themselves (i.e., no “feature extraction” is taking place). The face detector we use is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and sliding window detection scheme.

The pose estimator was created by using dlib's implementation of the paper: One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR 2014 and was trained on the iBUG 300-W face landmark dataset which is available as shape_predictor_68_face_landmarks.dat that shows 68 landmarks on face.

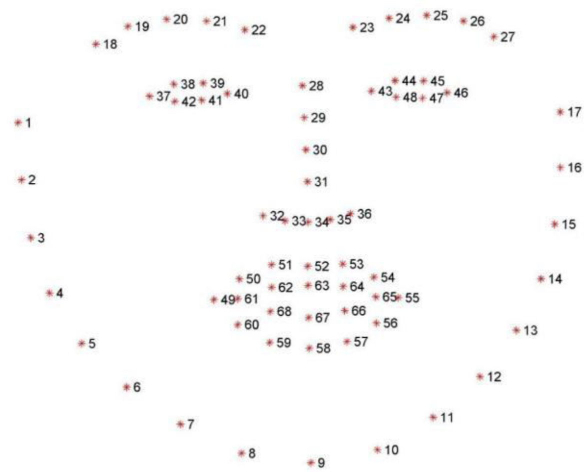
The end result is a facial landmark detector that can be used to detect facial landmarks in real-time with high quality predictions.

1.4. UNDERSTANDING DLIB’S FACIAL LANDMARK DETECTION

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face. These annotations are part of the 68 point iBUG 300-W dataset which the dlib facial landmark predictor was trained on. It’s important to note that other flavors of facial landmark detectors do exist. Regardless of which dataset is used, the same dlib framework can be leveraged to train a shape predictor on the input training data — this is useful to train facial landmark detectors or custom shape predictors.

1.5. PROBLEM STATEMENT

Drowsy driving is becoming one of the most important causes of road accidents. According to many surveys around 30% of road accidents is due to the driver fatigue and the percentage is increasing every year. Drowsiness can be due to



the adverse driving conditions, heavy traffic, workloads, late night long drive etc. Lack of sleep, absence of rest, taking medicines are also causes for drowsiness. When the driver drives for more than the normal period fatigue is caused and the driver may feel tired which will cause the driver to sleepy condition and loss of consciousness. This results in road accidents and death of drivers or serious injuries and also claims thousands of lives every year.

FIG 1.5.1 THE INDEXES OF THE 68 COORDINATES CAN BE VISUALIZED

Drowsiness is a phenomenon which is the transition period from the awake state to the sleepy state and causes a decrease in alerts and conscious levels of the driver. It is difficult to measure the drowsiness level directly but there are many indirect methods to detect the driver fatigue. Driver drowsiness detection can be measured using physiological measures, vehicle based measures, behavioural measures. Physiological measures include the measure of brain wave, heart rate, pulse rate, and using the physiological signals like ECG (Electrocardiogram), EOG (Electrooculogram), EEG (Electroencephalogram) etc. Though this method measures the drowsiness accurately but it requires a physical connection with the driver such as placing several electrodes on head, chest and face which is not a convenient method and

also discomfort for the driver in driving condition. Vehicle measures include deviations from lane position, pressure on acceleration pedals, movement of the steering wheels, etc. These are constantly monitored and any change in these which crosses a threshold indicates a probability that the driver is drowsy. Behavioural measures monitor the behaviour of the driver, which includes the yawning, eye closure, eye blinking etc. These are monitored through a camera and these drowsiness symptoms are detected. Behavioural state detection system helps to detect the drowsy driving condition early and avoid accidents. In this paper real time drowsy detection is used which is one of the best possible methods to detect driver fatigue early. Real time driver detection system using image processing captures driver eyes state non- intrusively using a camera.

Proposed algorithm is based on live monitoring of EAR (Eye aspect Ratio) AND (MAR) mouth aspect ratio by application of Image processing. HD live video is decomposed in continuous frames and facial landmarks have been detected using pre-trained Neural Network based Dlib functions. Dlib functions are trained using the HAAR Cascade algorithm. Intel's Open source Image processing libraries (OPEN CV) is used as primary Image processing tool. Python Language is used as the main coding language. EAR is calculated by calculating Euclidean distance between measured eye coordinates and MAR is calculated by calculating Euclidean distance between measured mouth coordinates. Blink and yawn detection mechanism is implemented by monitoring EAR and MAR against a threshold value. Blinks and drowsiness level are displayed on the monitor screen and warning is given when sleep is detected .

FIG 1.5.2 GRAPH GOOGLE TRENDS SHOWING NUMBER OF TIMES THIS PROBLEM WAS SEARCHED IN INDIA OVER A YEAR

Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term.

Hence it is seen that this problem statement had the highest popularity in the beginning of the year 2020. Considering this problem's popularity, we decided to come up with a good solution for the above problem.

II. LITERATURE SURVEY

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.[1] is an optimization based model for crowd density prediction, using pyramidal convolutional recurrent network architecture and the result obtained was that PCRN consistently outperforms the state-of the-art methods for crowd density prediction across two taxi datasets from Beijing and Singapore.[2] designed, trained and benchmarked a data-driven procedure to forecast crowd movements, which can in real-time predict crowd movement using GPS traces and training a Recursive Neural Network (RNN) with a Gated Recurrent Unit (GRU) and six benchmark models to forecast the next location of pedestrians and they arrived at a conclusion that using cell sequence data and a RNN-GRU crowd movements at large-scale events can be predicted without first map-matching the GPS trajectories. [3] uses a deep-learning-based approach, called ST-ResNet, to collectively forecast the inflow and outflow of crowds in each and every region of a city. This model is evaluated on two types of crowd flows in Beijing and NYC, achieving significant performances.[4] presents a solution to ensure effective management and security in large-scale public events in terms of WiFi-based crowd counting and long short-term memory (LSTM) neural network-based forecasting and they concluded that among different LSTM models, Convolutional LSTM delivered the best performance.[5] is a deep convolution neural network (DCNN) based system which can be used for near real-time crowd counting. The system uses NVIDIA GPU processor to exploit the parallel computing framework to achieve swift and agile processing of the video feed taken through a camera. The model is trained extensively by providing several scenarios such as overlapping heads, partial visibility of heads etc. This system provides significant accuracy in estimating the head count in dense population in reasonably less amount of time. The proposed system performs admirably in situations where manual counting is simply not possible. Deep learning also enables the system to perform in versatile environments and continuously learn from new inputs.[6] tells the overview and performance comparison of crowd counting techniques using convolutional neural networks (CNN) based on density map estimation. It is a comprehensive analysis and benchmarking of crowd counting based on the UCF-QNRF dataset that contains the largest number of crowd count images and head annotations available in the public domain. This article studied the network models for crowd counting released in recent years and summarized the network architectures of the different models, the method of learning objectives and the loss function. Since the BL model showed a good performance in the evaluated networks, the architecture of the BL model was replaced to verify the robustness of the BL model data processing method and loss function. [7] uses combination of deep and shallow, fully convolutional networks to predict the density map for a given crowd image. Such a combination is used for effectively capturing both the high-level semantic information (face/body detectors) and the low-level features (blob detectors), that are necessary for crowd counting under large scale variations. This method outperforms the state-of-the-art methods on the challengings UCF CC 50 dataset. [8] The proposed MCNN allows the input image to be of arbitrary size or resolution. By utilizing filters with receptive fields of different sizes, the features learned by each column CNN are adaptive to variations in people/head size due to perspective effect or image resolution. This model outperforms the state-of-art crowd counting methods on all datasets used for evaluation. Further, this model trained on a source domain can be easily transferred to a target domain by fine-tuning only the last few layers of the trained model, which demonstrates good generalizability of the proposed model.[9] is a deep convolutional neural network (CNN) for crowd counting, and it is trained alternatively with two related learning objectives, crowd density and crowd count. This proposed switchable learning approach is able to obtain better local optimum for both objectives. The learned deep model specifically has better capability for describing crowd scenes than other hand-craft features. [10] proposed a global-residual two-stream recurrent network, which lever-ages the consecutive crowd video frames as inputs and their corresponding density maps as auxiliary information to predict the future crowd distribution and demonstrated that the framework is able to predict the crowd distribution in different crowd scenarios and it delve into many crowd analysis applications. In [11] switching convolutional neural networks (S-CNN) is used for increasing the accuracy of crowd detection and counting. In earlier approaches the inter scene variation is not considered but while using S-CNN the inter scene variation and semantic analysis is considered to improve the estimation of the count. The advantages of switching convolutional neural network is that it leverages intra-image crowd density variation to improve the accuracy of localization of the people. The proposed algorithm is implemented using Shanghai Tech part A dataset and observed that it produces the Mean Average Error value is 98.87. In [12] a model for crowd counting in public places of high and low densities is proposed. The model works under various scene conditions and with no prior knowledge. A Deep CNN model (DCNN) is built based on convolutional neural network (CNN) structure with small kernel size and two fronts. This paper, provides accurate people counting model from an arbitrary single image, with any random crowd density and random camera perspective. The results indicate that our proposed model achieves a lower MAE combed to state-of-art methods. [13] is implemented using a deep neural-network-based approach , an end-to-end predictive model is configured to take in tweets as additional inputs to forecast the future flow of crowds in an urban environment. It extract various features from tweets, such as tweet counts, tweet tenses and sentiments as additional signals to the predictive model. This paper explored the effectiveness of using tweets to crowd flow prediction by extending upon an existing state-of-the-art crowd flow prediction model known as ST-ResNet, adding various linguistic features from realtime tweets. Through the empirical experiments with two different datasets used to represent traffic flows in Singapore, it was found that tweets are indeed useful to improving the prediction accuracy up to 3.28% on average, and tested to be statistically significant. [14] is a study on Crowd Detection and Density Analysis for Safety Control. It tells that pattern recognition technique helps to estimate the crowd detection count and density by using face and detection. The counting performance has been steadily improved because of Deep Convolutional Neural Network.This study concludes that the deep learning model is very efficient for crowd counting and analysis where we discussed on some methods of Convolutional Neural Network which is our basic framework to learn efficient features for counting. It is an end-to-end training method which performs a whole image based inference. To get better performance of crowd counting, it requires large labelled dataset. [15] proposes a variant of a deep learning model called convolutional LSTM (ConvLSTM) for crowd counting. Unlike the previous CNN-based methods, this method fully captures both spatial and temporal dependencies. Furthermore, this paper extend the ConvLSTM model to a bidirectional ConvLSTM model which can access long range information in both directions. This model outperforms existing crowd counting methods on the UCF CC 50 dataset, Mall dataset, and WorldExpo dataset, and achieve comparable results on the UCSD dataset.[16] talks about a system that consists of two main parts, the first one is a server side application connected to IP cameras to detect crowd level in certain location(s) while the other one is a mobile application with different users rights to receive alarm from the server side application. This framework provides an effective method to connect and alert all of the system users immediately, preventing high crowd level danger. This framework gives an early alert just seconds after the level exceeds the defined limits resulting in a good chance to solve problem with minimum loses or damages and preventing high crowd level danger. The system was also tested using Interface, unit and usability test to guarantee that the system working probably. The users reacted with it in an efficient way. The tests results showed a promising result.

III. REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

- Raspberry pi3
- IR raspberry pi Camera
- 32 GB SD card with Raspbian installed
- Computer with 8GB RAM
- VGA To HDMI cable for full PC setup
- Connecting wires
- Ribbon flat cable for raspberry pi camera

2.2 SOFTWARE REQUIREMENTS

- Anaconda prompt
- Python 3.6 and above

- Text editor (SUBLIME)

IV. SYSTEM DESIGN

The proposed real-time drowsy driver detection system consists of the hardware and the software modules which aid to identify and alert the user. The proposed method of drowsiness detection is devised on a hardware and is employed based on the information it receives from its end points. The user will actively handle the response provided by hardware. Also, an automatic warning is done for the user.

This proposed system makes use of Raspberry Pi. The Raspberry Pi is a series of small single-board computers that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. The Raspberry Pi is integrated with a webcam to get the real-time data. This real-time data that has been acquired by the hardware is then subject to python programming which uses its packages to compute the required information.

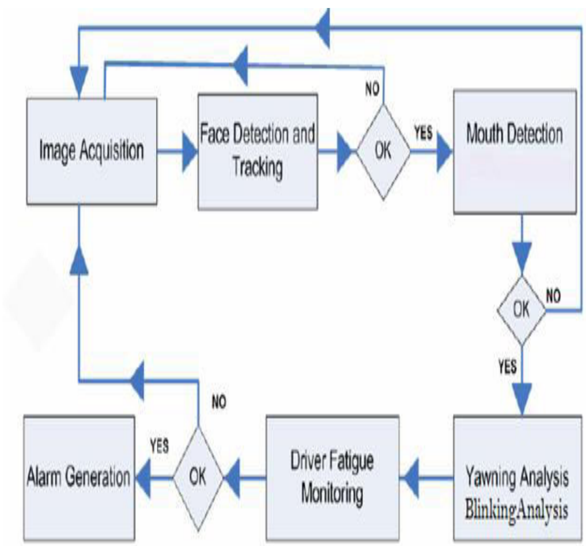


Fig 4.1 BLOCK DIAGRAM OF REAL-TIME DRIVER DROWSINESS DETECTION SYSTEM.

Shape predictor is used to analyse the data and identify the face as well as the facial features. Once the features have been identified, EAR is used to determine the user's current state. If the calculated EAR is below or above a particular threshold value, the user is then notified using an alarm. This threshold value is selected by repeated testing. OpenCV is used to project the incoming data onto a screen for visual purposes. The system will keep running unless and until stopped by the user.

V. IMPLEMENTATION

5.1 SYSTEM DESIGN

Algorithms can be divided broadly in following sub modules.

- Frame Acquisition
- Facial landmark detection
- Eye Localization and tracking
- Mouth localization
- Extracting eye and mouth geometrical coordinates
- Measuring EAR and MAR
- Monitoring of EAR for blinks detection and MAR for yawn detection
- Estimation of microsleep periods between blinking
- Audio Visual warning on microsleep detection.

5.2 EAR (eye aspect ratio)

Proposed algorithm is based on a computer vision method. The main focus is on the detection of blinks by estimating the EAR(Eye aspect Ratio). This is achieved by monitoring the eyes of the driver throughout the entire video sequence.
Micro sleep: A microsleep is a temporary episode of sleep or drowsiness which may last for a fraction of a second or up to 30 seconds where an individual fails to respond to some arbitrary sensory input and becomes unconscious.

Each eye is represented by 6 (x, y)-coordinates in landmarks returned Dlib predictor function , starting at the left-corner of the eye and then working clockwise around the remainder of the region. There is a relation between the width and the height of these coordinates.

$$\textbf{EAR (Eye aspect Ratio)} = \frac{|p2-p6|+|p3-p5|}{2|p1-p4|}$$

EAR is define as per below formula , where p1,p2….p6 is the eye coordinates

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only one set of horizontal points but two sets of vertical points. The eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place. When the person blinks the eye aspect ratio decreases dramatically, approaching zero.

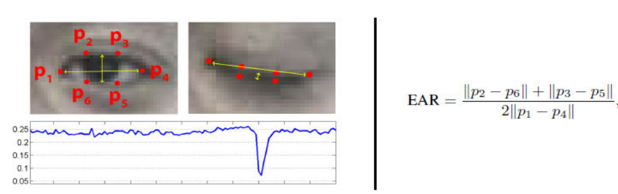


FIG 5.2.1 EAR calculation and formula

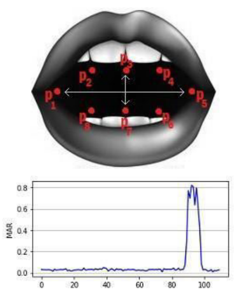
The threshold for EAR is 0.25

Python Function for calculating EAR

```
def eye_aspect_ratio(self, eye):
 A = dist.euclidean(eye[1], eye[5])
 B = dist.euclidean(eye[2], eye[4])
 C = dist.euclidean(eye[0], eye[3])
 ear = (A + B) / (2.0 * C)
 return ear
```

5.3. MAR (mouth aspect ratio)

It is similar to the calculation of eye aspect ratio. The threshold for MAR is 0.75



$$\mathrm{MAR} = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2\,\|p_1 - p_5\|}$$

FIG 5.3.1 MAR calculation and formula

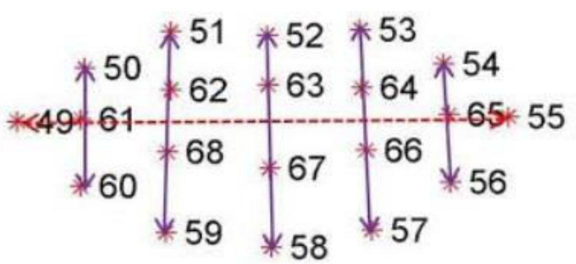


FIG 5.3.2 Indexes indicating mouth

5.4. RASPBERRY PI 3:

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) or cases. Using raspberry pi 3 and IR camera it is possible to continuously monitor a driver day and night.

Features of this camera:

- This camera can also work at night.
- Resolution: 5MP
- Interface Type: CSI(Camera Serial Interface)
- Supported Video Formats: 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 video
- Fully Compatible with Raspberry Pi 3 Model B.
- Plug-n-Play camera for Raspberry Pi 3 Model B.

Output achieved working with raspberry pi 3:

- Raspberry pi 3 with processing speed of 1GB RAM was used to visually assess a driver's state but the response time turned out to be slow due to the deployment of opencv and dlib.
- The Raspberry Pi isn't quite fast enough for real-time facial landmark detection.

FIG 5.4.1 IR Camera

- However Raspberry Pi can be optimized along with the dlib compile to enable real-time facial landmark detection.

5.5 OUTPUT

The python code for driver drowsiness was executed in a laptop computer with Ram of about 8 GB speed and in Python 3.6 and above environment. The response from the system was immediate with good results. The input is fed using a webcam and the output is displayed on the screen. The alarm file (.mp3 or .wav) is played using the playsound package of python. The alert message is given using python's text to speech convertor package.

VI. RESULTS

DROWSINESS DETECTION

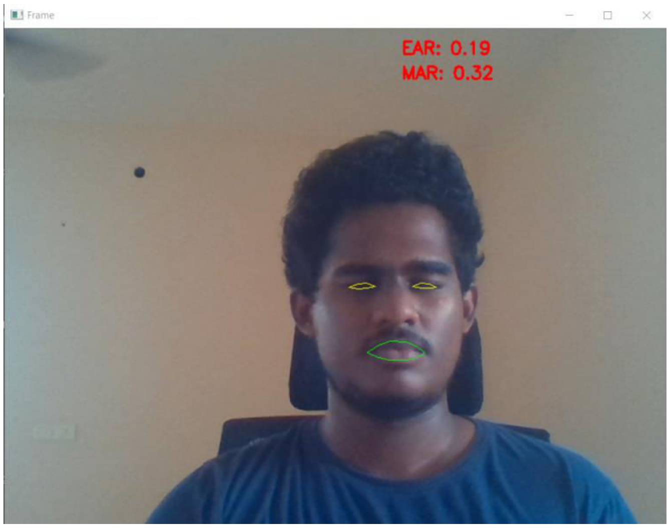


FIG 6.1 Detection for eyeclosed picture

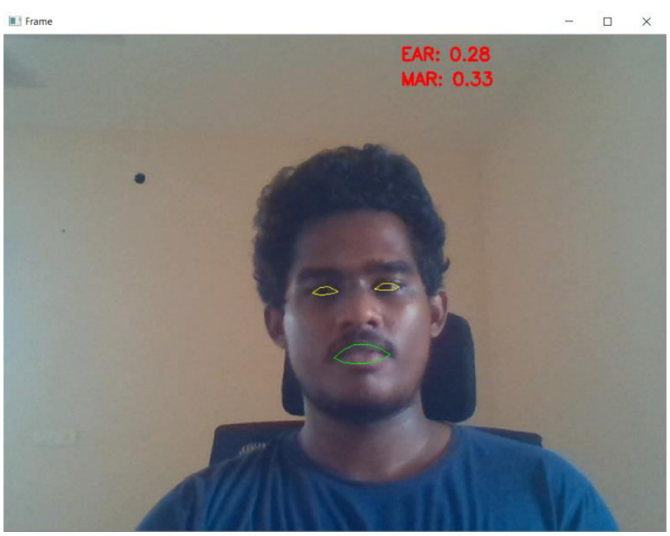


FIG 6.2 Example for detection during normal circumstance

HARDWARE

FIG 6.3. IR camera for raspberry pi 3

VII. CONCLUSION

In this paper, we presented the conception and implementation of a system for detecting driver drowsiness based on vision that aims to warn the driver if he is in a drowsy state. This system is able to determine the driver's state using a camera and Raspberry Pi. Face and eyes detection are implemented based on symmetry. The processor and camera are integrated and working successfully. All observations and experimental set up proves that the proposed system is a proper solution to detect the drowsiness of a driver.

Application of this project to society:

Driver drowsiness has become a major cause of traffic accidents. As a matter of fact, studies show that around one quarter of all serious motorway accidents are attributable to sleepy drivers.

The purpose of this system is to reliably quantify commercial motor vehicle driver drowsiness and provide a real-time warning to the driver. It is capable of not only alerting the driver, but also preventing such accidents from occurring by warning the user and keeping them awake before they fall asleep. The drowsiness detection system can be used for different applications. One of them is heavy vehicles for example trucks, since the drivers of trucks have long driving periods. It can also be used for commercial vehicles. Many people use public transport facilities for travelling. For their safety this system can be used in public vehicles. Heavy things are lifted by using cranes and transporting them to other places. So for overloaded cranes and mobile cranes this system can be used to avoid accidents related to drowsiness. In such a manner, this system ensures the safety of the users and protects them from any fatal accidents.

REFERENCES

[1] ZONOOZI, A., KIM, J., LI, X. AND CONG, G., 2018. PERIODIC-CRN: A CONVOLUTIONAL RECURRENT MODEL FOR CROWD DENSITY PREDICTION WITH RECURRING PERIODIC PATTERNS. [ONLINE] IJCAI.ORG. AVAILABLEAT: <HTTPS://WWW.IJCAI.ORG/ PROCEEDINGS/2018/0519.PDF> [ACCESSED 15 FEBRUARY 2021].

[2] DUIVES, D., WANG, G. AND KIM, J., 2019. *FORECASTING PEDESTRIAN MOVEMENTS USING RECURRENT NEURAL NETWORKS: AN APPLICATION OF CROWD MONITORING DATA*. [ONLINE] NCBI.NLM.NIH.GOV. AVAILABLE AT:<HTTPS://WWW.NCBI.NLM.NIH.GOV/PMC/ARTICLES/PMC6358800/> [ACCESSED 15 FEBRUARY 2021].

[3] ZHANG, J., ZHENG, Y. AND QI, D., 2017. DEEP SPATIO-TEMPORAL RESIDUAL NETWORKS FOR CITYWIDE CROWD FLOWS PREDICTION. [ONLINE] MICROSOFT RESEARCH. AVAILABLE AT: <HTTPS: //WWW.MICROSOFT.COM/EN-US /RESEARCH/PUBLICATION/DEEP-SPATIO-TEMPORAL-RESIDUAL-NETWORKS-FOR-CITYWIDE-CROWD-FLOWS-PREDICTION/> [ACCESSED 15 FEBRUARY 2021]

[4] SINGH, U., DETERME, J., HORLIN, F. AND DONCKER, P., 2020. CROWD FORECASTING BASED ON WIFI SENSORS AND LSTM NEURAL NETWORKS. [ONLINE] IEEEXPLORE.IEEE.ORG. AVAILABLE AT: <HTTPS://IEEEXPLORE.IEEE.ORG/DOCUMENT/8970246/> [ACCESSED 15 FEBRUARY 2021].

[5] BHANGALE, U., PATIL, S., VISHWANATH, V., THAKKER, P., BANSODE, A. AND NAVANDHAR, D., 2020. NEAR REAL-TIME CROWD COUNTING USING DEEP LEARNING APPROACH. [ONLINE] PDF.SCIENCEDIRECTASSETS.COM. AVAILABLE AT: <HTTPS://PDF.SCIENCEDIRECTASSETS.COM/280203/1-S2.0-S1877050920X00093/1-S2.0-S187705092031053X/MAIN.PDF?X-AMZ-SECURITY-TOKEN=IQOJB3JPZ2LUX2VJED8ACXVZLWVHC3QTMSJHMEUCIQCCVVZHNFR0NF7ZNVIF32RF7%2FUS7K6NNOGQKOMZEEAVMGIGWNRJRKDFEYRAXT6SC6ADFWNQP8%2BVKTSV1R71FRQYOEYQTAMISBADGGWWNTKWMDM1NDY4NJUIDJSAPKK8N

[6] ALOTAIBI, R., ALZAHRANI, B., WANG, R., ALAFIF, T., BARNAW, A. AND HU, L., 2020. PERFORMANCE COMPARISON AND ANALYSIS FOR LARGE-SCALE CROWD COUNTING BASED ON CONVOLUTIONAL NEURAL NETWORKS. [ONLINE] IEEEXPLORE.IEEE.ORG. AVAILABLE AT: <HTTPS://IEEEXPLORE.IEEE.ORG/STAMP/STAMP.JSP?ARNUMBER=9256256> [ACCESSED 16 FEBRUARY 2021].

[7] BOOMINATHAN, L., KRUTHIVENTI, S. AND BABU, R., 2016. *CROWDNET: A DEEP CONVOLUTIONAL NETWORK FOR DENSE CROWD COUNTING*. [ONLINE] ARXIV.ORG. AVAILABLE AT: <HTTPS://ARXIV.ORG/PDF/1608.06197.PDF> [ACCESSED 16 FEBRUARY 2021].

[8] BHANGALE, U., PATIL, S., VISHWANATH, V., THAKKER, P., BANSODE, A. AND NAVANDHAR, D., 2020. NEAR REAL-TIME CROWD COUNTING USING DEEP LEARNING APPROACH. [ONLINE] PDF.SCIENCEDIRECTASSETS.COM. AVAILABLE AT: <HTTPS://PDF.SCIENCEDIRECTASSETS.COM/280203/1-S2.0-S1877050920X00093/1-S2.0-S187705092031053X/MAIN.PDF?X-AMZSECURITYTOKEN=IQOJB3JPZ2LUX2VJED8ACXVZLWVHC3QTMSJHMEUCIQCCVVZHNFR0NF7ZNVIF32RF7%2FUS7K6NNOGQKOMZEEAVMGIGWNRJRKDFEYRAXT6SC6ADFWNQP8%2BVKTSV1R71FRQYOEYQTAMISBADGGWWNTKWMDM1NDY4NJUIDJSAPKK8N

[9] ZHANG, C., LI, H. AND WANG, X., 2015. CROSS-SCENE CROWD COUNTING VIA DEEP CONVOLUTIONAL NEURAL NETWORKS. [ONLINE] OPENACCESS.THECVF.COM. AVAILABLE AT: <HTTPS://OPENACCESS.THECVF.COM/CONTENT_CVPR_2015/PAPERS/ZHANG_CROSSSCENE_CROWD_COUNTING_2015_CVPR_PAPER.PDF> [ACCESSED 17 FEBRUARY 2021].

[10] NIU, Y., SHI, W., LIU, W., HE, S., PAN, J. AND CHAN, A., 2020. OVER-CROWDEDNESS ALERT! FORECASTING THE FUTURE CROWD DISTRIBUTION. [ONLINE] RESEARCHGATE.NET. AVAILABLE AT: <HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/342093820_OVERCROWDEDNESS_ALERT_FORECASTING_THE_FUTURE_CROWD_DISTRIBUTION> [ACCESSED 17 FEBRUARY 2021].

[11] R., P., A, A. AND E, B., 2019. REAL TIME CROWD DETECTION AND COUNTING USING SWITCHING CONVOLUTIONAL NEURAL NETWORKS. [ONLINE]

ERPUBLICATION.ORG. AVAILABLE AT: <HTTPS://WWW.ERPUBLICATION.ORG/IJMCTR/PUBLISHED_PAPER/IJMCTR270.PDF> [ACCESSED 17 FEBRUARY 2021].

[12] JARRAYA, S., ALOTIBI, M. AND ALI, M., 2020. A DEEP-CNN CROWD COUNTING MODEL FOR ENFORCING SOCIAL DISTANCING DURING COVID19 PANDEMIC: APPLICATION TO SAUDI ARABIA'S PUBLIC PLACES. [ONLINE] HTTPS://WWW.RESEARCHGATE.NET/. AVAILABLE AT: <HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/348128022_A_DEEP-CNN_CROWD_COUNTING_MODEL_FOR_ENFORCING_SOCIAL_DISTANCING_DURING_COVID19_PANDEMIC_APPLICATION_TO_SAUDI_ARABIA%27S_PUBLIC_PLACES> [ACCESSED 17 FEBRUARY 2021].

[13] GOH, G. AND KOH, J., 2021. TWITTER-INFORMED CROWD FLOW PREDICTION. [ONLINE] KOHJINGYU.COM. AVAILABLE AT: <HTTP://KOHJINGYU.COM/PAPERS/TWITTER_INFORMED_ICDM2018.PDF> [ACCESSED 25 FEBRUARY 2021].

[14] CHAUDHARI, M. AND GHOTKAR, A., 2021. A STUDY ON CROWD DETECTION AND DENSITY ANALYSIS FOR SAFETY CONTROL. [ONLINE] RESEARCHGATE.NET. AVAILABLE AT: <HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/325814838_A_STUDY_ON_CROWD_DETECTION_AND_DENSITY_ANALYSIS_FOR_SAFETY_CONTROL> [ACCESSED 25 FEBRUARY 2021].

[15] XIONG, F., SHI, X. AND YEUNG, D., N.D. SPATIOTEMPORAL MODELING FOR CROWD COUNTING IN VIDEOS. [ONLINE] OPENACCESS.THECVF.COM. AVAILABLE AT: <HTTPS://OPENACCESS.THECVF.COM/CONTENT_ICCV_2017/PAPERS/XIONG_SPATIOTEMPORAL_MODELING_FOR_ICCV_2017_PAPER.PDF> [ACCESSED 25 FEBRUARY 2021].

[16] SHALASH, W., HAZIMI, A. AND ZAHRANI, B., 2017. *A MOBILE BASED CROWD MANAGEMENT SYSTEM*. [ONLINE] RESEARCHGATE.NET. AVAILABLE AT: <HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/318058287_A_MOBILE_BASED_CROWD_MANAGEMENT_SYSTEM> [ACCESSED 25 FEBRUARY 2021].