IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Driver Drowsiness Detection System

[1]Aniket Gupta, [2]Shreya Ingale, [3]Saloni Goyal,[4]Afsha Shaikh, [5]Suchitra Deokate,
[1]Student ,[2]Student,[3]Student,[4]Student, [5]Professor
[1]Computer Engineering,
[1]Savitibai Phule Pune University, Pune, India

Abstract: One of the most frequent causes of driving accidents around the world is driver drowsiness or fatigue. As a result, many people suffer fatal injuries and severe enough injuries that they can never even stand unassisted again. Their entire lives dependent on others. The primary focus of the project is image processing, and by simulating it on the Raspberry Pi Microprocessor, we will build a Driver Drowsiness Detection System. Every type of vehicle in the world can be equipped with the system, which will alert the driver or passengers if it notices signs of sleepiness in the driver's eyes. By calculating the amount of time, the system can determine whether the eyes are sleepy.

Index Terms - Drowsiness, Fatigue, unassisted, simulating

INTRODUCTION

One of the main factors contributing to motor vehicle accidents is driver fatigue. It is widely acknowledged that driver drowsiness contributes significantly to the rising number of accidents on today's roads. This Real Time Detection System project will be a very helpful change in enhancing the safety of the driver of the vehicle and will lower the percentage of accidents brought on by driver fatigue and sleepiness. More and more professions today require long-term focus. People who work in the transportation industry must pay close attention to the road so they can respond quickly to unexpected events. Long hours behind the wheel make the driver tired, which slows down their reaction time. The development of a drowsiness detection system simulation is the goal of this project.

The emphasis will be on creating a system that can precisely track whether the driver's eyes and mouth are open or closed. It is thought that the eyes can be monitored that it is possible to identify the signs of a sleepy driver early enough to prevent a car accident. A significant challenge in the development of accident-avoidance systems is the detection or prevention of driver drowsiness. In addition to the ability to detect drowsiness or fatigue in the driver's eyes, this system will also include other safety features like lane detection technology and brake synchronization. Due to lack of sleep, long periods of non-stop driving, or any other medical condition, such as brain disorders, etc., the attention level of the driver deteriorates. According to several studies on traffic accidents, driver fatigue is a factor in about 30% of collisions. When a driver drives for a longer period of time than is typical for a human, this causes excessive fatigue and also results in tiredness, which causes the driver to become sleepy or lose consciousness. Drowsiness is a complicated phenomenon that causes the driver to become less alert and conscious. Detecting drowsiness cannot be done directly, but there are a number of indirect methods that can be employed. In order to decrease the number of traffic accidents caused by drivers.

BACKGROUND

2.1 Drowsiness

Drowsiness is a serious issue that needs to be addressed for improvement in the safety of road driving. Drowsiness is defined as a decreased level of awareness portrayed by sleepiness and trouble in staying alarm but the person awakes with simple excitement by stimuli. Past statistical data on road accidents has shown enormous increases in vehicle crashes due to drowsy feelings. This study comprehensively summarizes all aspects of the drowsy state and its effects during vehicle driving; its symptoms, sleep stages, and the behavioral, physiological and neural activation changes occurring during wakefulness and in the drowsy state. The drowsiness caused by an absence of rest, medicine, substance misuse, or a cerebral issue. It is mostly the result of fatigue which can be both mental and physical. Physical fatigue, or muscle weariness, is the temporary physical failure of a muscle to perform ideally. Mental fatigue is a temporary failure to keep up ideal psychological execution. The onset of mental exhaustion amid any intellectual action is progressive, and relies on an individual's psychological capacity, furthermore upon different elements, for example, lack of sleep and general well-being. Mental exhaustion has additionally been appeared to diminish physical performance. It can show as sleepiness, dormancy, or coordinated consideration weakness. The below table 2.1 shows the symptoms, causes and reduction strategies for drowsy state.

Table no. 2.1

Drowsiness		
Causes	Symptoms	Reduction Strategies
• Poor sleep • Disturbing light inner vision • Acoustic and tension • Lack of oxygen	• Fatigue • Eye pressure • Headache • Eye blinking • Yawning • Pressure on shoulder	• Short sleep • Temperature reduction • Fresh oxygen

2.2 Measures for detection of Drowsiness

There are various measures to determine the level of driver drowsiness. These measures can be grouped into three categories.

2.2.1 Vehicle Behavioral Measure
2.2.2 Physiological Signal Measure.
2.2.3 Driver Behavioral Measure.

The study states that the reason for a mishap can be categorized as one of the accompanying primary classes: (1) human, (2) vehicular, and (3) surrounding factor. The driver's error represented 91% of the accidents. The other two classes of causative elements were referred to as 4% for the type of vehicle used and 5% for surrounding factors. Below figure 2.1 shows flowchart for Measurement of Driver Drowsiness.

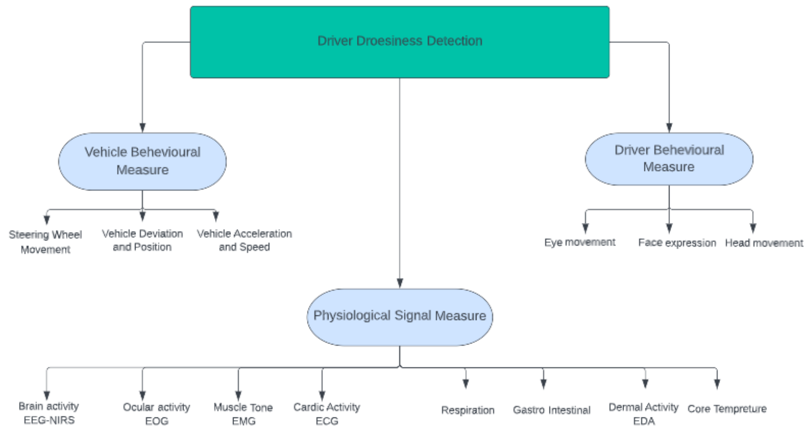

Figure 2.1 flowchart for Measurement of Driver Drowsiness.

2.2.1. Vehicle Behavioral Measure

In this category, a driver's drowsiness is analyzed based on the vehicle control systems, which could include steering wheel movements, braking patterns, and lane departure measurements. Steering wheel measurements tend to yield better results than other vehicle-based methods. Vehicle-based methods are non-invasive, but may not be as reliable in detecting drowsiness accurately because they are dependent on the nature of the road and the driver's driving skills. Vehicle-based measures survey path position, which monitors the vehicle's position as it identifies with path markings, to determine driver weakness, and accumulate steering wheel movement information to characterize the fatigue from low level to high level. In many research projects, researchers have used this method to detect fatigue, highlighting the continuous nature of this non-intrusive and cost-effective monitoring technique.

2.2.2. Physiological Signal Measure

In this category, measurements are obtained by accessing driver's conditions through addition of electronic devices onto the skin. This includes Electroencephalography (EEG), Electrocardiography (ECG) and Electrooculogram (EOG). Although these devices yield highly accurate results, they are not widely accepted because of their practical limitations.

- **Monitoring Heart Rate**: An ECG sensor can be installed in the steering wheel of a car to monitor a driver's pulse, which gives a sign of the driver's level of fatigue indirectly giving the state of drowsiness. Additionally, the ECG sensor can be introduced in the back of the seat.
- **Monitoring Brain Waves**: Special caps embedded with electrodes measures the brain waves to identify fatigue in drivers and report results in real time. Then each brain waves can be classified accordingly to identify drowsiness.
- **Monitoring muscle fatigue**: As muscle fatigue is directly related to drowsiness. We know during fatigue the pressure on the steering wheel reduces and response of several muscle drastically reduces hence it can be measured by installation of pressure sensors 7 at steering wheel or by measuring the muscle response with applied stimuli to detect the fatigue.

- **Monitoring eye movements**: Invasive measurement of eye movement and eye closure can be done by using electro oscillogram but it will be very uncomfortable for the driver to deal with. Though this method gives the most accurate results regarding drowsiness. But it requires placement of several electrodes to be placed on head, chest and face which is not at all a convenient and annoying for a driver. Also, they need to be very carefully placed on respective places for perfect result.

2.2.3. Driver Behavioral Measure

This category consists of behavioral or computer vision measures that tend to reliable than vehicle-based because they focus on the person rather than the vehicle. Furthermore, behavioral measures are non-invasive and more practical than physiological measures. Here, information is obtained by using cameras to detect slight changes in driver's facial expression. As behavioral measures are non-invasive in nature, they are becoming a popular way of detecting drowsiness.

Certain behavioral changes take place during drowsing like:

1. Yawning 2. Amount of eye closure 3. Eye blinking 4. Head position

LITRETURE SURVEY

Drowsiness interferes with people's daily lives and can cause ineffective work or automobile accidents that cause harm to persons and property. Particularly when it involves public transit, it may get worse. As a member of the Department of Land Transport (DLT), the author is aware that tiredness might contribute to auto accidents. Therefore, an efficient sleepiness detection system is carried out as a result of this worry. This paper's goal is to create an algorithm that can assess facial structure from video data while also spotting tiredness. The system is also used to assess the effectiveness of the algorithm. The outcome demonstrates that employing facial landmarks may assist in efficiently generating the eyes and mouth component, which can assist in creating equations to accurately evaluate sleepiness by using Nvidia Nano Jetson. The Nvidia Jetson Nano is a tool that tracks eye closure gestures that last longer than 1.5 seconds (35 FPS) and mouth opening motions that last longer than 50 FPS (2 seconds). The potential for tiredness is indicated, alerting drivers. Therefore, it is recommended that the government support the system and adopt Thailand policy in order to decrease automobile accidents caused by intoxication.

Driver weariness or sleepiness is one of the major factors in accidents. Eliminating sleepiness promotes road safety and reduces accidents. There are several different ways to analyze tiredness in a driver, including yawning, eye closing, and head movement. The approach that is suggested in this research, known as the "percentage eye closure" method, focuses on eye closure (PERCLOS). A parameter level is established by PERCLOS to identify sleepiness. The ViolaJones detector is used for the detection, and it separates the driver's face and the detector's picture of the eye.

Driver fatigue is thought to be a significant factor in a significant number of deaths globally. Therefore, creating a system that can aid in reducing such incidents becomes vital. When compared to when they are awake, the driver's facial characteristics show clear alterations when they are sleepy. After tracking the driver's physiological status, the method suggested in the research focuses on detecting and warning the driver. We used a non-intrusive monitoring technique, where the operator's mouth shape (yawn) and eye blinking are detected in real-time, and if the operator's eyelids are closed for longer than the threshold value, or if the operator is yawning, or the driver's status is assumed for safeguards if both of them are discovered at the same time. The Viola-Jones algorithm is utilized to detect face characteristics in the suggested system, which was developed using the Python programming language and the OpenCV image processing programmed.

PROPOSED METHDOLOGY

Every now and then, road accidents are brought on by drowsy drivers. Face movement and eye blink rates are recorded to analyze the driver's various behavioral or visually based attitudes. Most of this eye blinking is focused on identifying driver drowsiness Without any sign of exhaustion, an EAR's threshold value is above 0.25. When a driver shuts down automatically, the EAR threshold value drops below the predetermined range. The quantity of video frames with the driver's eyes closed is represented by the threshold value of a drowsy eye blink sample. The driver's drowsiness is identified if the number of consecutive counting frames rises above the threshold value. When the Raspberry Pi camera module is successfully integrated, it continuously records every facial movement made by the driver. The proposed work specifically focuses on the driver's behavior measurements with severity measurements of collisions. The Raspberry Pi Model B and Pi camera modules are used to take a persistent recording of the facial landmarks that are localized through facial landmark points, allowing the EAR to be calculated with accuracy. Therefore, this system is a very reliable way to both send the owner an alert message and to take pictures of the driver's drowsiness.

SYSTEM ARCHITECTURE

The first step in the process is the detection of the face, then the eyes. It is then transformed into an image that can be used to determine whether or not the eyes are closed. In order to determine whether the eyes are present, eye features are extracted when the eyes are detected. eyes can be open or shut. After determining whether or not the driver is sleepy, the alarm signal is turned on. If the vertical distance between the eyes is minimum, the driver is drowsy; if the vertical distance between the eyes is maximum, the driver is not drowsy and there won't be an alarm. This is how the project's driver drowsiness detection system works in its entirety.

Following fig 4.1 shows the flowchart for entire process of drowsiness detection system from starting to termination of program.

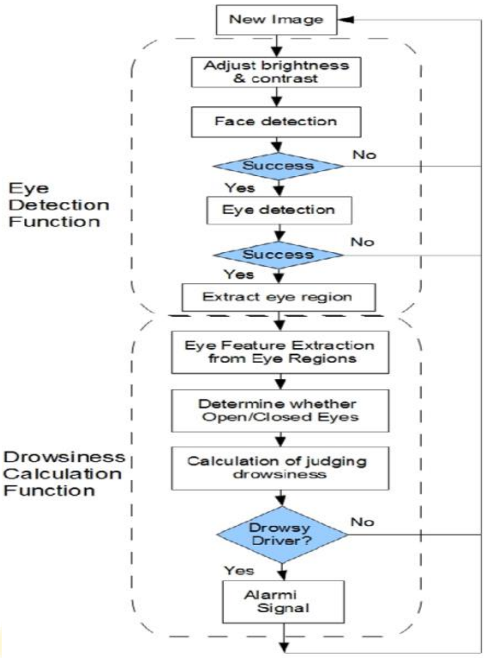

Figure 4.1: Flowchart for entire process of drowsiness detection system

ALGORITHM STAGES

- Image Capture: Utilizing a camera module introduced inside the automobile we can get the picture of the driver. Despite the fact that the camera creates a video clip, we have to apply the developed algorithm on each edge of the video stream. The used camera is a low-cost camera module. e. The 5 MP camera module is shown below in fig 2.3.

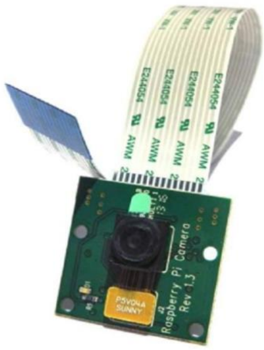

Figure 2.3: Camera used for implementing drowsiness detection system

Features of the 5MP Camera Module Board

The Raspberry Pi 5MP camera board is a custom designed camera board that is equipped with a flexible ribbon cable, making it compatible with Raspberry Pi boards. The camera board is integrated with a fixed lens that has a resolution of 5 megapixels. This camera board can capture beautiful moments with a resolution of 2592 x 1944 pixels and can record high quality videos that supports 1080p at 30fps, 720p at 60fps and 640x480p 60/90 format.

Specification

Table No. 2.2 Specification of camera module

Camera Module	5MP Omni vision 5647
Photo Resolution	2592 × 1944 Pixels
Video	1080 at 30fps, 720 at 60fps, 640x480p 60/90
Size	25mm × 23mm × 8mm
Weight	3 grams
Camera Serial Interface	15 – pin MIPI (Mobile Industry Processor Interface)
Compatibility	Raspberry Pi Model A, B and B+

1. Dividing into Frames: In this project, we are dealing with real time situation where video is recorded and has to be processed. But the processing or application of algorithm can be done only on an image. Hence the captured video has to be divided into frames for analyzing.

2. Face Detection: In this work, a face detection algorithm was developed, where an improved cascaded adobos classifier based on the extended set of Haar-like features is utilized to automatically recognize the driver's face. In this stage, we detect the region containing the face of the driver. A specified algorithm is for detection of face in every frame.

3. Eye Detection: After the successfully completion of face detection, the next step is to detect the driver's eye. In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it's really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features. Generally, Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

4. State of eyes: In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement. It is achieved by an algorithm which will be clarified in the later parts. We channelize a warning message if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye.

METHODOLOGY

It deals with implementation of Eigen face approach which includes mathematical description of its subparts such as Eigen Values and Eigen Vectors, Face Image Representation, Mean and Mean Centered Images, Covariance Matrix and Eigen Face Space.

- **Principal Component Analysis (PCA)**

PCA basically removes the variables to reduce redundancy. So, after reduction of variables, we will get less variables named as Principal Components. Principal components will generally represent all the variables present in the obtained variable. But it only reduction of variables does not solve the purpose. Main Problem appears when we try to achieve face recognition in a more and high dimensional space. The main objective of PCA is to decrease the number of dimension as well as retain more and more possible variation in the given data set. But we know that reduction in dimension results in information loss as information are directly linked with dimension. Hence, we can overcome the problem of data loss by choosing the best principal components as main principal components determines the low dimension. Though use of PCA has many advantages but mostly it is used for eigen face approach. In eigen face approach the reduction of size of the data base is achieved for recognizing the test images.

- **Eigenfaces**

Fundamentals of the Eigenfaces algorithm were first presented by Sirovich and Kirby in their 1987 paper, Low-Dimensional Procedure for the Characterization of Human Faces. The first step in the Eigenfaces algorithm is to input a dataset of N face images. For face recognition to be successful (and somewhat robust), we should ensure we have multiple images per person we want to recognize.

Let's now consider an image containing a face:

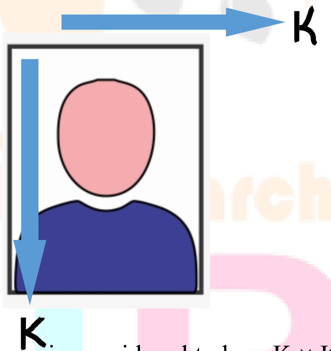

Figure 4.2: Each face in our image is considered to be a K × K image.

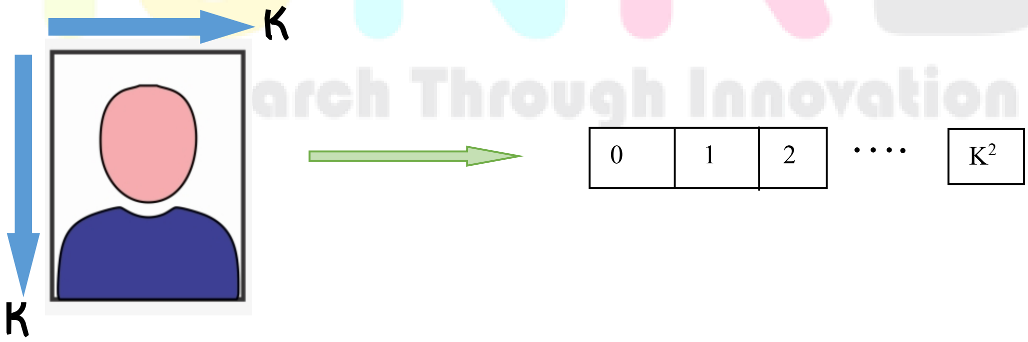

Figure 4.3: We can take our K×K image and then flatten it into a single "feature vector" of raw pixel intensities

Again, all we have done here is taken a K×K image and concatenated all of the rows together, forming a single, long list of grayscale pixel intensities. After each image in the dataset has been flattened, we form a matrix of flattened images like this, where Z is the total number of images in our dataset:

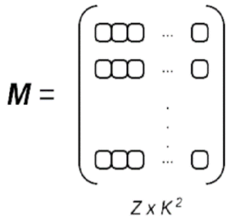

Figure 4.4: Our entire CALTECH Faces dataset represented as a matrix with Z rows, the number of images, and K-squared columns, the flattened image representation

Our entire dataset is now contained in a single matrix, M.
Given this matrix M, we are now ready to apply Principal Component Analysis (PCA), the cornerstone of the Eigenfaces algorithm.

A complete review associated with the linear algebra underlying PCA is outside the scope of this lesson (for a detailed review of the algorithm, please see Andrew Ng's discussion on the topic), but the general outline of the algorithm follows:

1. Compute the mean of each column in the matrix, giving us the average pixel intensity value for every (x, y)-coordinate in the image dataset.
2. Subtract the from each column — this is called mean centering the data and is a required step when performing PCA.
3. Now that our matrix M has been mean centered, compute the covariance matrix.
4. Perform an eigenvalue decomposition on the covariance matrix to get the eigenvalues and eigenvectors.
5. Sort by, largest to smallest.
6. Take the top N eigenvectors with the largest corresponding eigenvalue magnitude.
7. Transform the input data by projecting (i.e., taking the dot product) it onto the space created by the top N eigenvectors — these eigenvectors are called our eigenfaces.

Eigen values and eigen vectors:

In linear algebra, a linear equation in matrix form is represented by Ax= D. The eigenvectors of a linear operator are non-zero vectors which, when operated by the operator. The result of this is a scaler multiple of them. For the eigen vector X the obtained scaler called eigen value (λ). A vector which is paralleled by linear transformation is called an Eigen vector. It is one of the properties of matrix. When we calculate a matrix on it then the magnitude of the vector is changed. The direction of vector remains as it is. So, we define as Ax = λ x, where A is represented as a vector 20 function. Then transforming the RHS part and writing it as (A – λ I) x = 0, where I am called the identity matrix. The above form is a homogeneous equation and is fundamental part of linear algebra. Existence of non-trivial solution is decided by considering that if and only if Det (A – λ I) = 0, where Det represents determinant. When it is evaluated, we deal with the polynomial of degree n. This is known as the characteristic polynomial of A. If we represent the dimension of A by N x N then the solutions result in n roots of the characteristic polynomial. So it gives n Eigen values of A which satisfy the A xi = λ I xi , where I = 1,2,3,.....n. If the obtained eigen values are all distinct then we get an associated linearly independent eigen vectors with unique directions.

Face Image Representation:

In this approach we represent set of let's say m images of each having size N ×N. This is done by vectors of size N2. We represent each face Γ1, Γ2, Γ3… In. All those ብobtained feature vectors are stored in the matrix with size N × N. One example is shown below which describes the entire process. For example:

$$\begin{bmatrix} 3 & 7 \\ 6 & 5 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \\ 6 \\ 5 \end{bmatrix}$$

Mean and mean centered Image and Covariance matrix:

We calculate the average face by

Ψ=Σ, Γ𝑛∞……………………………………. (1)
𝑛=1

Then we find the difference of each face from their average face which is

i = Γ i – Ψ………………………………. (2)

We can construct a covariance as mentioned below.

C = AAT………………………. (3)

where A = [ϕ1, ϕ2,… ϕm] of size N2 × N2. As we can see that the size of covariance matrix will be N2 × N2. which is huge actually and we need to find the Eigen vectors for the covariance matrix. But the large size make it time consuming and tedious. To encounter this problem we go for calculating ATA .

Now let's consider the eigenvectors Vi of A TA such that A TA Xi = λiXi.

The eigenvectors Vi of A TA are X1… Xn2. Now for simplifying we multiply the above equation with A both sides and we get.

A TAA Xi = AλiXi,………………..(4)
A TA (A Xi) = λi(AXi)………………..(5)

From above we clearly see that Eigen vectors responding to A TA is now firmly computed by reduction in dimension where AXi is the Eigen vector and λi is the Eigen value.

IMPLEMENTATION

The implementation of the drowsiness detection system with the hardware. The hard ware used is Raspberry Pi and a camera module. In this chapter we described use of hardware with its feature, installation and setup procedure. We described how the entire process of drowsiness detection occurs. For this we used libraries of OpenCV like pygmy, karas were used. Different .xml files of OpenCV is operated on the input and provide the required result. Different .xml files of OpenCV is operated on the input and provide the required result The .xml files written for drowsiness detection includes face and eye detection which basically done by algorithm by Adrian Rose Brock. That algorithm includes Haar features, Formation of integral Image and Cascading.

About Raspberry Pi:

The Raspberry Pi is a credit card-sized computer with an ARM processor that can run Linux. This item is the Raspberry Pi Model B, revision 2.0, which has 2 GB of RAM, an Ethernet port, HDMI output, RCA composite video output, audio output, two USB ports, and 0.1″-spaced pins that provide access to general purpose inputs and outputs (GPIO). The Raspberry Pi requires an SD card with an operating system on it. The Raspberry Pi is very popular, with lots of example projects and information available online.

The Raspberry Pi was designed by the Raspberry Pi Foundation in order to provide an affordable platform for experimentation and education in computer programming. The Raspberry Pi can be used for many of the things that a normal desktop PC does, including word-processing, spreadsheets, high-definition video, games, and programming. USB devices such as keyboards and mice can be connected via the board's two USB ports. With its 0.1″-spaced GPIO header and small size, the Raspberry Pi also works as a programmable controller in a wide variety of robotics and electronics applications. Over two million Raspberry Pi have been sold.

Features:

1) 2 GB of RAM
2) Ethernet port
3) Four USB ports
4) Two video output options: HDMI or composite
5) 3.5 mm audio output jack
6) Camera Interface
7) Display Interface
8) Power Supply Port

Specification:

Processor Chipset	1.2GHz 64-bit quad core ARMv8
RAM	2 GB
Storage	Micro SD
USB 2.0	4 Ports
Ethernet Port	Yes
Wireless LAN	802.11n
Bluetooth	Yes
GPIO	40
Power Draw/Voltage	1.8A at 5v

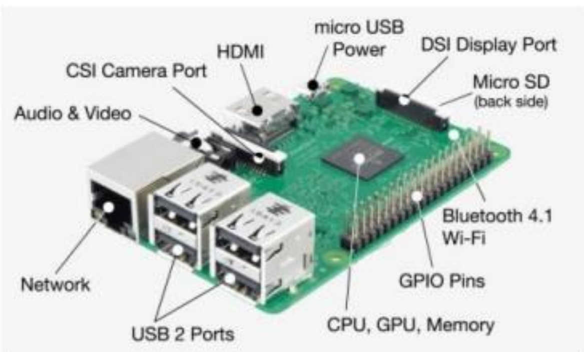

Figure 5.1: figure shows different components of Raspberry Pi with 40 GPIO pins

1. The "haar cascade files" folder consists of the xml files that are needed to detect objects from the image. In our case, we are detecting the face and eyes of the person.

2. The models folder contains our model file "cnnCat2.h5" which was trained on convolutional neural networks.
3. We have an audio clip "alarm.wav" which is played when the person is feeling drowsy.
4. "Model.py" file contains the program through which we built our classification model by training on our dataset. You could see the implementation of convolutional neural network in this file.
5. "Drowsiness detection.py" is the main file of our project. To start the detection procedure, we have to run this file.

Tools used for implementation of face detection:

1. Raspbian operating system.
2. Python IDLE.
3. OpenCV (Open-source Computer Vision) for python with Harr object detection trainer.

Components used:

1. Raspberry Pi Model B
2. Camera Module.
3. HDMI to VGA Cable.
4. C-Type Cable.
5. Cooling system along with fiber box.

Stages for face detection:

1. Haar features used in face detection.
2. Haar Cascade Classifier.
3. Take image as input from a camera.
4. Detect the face in the image and create a Region of Interest (ROI).
5. Detect the eyes from ROI and feed it to the classifier.
6. Classifier will categorize whether eyes are open or closed.
7. Calculate score to check whether the person is drowsy.

Haar features used in face detection:

Haar features are sequence of rescaled square shape functions proposed by Alfred Haar in 1909. They are similar to convolution kernels taught in the Convolution Neural Networks course. We will apply these haar features to all relevant parts of face so as to detect human face.

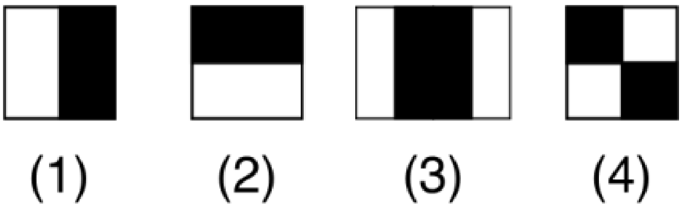

Figure 5.2: Figure shows example of Haar features

As we see in the above figure, there are edge features (1 and 2), line features (3). They are white and black pixels' images (value 0 or value 1). But usually we have greyscale/color image (pixel value range from 0 to 255). For now, assume ideal scenario that we have black and white image. To detect eyebrow, we will use Haar feature image (1) because forehead and eyebrow form lighter pixels- darker pixel like image. Similarly, to detect lips we use similar to Haar like feature image (3) with lighter-darker-lighter pixels. To detect nose, we might use darker-lighter Harr like feature from image (1). And so on.

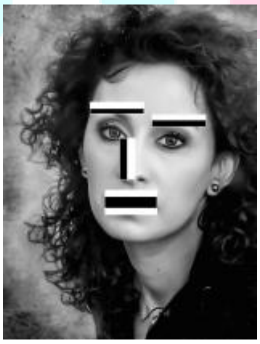

Figure 5.3: Figure shows haar features applied on relevant parts of face

Let's look at some computations that are required. For black and white image (refer first box below), pixel values are 0 or 1 (ideal case) but in real cases we have normalized greyscale image as shown in bottom box containing pixel values.

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

0.1	0.2	0.6	0.8
0.3	0.2	0.6	0.8
0.2	0.1	0.8	0.6
0.2	0.1	0.8	0.9

Figure 5.4: Pixel values: 0 and 1 for ideal case (Top box), Normalized greyscale values for real cases (Bottom box)

According to Viola-Jonas algorithm, to detect Haar like feature present in an image, below formula should give result closer to 1. The closer the value is to 1, the greater the change of detecting Haar feature in image

$\Delta = dark - white = 1\ n \sum 1(x) - n\ dark\ 1\ n \sum 1(x)\ n\ white$ ....(1)

Ideal case: Delta = (1/8) *(8) — (1/8) *0 = 1
Real case: Delta = (1/8) *(5.9) — (1/8) *(1.3) =0.575

(For greyscale image, assume we have set White-Dark threshold to 0.3. Meaning pixels with value less than or equal to 0.3 are considered white and anything greater that 0.3 is considered as dark). We can define another threshold parameter to detect edge or Haar feature. We are calling it Delta (this is different from White-Dark threshold). Assume here we have set threshold to 0.5. Any delta value greater than 0.5 detects Haar feature. In this fashion we can apply, get and detect most relevant features on a given image like eyebrow, lips, nose etc.

Haar Cascade Classifier:

They are series of classifiers or features (as we have seen above) used to identify object in an image. Using sliding windows and number of haar features (increases as number of stages increase), finally leading to detect face or not. There are total 38 stages defined for Viola Jonas Method. Depending upon the sliding windows size and face location, number of features, face can be detected at a certain stage.

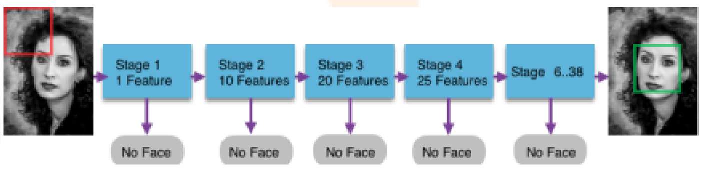

Figure shows Haar Cascade Classifiers method

Take Image as Input from a Camera:

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, cv2.VideoCapture(0) to access the camera and set the capture object (cap). cap.read() will read each frame and we store the image in a frame variable.

Detect Face in the Image and Create a Region of Interest (ROI):

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier face = cv2.CascadeClassifier('path to our haar cascade xml file'). Then we perform the detection using faces = face.detectMultiScale(gray). It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face.

Detect the eyes from ROI and feed it to the classifier:

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in leye and reye respectively then detect the eyes using left_eye = leye.detectMultiScale(gray). Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code. l eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into eye.

Classifier will Categorize whether Eyes Are Open or Close:

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the colour image into grayscale using r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY). Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images cv2.resize(r_eye, (24,24)). We normalize our data for better convergence r_eye = r_eye/255 (All values will be between 0-1). Expand the dimensions to feed into our classifier. We loaded our model using: - model = load_model('models/cnnCat2.h5'). Now we predict each eye with our model lpred = model.predict_classes(l_eye). If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.

Calculate Score to Check whether Person is Drowse:

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of the person. A threshold is defined for example if score becomes greater than 15 that means the person's eyes are closed for a long period of time. This is when we beep the alarm using sound.play().

ADVANTAGES

- The detected abnormal behavior is corrected through alarms in real time.
- Component establishes interface with other drivers very easily.
- Life of the driver can be saved by alerting him using the alarm system.
- Speed of the vehicle can be controlled.
- Traffic management can be maintained by reducing accidents.
- Practically applicable.

APPLICATION

- This system can be used in factories to alert the workers.
- If found drowsy, the alarm system gets activated and the driver is alerted.
- If there are any obstacles it is alerted to the driver.
- This system can also be used for railway drivers.

CONCLUSION

'Real time Drowsiness Detection System', this project is purely based on to increase the safety of the driver operating on any kind of vehicles. This system can be installed in all kind vehicles irrespective of whether it will run, fly or float on water. The system will detect the fatigue on the driver's face and will alert him/her for the same. This project has a huge business potential at it can be installed in every vehicle. The main working of drowsiness detection will remain same in all kind of vehicles and the additional security features will be differed depending upon the type of vehicle. This project has a huge opportunity to exist in the market. The system when installed in any kind of vehicles will completely transform its safety value with a huge margin, and would be compatible even with the cost at the final stage. It can create a huge market and business opportunities even for the locals and small businessmen. By using all four libraries of python we created a base program to detect sleepiness and now we will be interfacing this base program in raspberry pi -3 along with camera module to create real time drowsiness detection system.

FUTURE SCOPE

The future works may focus on the utilization of outer factors such as vehicle states, sleeping hours, weather conditions, mechanical data, etc., for fatigue measurement. Driver drowsiness pose a major threat to highway safety, and the problem is particularly severe for commercial motor vehicle operators. Twenty-four-hour operations, high annual mileage, exposure to challenging environmental conditions, and demanding work schedules all contribute to this serious safety issue. Monitoring the driver's state of drowsiness and vigilance and providing feedback on their condition so that they can take appropriate action is one crucial step in a series of preventive measures necessary to address this problem. Currently there is not adjustment in zoom or direction of the camera during operation. Future work may be to automatically zoom in on the eyes once they are localized.

REFERENCE

[1] Parra Alonso, R. Izquierdo Gonzalo, J. Alonso, A. Garcia Morcillo, D. FernandezLlorca, and M. A. Sotelo, "The experience of drivertive-driverless cooperative vehicle team in the 2016 gcdc," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 4, pp. 1322–1334, 2017.

[2] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," Remote Sensing, vol. 9, no. 4, p. 312, 2017.

[3] R. P. Balandong, R. F. Ahmad, M. N. Mohamad Saad, and A. S. Malik, "A review on EEG-based automatic sleepiness detection systems for driver," Ieee Access, vol. 6, pp. 22908–22919, 2018. [4] F. Rohit, V. Kulathumani, R. Kavi, I. Elwarfalli, V. Kecojevic, and A. Nimbarte, "Realtime drowsiness detection using wearable, lightweight brain sensing headbands," IET Intelligent Transport Systems, vol. 11, no. 5, pp. 255–263, 2017.

[5] A. Kulathumani, R. Soua, F. Karray, and M. S. Kamel, "Recent trends in driver safety monitoring systems: state of the art and challenges," IEEE Transactions on Vehicular Technology, vol. 66, no. 6, pp. 4550–4563, 2017.

[6] J.-J. Yan, H.-H. Kuo, Y.-F. Lin, and T.-L. Liao, "Real-time driver drowsiness detection system based on PERCLOS and grayscale image processing," in Proceedings of the 2016 International Symposium on Computer, Consumer and Control (IS3C), pp. 243– 246, Xi'an, China, July 2016