Real-Time Driver Drowsiness Detection using an Embedded system

Kashish jain(2021JCS2240)
K.Laxman(2018CS50408)
Varshitha Reddy(2019CS51065)

Abstract:

Vehicle Accidents due to driver drowsiness are common in our country. Generally, Indian truck drivers disobey company policy on the maximum number of driving hours, due to which their fatigueness results into drowsiness and loss of concentration which eventually results into vehicle accidents. However, the lives of many such drivers could be saved, if their drowsiness can be detected early. Any vehicle driver, mine worker, or person on bed can use our developed prototype. To detect drowsiness, many methods exist, but not all of them are efficient as some methods require costly sensors or some are based upon inefficient algorithms. Through this project we have implemented an efficient drowsiness detection algorithm using python, OpenCV,dlib models, and a metric EYE ASPECT RATIO(EAR) on raspberry pi 4. EAR is an efficient, easy to implement, fast-eye blink detection method as compared to other existing methods. Also in order to minimize the memory and computational overheads, we implemented our drowsiness detection system on the arduino board as well. Our developed prototype will significantly reduce the number of road accidents without requiring any physical contact with drivers.

Keywords: **Raspberry pi 4, arduino mega board, IR sensor, buzzer, picamera2, python, opencv, dlib**

Introduction:

Advancement in technology enhances our standard of living. Each year about 1 lakh cases are reported involving vehicle accidents due to drowsiness which is 40% of total road accidents. In our country, generally, drivers don't follow company policies and drive vehicles for more than the permitted time which results in drowsiness. Drowsiness is mainly caused due to fatigueness, loss of concentration, lack of sleep or brain disorder, etc. Driver drowsiness can result in vehicle accidents and loss of lives. Also workers at the mining site suffer a lot due to this drowsiness problem. So there is an urgent need to handle this drowsiness problem.

Although there does not exist any direct method for drowsiness detection, there are several indirect methods for its detection. In the first part of this project, we have used raspberry pi to develop a real-time application for drowsiness detection.

We have used a Raspberry pi board with some peripherals devices such as Picamera and buzzer. Many algorithms have been developed to encounter drowsiness problems which commonly involves face detection and eye feature extraction. But Existing algorithms involve less efficient approaches for eye blink detection, which is key to drowsiness detection. Traditional Eye blink detection methods such as finding the white area of the eye first and then detecting its disappearance for a certain time period is an inefficient method. Rather a new method -Eye aspect ratio(EAR)" introduced in the paper[11], can be used for efficient eye blink detection. We have used OpenCV and python to detect the driver's drowsiness and dlib to detect the facial landmarks. The model Shape_predictor_68_landmarks.dat is being used to train our dlib shape predictor method which is then used to detect and extract the facial landmarks.

Facial landmarks are determined to locate important parts of the face such as the eye, nose, ear, and lips. We have used 68 facial landmarks in our approach. Although for blink detection, we require eye localization only. The eye is identified by 6 coordinates(x,y), once the eye is detected using a facial landmark detection algorithm, the Eye aspect ratio is computed to determine whether the eye is closed or not. So by tracking eye movements we can detect the blinking of eyes and thus detect drowsiness. Whenever eyes are detected blinking, a buzzer will be sounded to wake up the driver and disturb his sleep. To develop this real-time application, we require the camera to be fixed on the dashboard of the car. This camera will capture a live stream of the driver's face and continuously send it to our application. From these video frames, the trained Dlib facial landmark detector model would spot 68 facial landmarks, and thus eye would be detected after which the computed Eye aspect ratio would be compared to a constant on basis of which drowsiness will be detected and if drowsiness is detected then the driver will be alerted by an alarm/buzzer.

In the second part of this project, we have extended our work by overcoming the limitations of our developed prototype build using raspberry pi. We have focused primarily on reducing memory and computational overheads by making changes in our code and switching to microcontroller-arduino Mega board. As compared to microprocessors, Microcontrollers require less memory, less computing power and are also cheaper. We have utilized an arduino mega

board for improving our drowsiness detection system. This prototype can work 24/7 at low cost and power as compared to our previous raspberry pi prototype. We have written separate code for drowsiness detection which works on an arduino board. Also we have used IR sensor which are attached to eyeglasses that would be worn by the driver. This IR sensor will be detecting eye blinks of the driver and determines drowsiness. If the driver is detected drowsy, buzzer alarms and wakes up the driver. This prototype does not require any picamera unlike the raspberry pi prototype. The overall cost of using an arduino board with IR sensor is less than using a raspberry pi board with picamera2. We have been successfully able to reduce memory and computational overheads of our driver drowsiness detection system. Proofs for this are shown in the Evaluation section. We have shown complete cost analysis of both the models in the Evaluation section. From Cost analysis , we can conclude the arduino model is pocket friendly also.

BACKGROUND WORK:

In this survey, we came to know about many technologies and research which are related to the topic of our project. Through this survey, we can understand better which aspect we should focus more on while developing this project. Current drowsiness detection technologies are for blink detection, and yawn detection. CNN, Computer vision, Behavioral measures, and machine learning are used which have different levels of accuracy advantages and challenges. Mainly when it comes to blinking detection and yawn detection, research is done on EAR-based technology and MAR-based technology. Rateb et al.(R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang) in 2018 detected drowsiness of a driver using neural networks and developed an android application to detect drowsiness of the driver. Tereza Soukupova et al. (T. Soukupova and J. Cech) 2016 used the eye-aspect ratio to detect the drowsiness of the driver. They also detailed active and passive systems which were used to detect the drowsiness of a driver. Active systems are reliable but use hardware that is intrusive and expensive such as infrared cameras and passive systems are inexpensive and rely on standard cameras. Vibin Varghese et.al (V. Varghese, A. Shenoy, S. Ks, and K. P. Remya) in 2018 detected landmarks for every frame to calculate the eye aspect ratio using landmark points on the face. After calculating, If the eye aspect ratio is less than the limit of 2 to 3 seconds then the driver is determined as drowsy because eye blink lasts up to 100-400ms.

In [1] the authors discussed the drowsiness system which is based on computer vision for vehicles with web push notifications and buzzer sounds. This system uses an eye-aspect ratio to know whether the driver is sleeping or not. Buzzer sounds are used to alert drivers and web push notifications show nearby coffee shops in order to alert drivers. The trail run system detected the driver's drowsiness and a buzzer was generated and a web notification was sent to the driver

about nearby coffee shops. The limitations of this paper are that they used raspberry pi which does not work at nighttime. Instead, we can use night vision camera

In [2] "the authors discussed how they used DMS (driver monitoring system) to detect the drowsiness of the driver along with different kinds of sensors. Results obtained from the system showed that the system could successfully classify drivers' drowsiness into low, moderate, and severe levels of drowsiness. The limitations of this paper are that the system could not classify drowsiness into moderate and severe levels of drowsiness.

In [3] authors discussed about the system which detects drivers' drowsiness by analyzing the eye blinks of humans using facial landmark detection and making use of eye aspect ratio (EAR). eye aspect ratio is used for fast, easy, and efficient blink detection. The results of this detector showed that the system effectively detected the driver's drowsiness by providing a precise enough estimation of the level of eye openness. As performance cost is very low in this system we can use this system in real-time.

In [4] the authors discussed a driver detection system in which the driver's eye movements and mouth movements are tracked using Haar cascade classifiers. By tracking eye and mouth movements we can detect the drowsiness of the driver. After the driver's drowsiness is detected then a buzzer sounds to alert the driver. Results of this driver drowsiness detection system showed that in 85 % of the cases the system detected faces and facial features accurately which is required. The Limitation of this paper is that system accuracy decreases in bad lighting conditions.

In [12] the authors proposed a method for detecting drowsiness on Arduino microcontroller boards based on Atmega328P for real-time monitoring.. System detects the driver when the driver is drowsy successfully. Results showed that this system used very low power and less complexity with low cost manners . Because it uses low power and has low cost we can use it in vehicles in real time. They used IR sensors for detecting which showed good results even when the driver is at a long distance .

In [14] system detects driver drowsiness using eye blink sensor. Eye blink sensor is continuously working on Arduino board. This detects when eye is closed continuously for 5 seconds. Then a buzzer beeps.

PROBLEM STATEMENT:

To encounter the problem of drowsiness many solutions exist, Although there are no direct methods for measuring drowsiness but indirect methods exist. Such as methods involving fingerprints, behavioral and physiological measures for drowsiness detection. But fingerprint method and physiological methods are inefficient, also require continuous physical contact with the driver's body, and are costly. So there is a need of a drowsiness detection method that does not require physical contact with the driver and is cost-effective. Through this project, we have developed a real time prototype for drowsiness detection. Which is based upon a metric *Eye aspect ratio*, and utilizes dlib, OpenCV. Dlib is a popular library and more accurate than OpenCV for face detection and facial landmarks detection. To detect drowsiness, we need to determine certain eye movements such as eye blinking, which will ensure that driver is sleepy. Eye blinking detection is key to drowsiness detection. So we need to efficiently detect eye blinking of a driver. We will detect the driver's drowsiness by calculating *eye aspect ratio* (EAR) for every frame in real-time. We need to alert the driver whenever he is detected as drowsy, we can use an alarm for this. So Whenever a driver is detected as drowsy then the buzzer sounds to alert the driver, to reduce drowsiness. But using raspberry pi for developing embedded system incurs many overheads such as large memory consumption and computational overheads, so we have to work upon minimizing these overheads . Also raspberry pi is costlier, so we have to find cheaper alternate.

PROPOSED SOLUTION:

For this drowsiness detection project, we have used Opencv and python3. Opencv is very useful for image processing additionally, There are many methods for face detection but we used the dlib library for face detection using the HOG (Histogram of Oriented Gradients)algorithm as the Dlib library is more accurate than OpenCV for face detection.

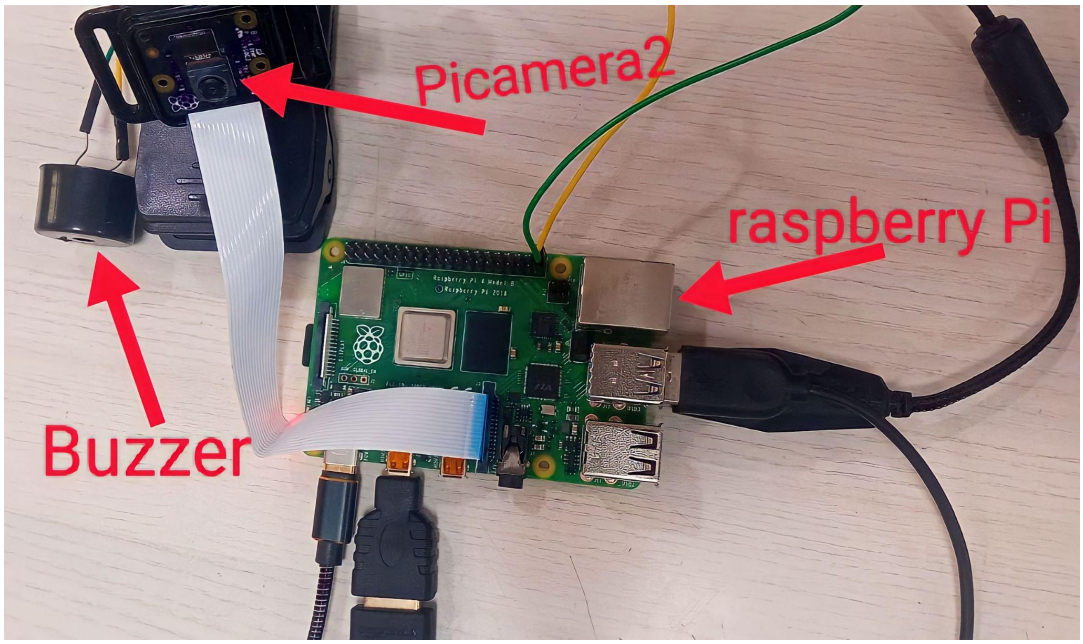
The dlib library is used to detect and isolate the facial landmarks using pre-trained data. We have used the Shape_predictor_68_landmarks.dat model to train the shape predictor method of dlib. Dlib's facial landmark detector is detecting 68 facial landmarks on drivers' facial images from which we are utilizing the eye landmarks in order to localize the eye. Because eye localization is only necessary for eye blink detection.

We have used a metric Eye aspect ratio which is one of the efficient and fast algorithms for detecting eye blink. Captured eye landmarks which are basically 6 (x,y) coordinates points are then fed to this metric *Eye aspect ratio,* which then determines whether the eye is blinked or not. *The eye aspect ratio* computes the ratio of distances between the horizontal and vertical eye landmarks.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

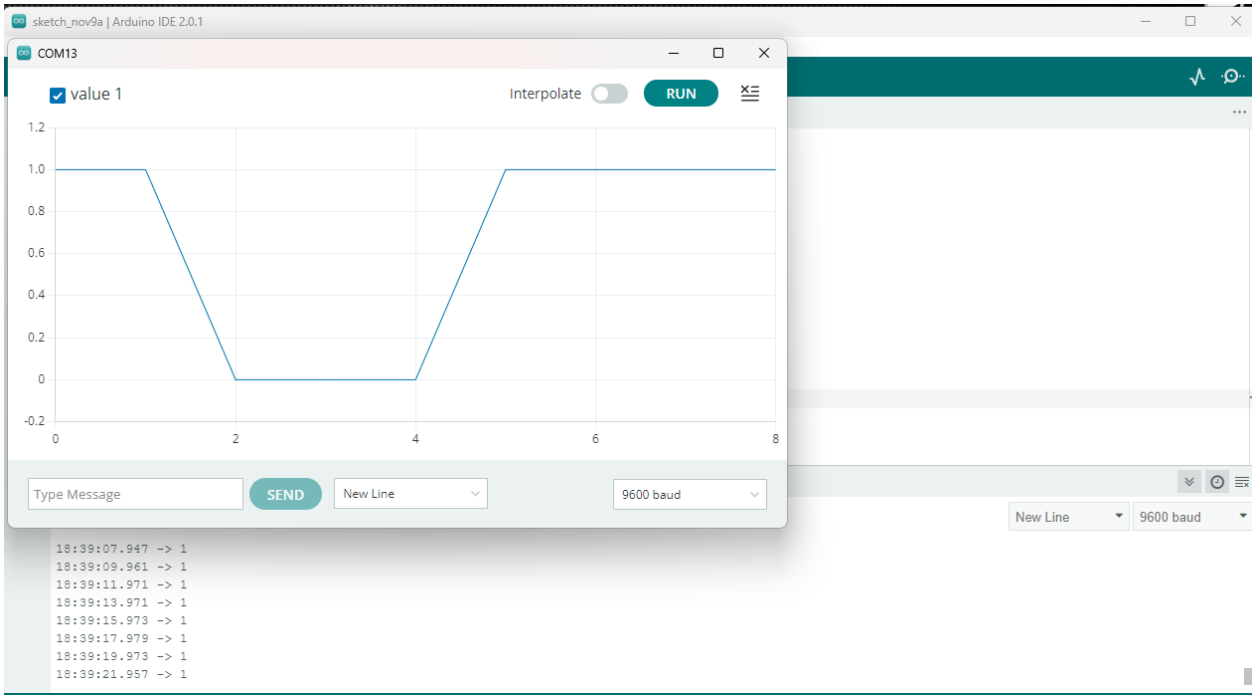
Here p1….p6 are 2D pupil coordinates.EAR value is compared with the threshold value taken as 0.25. If the value is lower than the threshold, the driver is in a drowsy state. In that case, he will be alerted by an alarm. Generally, when eyes are open and the EAR value is nearer to 0.25 so, that is the threshold set.We have installed the picamera2 for taking the real-time input video of the driver to analyze the drowsiness. The facial landmark detection will be detecting facial landmarks for each frame of this real-time video. This picamera2 needed to be fit on the dashboard of the driving vehicle to continuously capture the face of the driver. For the initial setup, we connected the raspberry pi board with a buzzer to alert the driver if drowsiness was detected. We wrote separate code for generating an alarm from the buzzer whenever drowsiness is detected.

Raspberry pi model looks as shown:



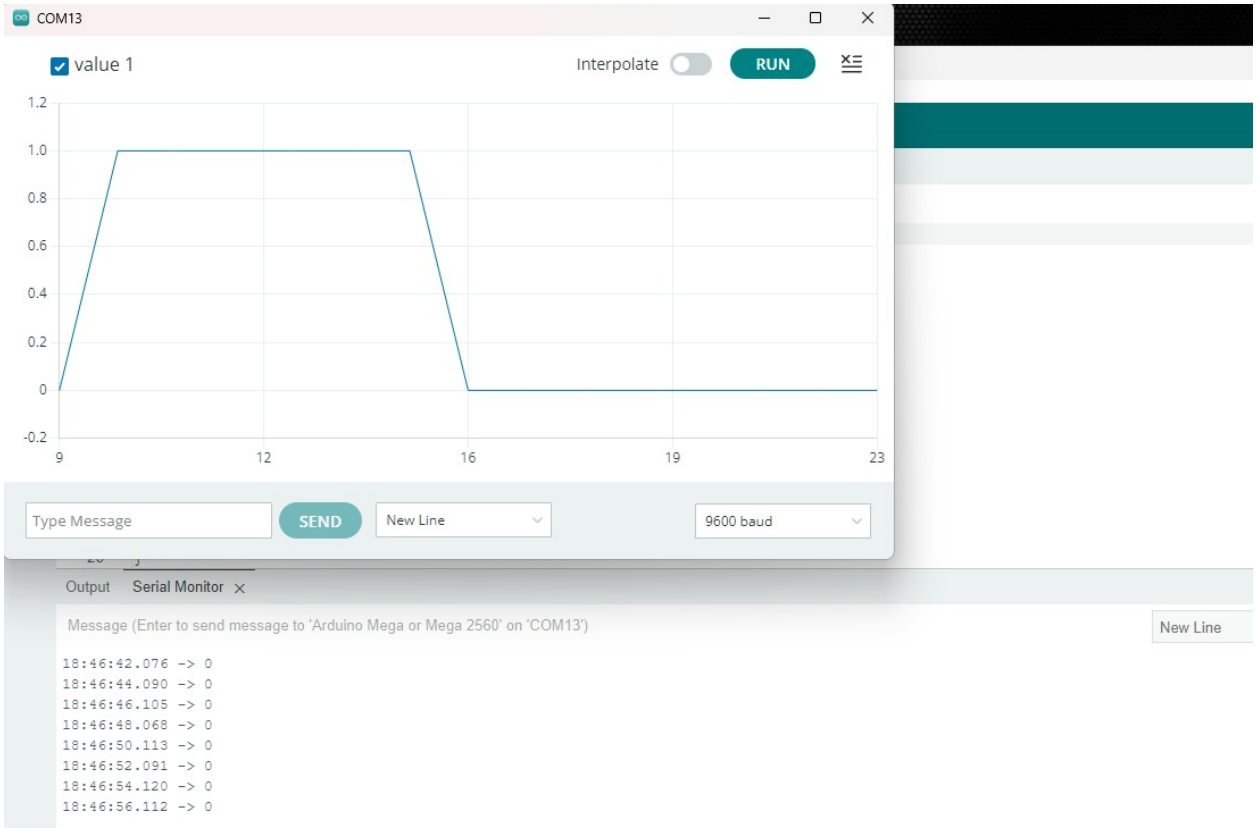
After implementing the raspberry pi based driver drowsiness detection system we worked upon minimizing its memory and computational overheads. In order to minimize memory overheads , we made changes in code and removed GUI part form code. Doing this, affected memory consumption significantly. Results of memory consumption before optimization and after optimization are shown in the table below. Also CPU utilization is shown along. We measured the temperature of pi as well before optimization and after optimization. Results are shown in the

table below. Details on these measurements are given in the EVALUATION section. In order to minimize computational overheads, we switched to an arduino Mega board. Because raspberry pi is a microprocessor and hence performs more computations thus more overheads. Whenever pi starts, the operating system starts and other modules load thus more overheads and hence consumes more power. So we switched to a microcontroller arduino mega board. We wrote separate code for the arduino board for drowsiness detection. We used an IR Sensor to detect eye blinks in order to detect drowsiness. We have shown graphs to demonstrate working of our arduino model.



Graph for normal eye - non sleepy person

In the above graph, we can see square waves with values between 1 and 0. These are the values of the IR sensor. The Wave at 1 represents that when the person's eye is open and whenever the person's eye is normally blinked then the wave value is 0. Generally eye blink is for 2 microsec in human behavior. So the wave value is 0 for 2 microsec intervals, which we can clearly see in the above graph. Whenever a person is drowsy, the eye will be closed for more than 2 microsec and hence the wave value will be 0 for more than 2 micro sec interval , as shown in below graph.

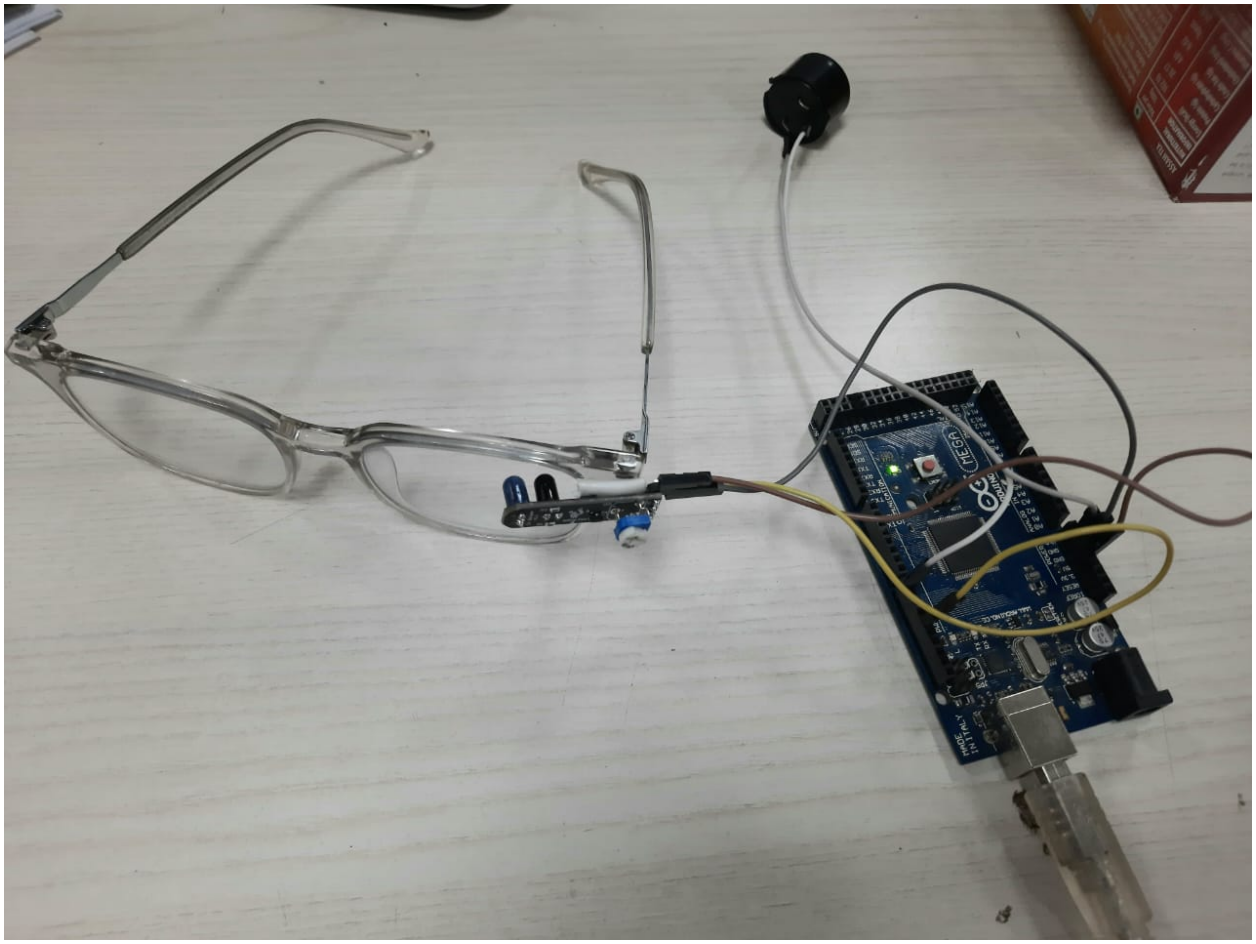


Graph for sleepy/drowsy person(time interval > 2sec)

In the above image, below the graph, in the output terminal we can see digits 0 and 1. These are values given by the IR sensor.

So whenever a person is detected drowsy, a buzzer connected with an arduino board rings up.

Our whole setup for this arduino model is shown below:



Arduino model for drowsiness detection

```
void loop()
{
  int x=analogRead(A0);
  //Serial.println(x);

    if(x<400 && flag==0)
  {
    flag=1;
    t1=millis();
  }
  else if(x>400 && flag==1)
  {
    flag=0;
    t2=millis();
    Serial.println(t2-t1);
    if((t2-t1)>1000)
    {
      digitalWrite(2,HIGH);
      Serial.println("Alert
2!!!!!!!!!!!!!!!!!");
      delay(2000);
      digitalWrite(2,LOW);
    }
    else;
  }
}
```

Arduino code

We have connected a data cable with A0 pin . This model has significantly reduced memory and computational overheads, as shown in below table.

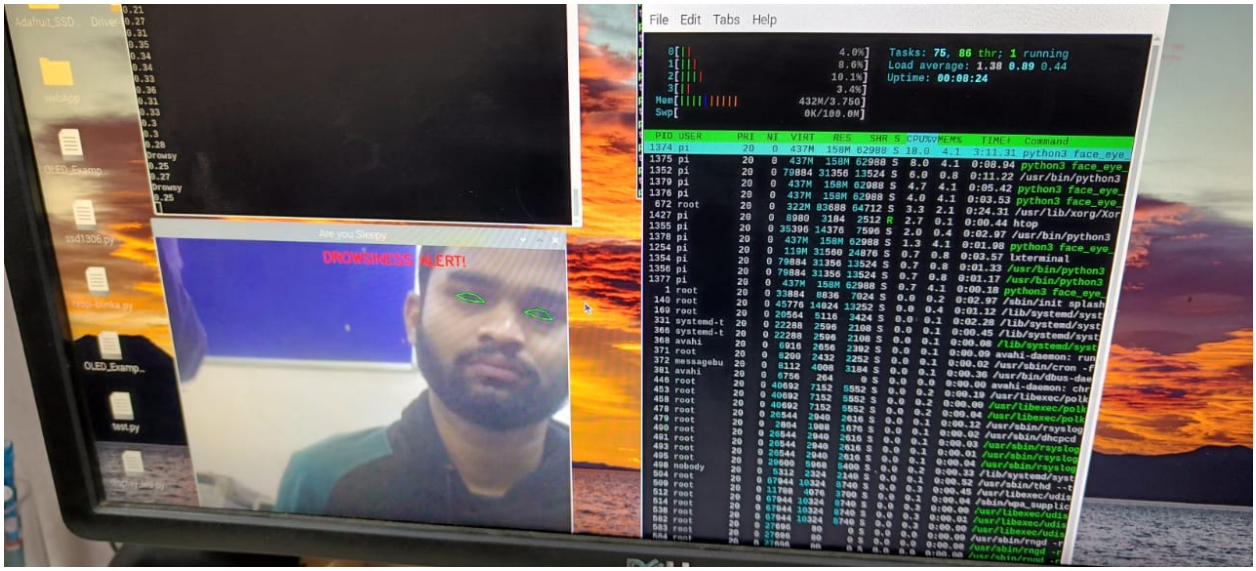
Our device can work for 24x7 hours as we don't know when the driver may go drowsy, unlike raspberry pi which gets heated up early .

Comparison of Readings of the values which shows the **optimization trend** :

Borad	Cpu % Used	Mem% Used	Time+ (ms)	Temp(°C)	Load Avg time
Raspberry with GUI	6	4.1	3:15.17	50.6	1.24
Raspberry without GUI	1.3	3.8	0.00.14	45.7	0.79
Arduino Mega	0.4	0.2	0.00.03	25	0.30

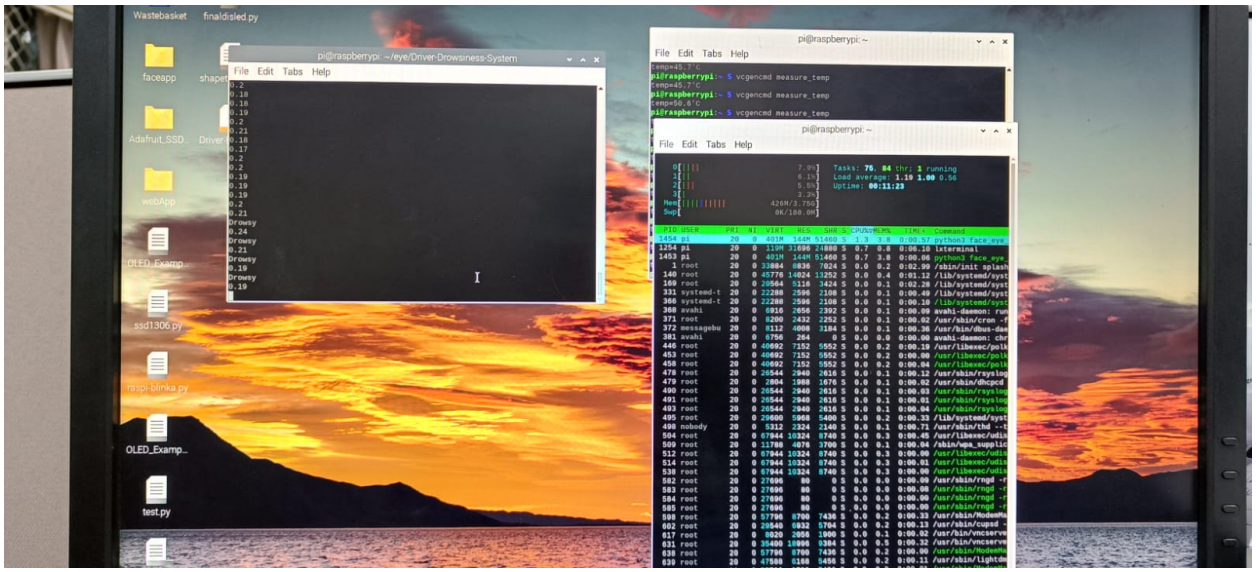
EVALUATION:

While designing any embedded device prototype, its memory consumption plays a significant role in its overall performance. Using any ML model on raspberry pi is not an easy task as ML models utilize more memory and have large running overheads than any non ML code. We are using a non ML code for drowsiness detection so overheads are significantly minimized. But still we can improvise our model. We are using raspberry pi of 4gb. Our raspberry pi prototype for drowsiness detection consumes a large amount of memory. Memory consumption can be seen in below image:

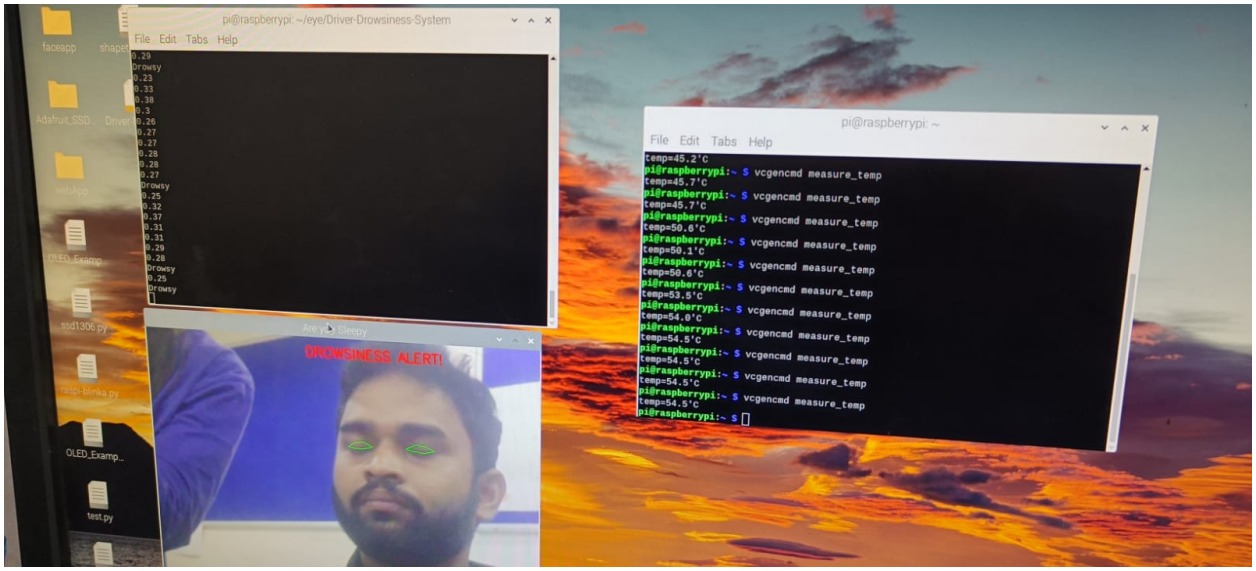


See Black part at top right for memory consumption

From the above image, we can see that our raspberry pi model is utilizing 432 MB out of 3.75 GB available. So if this prototype has to be used on a regular basis, this much memory consumption will create trouble. We tried to improvise our code in order to reduce memory overheads. Also when the driver is driving and using our prototype , there is no use of GUI to see face capturing. Hence, We removed code for the GUI part. After doing this much, we again measured our memory consumption. We can see in below image, that now our prototype is consuming :426 Mb

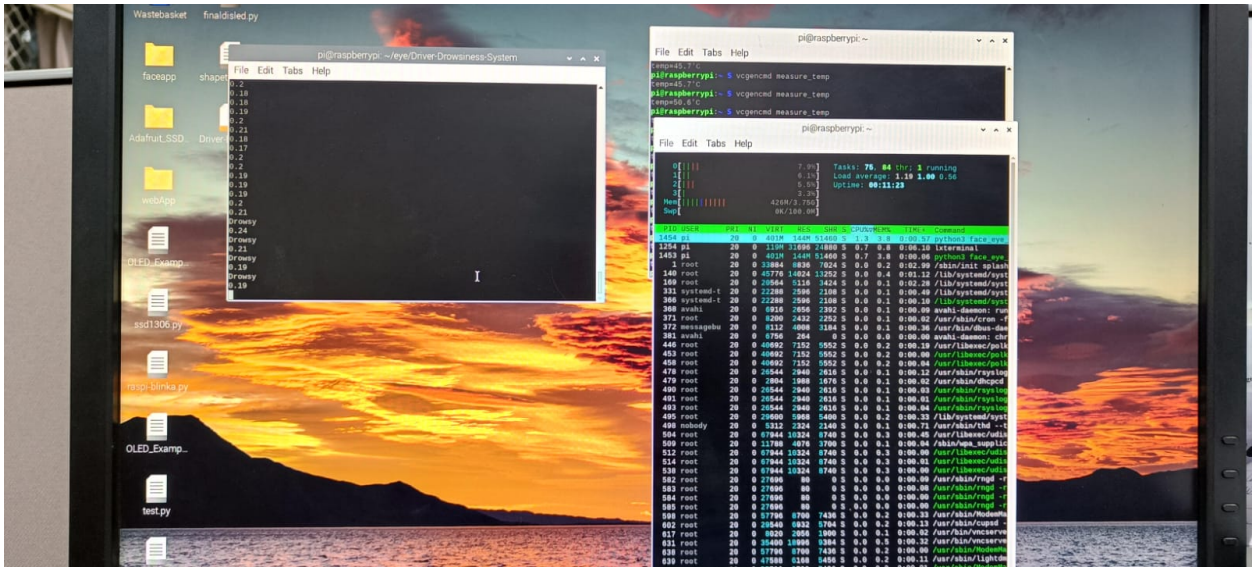


Also raspberry pi gets heated up easily after using it for 1 hour continuously So ,we measured pi temperature before and after. If we are using pi with GUI then the temperature of pi is rising to **55 °C**.



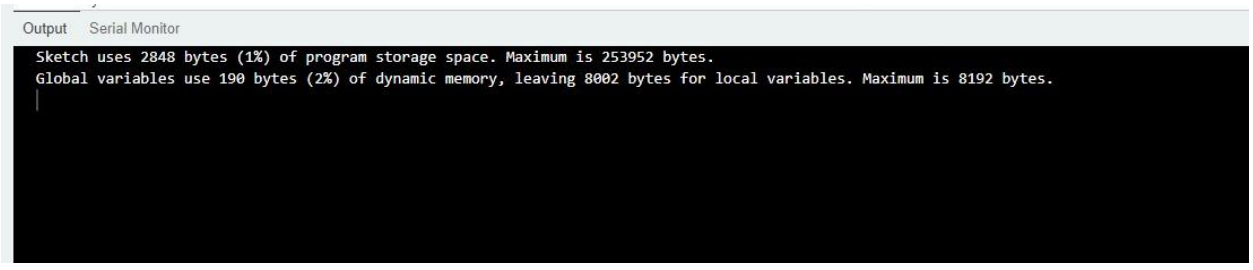
See right black part for measured temperatures

Temperature of pi after improvising code and removing the GUI part is reduced to 50.6 °C, as shown in the image below.



See top right black part for measured temperatures

After minimizing memory overheads, we decided to minimize computational overheads. Because raspberry pi has many limitations. One of which is that it cant be used 24/7. It will get damaged also after a few hours it will get heated up badly even by using heat sinks and consumes more power. Also as raspberry pi is a microprocessor, so whenever we start it, firstly operating system loads and other necessary modules , which definitely results in increased computational overheads. So we switched to a microcontroller- Arduino mega board. Microcontrollers have no operating system , so it doesn't require any loading of Operating system whenever the system starts, which directly results in reduced computational overheads. Also microcontrollers require less power, less memory which makes them efficient to use over microprocessors and it is very much customizable as per convenience . So we used an arduino-mega board to design a driver drowsiness detection system. We have utilized an IR sensor to detect the eye blink of the driver from which drowsiness is detected. After designing a full working prototype, we measured its memory consumption. We were amazed by seeing that it is utilizing only- 2848 bytes or 0.02848 MB, which can be seen in the image below.



Memory consumption of arduino model- 2848 bytes

There is a huge difference between memory consumption of the raspberry pi model and arduino model. The Raspberry pi model is consuming 426 Mb and the Arduino model is consuming 0.002848 Mb which runs a few lines of code continuously in a loop . So memory overheads are significantly reduced. Also we are not using picamera with arduino because we are using IR sensor. So the overhead of the camera reduces. Also arduino boards can be used 24/7, as there are no computational overheads in using arduino unlike raspberry pi. Also the arduino board does not get heated up easily unlike raspberry pi. Arduino have many pros because microcontrollers have simple software and hardware unlike raspberry pi which have complex architecture and needed Operating system.

COST ANALYSIS :

Raspberry pi model:
Raspberry pi 4 4GB:INR 5000 approx.
BUZZER : INR 30
Picamera2: INR 300
Total : INR 5330 approx
Arduino model:
Arduino mega board: INR 1500
BUZZER: INR 30
IR sensor with glasses:: INR 250
Total:INR 1780 approx
So above cost analysis, clearly shows that the arduino model outperforms raspberry pi model. So we successfully minimized memory and computational overheads of our driver drowsiness detection model.

CONCLUSION/FUTURE SCOPE:

We have developed a prototype which will help saving lakhs of lives. Our prototype is cost efficient, robust, efficient, free of memory and computational overheads and can be used 24/7. Because of robustness and 24/7 usage, our model can be easily used by workers working at mining sites. Several accidents happen at mining sites because of drowsiness of workers as they work for long hours in the tunnels. So if drowsiness is detected earlier in workers, then heavy duty vehicle accidents can be prevented and many worker's lives can be saved. We proved the cost efficiency of our arduino model to that of the raspberry pi model. As future work of this project, we will try to add components to our prototype which will send the car a signal that the driver is feeling drowsiness , if the driver is found drowsy or alcoholic. We will use an alcoholic sensor to detect alcoholism in drivers. At present the prototype needs to be **connected to power source** continuously ,it can be extended by **adding battery** and other components for making **real time Embedded system.**

Also the IR sensor does not work at night vision , so in place of this we can use a night vision IR sensor . Also arduino uno or arduino nano board can be used in place of arduino mega board because arduino nano board is very small board and has negligible weight so it will stick to glasses and is more efficient in terms of memory and computational overheads.

CODE LINK:https://drive.google.com/file/d/1N1NYrHbGWXC3wTvjLLY-FDpvFN6iOlG4/view

REFERENCES

1 . Rahul Atul Bhope, "Computer Vision based drowsiness detection for motorized vehicles with Web Push Notifications", IEEE 4th International Conference on Internet of Things, IEEE, Ghaziabad, India, 2019.

2 . Chris Schwarz, John Gaspar, Thomas Miller & Reza Yousefian, "The detection of drowsiness using a driver monitoring system" , in Journal of Traffic Injury Prevention (Taylor and Francis Online), 2019.

3 . Aditya Ranjan, Karan Vyas, Sujay Ghadge, Siddharth Patel, Suvarna Sanjay Pawar, "Driver Drowsiness Detection System Using Computer Vision.", in International Research Journal of Engineering and Technology(IRJET), 2020

4. B.Mohana, C.M.Sheela Rani, "Drowsiness Detection Based on Eye Closure and Yawning Detection", in International Research Journal of Engineering and Technology(IRJET), 2019.

5. T. Danisman, I.M. Bilasco, C. Djeraba and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," Universite Lille 1 & Telecom Lille 1, Marconi, France, 2010

6. Kyong Hee Lee, Whui Kim, Hyun Kyun Choi, Byung Tae Jan. “A Study on Feature Extraction Methods Used to Estimate a Driver’s Level of Drowsiness”, IEEE, February 2019
7. Tianyi Hong, Huabiao Qin, “Drivers Drowsiness Detection in Embedded System.”, IEEE, December 2007.
8. Hardeep Singh, J S Bhatia and Jasbir Kaur, “Eye Tracking based Driver Fatigue Monitoring and Warning System”, IEEE, January 2011.
9. Varsha E Dahiphale, Satyanarayana R, “A Real-Time Computer Vision System for Continuous Face Detection and Tracking”, IJCA, Volume 122 Number 18, July 2015.
10. SaeidFazli, Parisa Esfehani, “Tracking Eye State for Fatigue Detection”, ICACEE, November 2012. Gao Zhenhai, Le DinhDat, Hu Hongyu, Yu Ziwen, Wu Xinyu, “Driver Drowsiness Detection Based on Time Series Analysis of Steering Wheel Angular Velocity”, IEEE, January 2017.
11.https://www.ijset.in/wp-content/uploads/IJSET_V8_issue4_226.pdf
12 .Lestin Jills Joseph and Dr. Lokesha M, “Arduino based real time driver drowsiness detection and Alert System” september 2016
13.http://www.ijecs.in/index.php/ijecs/article/view/2537
14. Archana Jenis M.R M.E., Assistant Professor Soban Mathew. J “Driver Drowsiness and Alcohol Detection System Using Arduino”