Eye of Gatsby

Final Report - L20 - April 2022

Fima Kazmin
Harsheen Sodhi
Gurjas Bassi
Mohammed Abbas

Don't Sleep and Drive

Table of Contents

INTRODUCTION

- Motivation

When it comes to car accidents, there is not always a second chance. Drowsy driving is a prime example of such accidents. It is estimated that 21% of fatal car crashes involve a person driving while sleepy and studies say that 6,000 deadly crashes happen every year due to drowsy driving (7). Drowsy driving occurs due to sleep deprivation, late or long shifts at work, and driving fatigue from a long drive for the holidays. Another factor that will be accounted for in the system design is being distracted while driving as it contributes to 21% of fatal collisions every year (5). Driver impairment caused by sleepiness and visual distractions leads to critical accidents that need to be predicted or detected. Two members have personally experienced the disastrous and unfortunate effects of drowsy driving, while many more experience this tragedy worldwide. Driving to a conference in India early one morning, Harsheen's cousin fell asleep and hit a tree. He passed away on the spot. Another tragic accident was experienced by a very close friend of Mohammed's back in Oman. His friend was driving back from a long vacation trip without making a quick stop to rest though he was feeling a bit drowsy. Unfortunately, he fell asleep while driving and made his final stop. The terrible consequences and personal mark on most of the group members were the motivation to build a device that can detect drowsiness, alert drivers and subsequently prevent the occurrence of critical accidents and save the lives of our loved ones.

From an applied point of view, the driver's state should be observed with the help of unobtrusive safety systems. A camera-based system is an example that is convenient for driver applications. Information such as head tilt, eye direction, and eyelid movement for a certain number of seconds can be obtained to be used to detect drowsiness and alert drivers potentially.

- Background

Driving a car is the largest means of transportation since the beginning of the 20th century. It started in the 1880s with cars that ran on steam, then, by 1910, gasoline cars became more popular due to their size, power, and folding tops to keep drivers and passengers safe (8). Fast forward to nowadays with constantly updating technologies, there have been many new and upgraded safety features installed on vehicles. However, we are still unable to control most of the accidents that cost a lot of lives. Drowsy driving is a disturbingly common phenomenon that is majorly underestimated as it is a rather difficult symptom to measure or observe. Thus, there has long since been a need to find suitable indicators for the detection or prediction of such distracting states and to alert the driver before it is too late. There are a lot of studies in the automobile industry around designing feasible, low-cost driver safety support systems and some are focused on the selection of the most effective indicators and algorithms to detect or predict sleepy drivers. A literature review on sleep indicators and classifiers was conducted for evaluating the selection of drowsiness indicators to gather ideas on combining those indicators into classifiers for our approach. From the literature review, it is evident that these indicators are often grouped as vehicle-based, physiological, behavioral-based, or a mix of each. Vehicle-based indicators account for the driver's ability to operate the vehicle in an expected manner which includes measuring the rotation of the steering wheel, distance to surrounding vehicles, and the distance from the lane or lateral direction (lane departure warning system). Research shows that this approach achieves an accuracy of 86% in detecting drowsiness based on correlations between these measurements and drowsiness (9).

On the other hand, physiological indicators include measurements of brain activity and the development of electroencephalography (EEG) algorithms. This fatigue countermeasure technology incorporates data from physiological sensors attached to the person. EEG impulses provide information about brain function using four cues; delta (δ), theta (θ), alpha (α), and beta (β) signals to calculate the driver's drowsiness. The alpha (8-12 Hz) and theta (4-7 Hz), and delta (0-4 Hz) signals increase when the driver is getting drowsy. Whereas the increased beta content (12–30 Hz) is a sign of alertness. Delta and theta are considered slow-wave activities, whereas alpha and beta are fast-wave activities. The ratio between fast-wave and slow-wave activities such as (β/α) increases as fatigue progresses (2). Electrooculography (EOG) is another psychological indicator that measures longer blink durations and slow eye movements (10). In a paper, psychological sensors give better accuracy of more than 90% among all three methods (12). Although the psychological solution has very good accuracy, one major disadvantage is making the driver wear sensors to detect these signals, which may bother them. They are not feasible for commercial use because they are too obtrusive and non-invasive bio-signals detectors do not function or are impractical, and come with a big cost.

Our main goal is to capture these drowsiness indicators with the help of an unobstructed, reliable, and cost-efficient system. Therefore, non-invasive camera-based systems can provide several measures of drowsiness based on behaviors such as blinking and head tilting. Almost all recent literature on drowsiness detection emphasizes the fact that a combination of several different indicators is needed in order to be able to accurately classify the level of sleepiness. Researching different types of parameters will be used to find indicators for a driver not paying attention to the road including factors such as sleeping includes blink duration, PERCLOS, blink frequency, and facial lateral position variation. PERCLOS is the PERcentage of eyelid CLOSure over the pupil over time.

Schleicher et al. have investigated several EOG indicators and concluded that blink duration, delay of lid re-opening, blink interval (frequency), and lid closure speed are the best indicators (4). When sleepy, the pattern of fixations changed such that the proportions of very short, <150 ms, and very long, >900 ms, fixations increased. Bergasa et al. have investigated camera-based indicators and found that the most important measures are fixed gaze, PERCLOS, and blink duration (3).

Then, PERCLOS is calculated by the number of closed eyes frames over a period of time. When the eyelid covering the pupil is over 80% this would count as the eyes being closed to the system. This is a low-cost system however it does have its issues and other methods should be implemented to ensure a wider variety of situations are covered for when the driver’s eyes aren't all the way visible for example.

This solution of using PERCLOS is not a solution that works for every case possible. An example of this would be when the driver is wearing sunglasses and their eyes are not visible. Due to these limitations, a neural network such as a convolution neural network can be adopted into the project which would be used to analyze visual imagery without necessarily requiring visibility of the eyes. Deep neural networks have already become commonplace in facial recognition software (11).

The method which was used in the final project was the method of Eye Aspect Ratio(EAR) to detect drowsiness. Each eye is represented using 6 landmark points. An open eye would have more EAR while a closed eye would have far less EAR (6). This method along with Dlib’s face detection using facial landmarks would be used to determine when the eyes are open and closed far more accurately than the previously mentioned methods and would result in our device working more effectively.

In order to successfully complete this project, the team members must have a good understanding of neural networks and machine learning. Training the algorithm and using the neural network to combine the different working subsystems (deep learning) is an important segment of the system. Some working knowledge of image processing and optimization techniques would be helpful in building the project as identifying a drowsy or sleepy person is detected via images taken by the camera and further processed accordingly. This will require knowledge of Python which has open libraries such as OpenCV and Dlib that are useful for computer vision in image processing and in finding the frontal human face to estimate its pose.

PROJECT GOALS & PLANNED APPROACH

The aim of this project is to be able to properly identify if a driver is falling asleep and sound an alarm to wake them up if falling below a certain threshold of fatigue. A high-definition camera would be equipped with a microcontroller (Raspberry Pi) that has a speaker/buzzer attached. The camera must be precise enough to clearly see if the eyes are open or closed. A machine learning algorithm will be running on the board that reads visual cues from the driver and decides if it is necessary to sound an alarm. The machine learning algorithm will be trained before installation, with a focus on identifying the face and eyes of any driver, their eye openness state, their head tilt, their eye direction, and if they are wearing anything that blocks the direct vision of the eyes. The algorithm will return whether or not the driver is awake or drowsy. If the driver starts to appear fatigued, a warning will sound off on the speaker. If the driver appears to be entirely asleep, a harsher alarm will sound off the speaker.

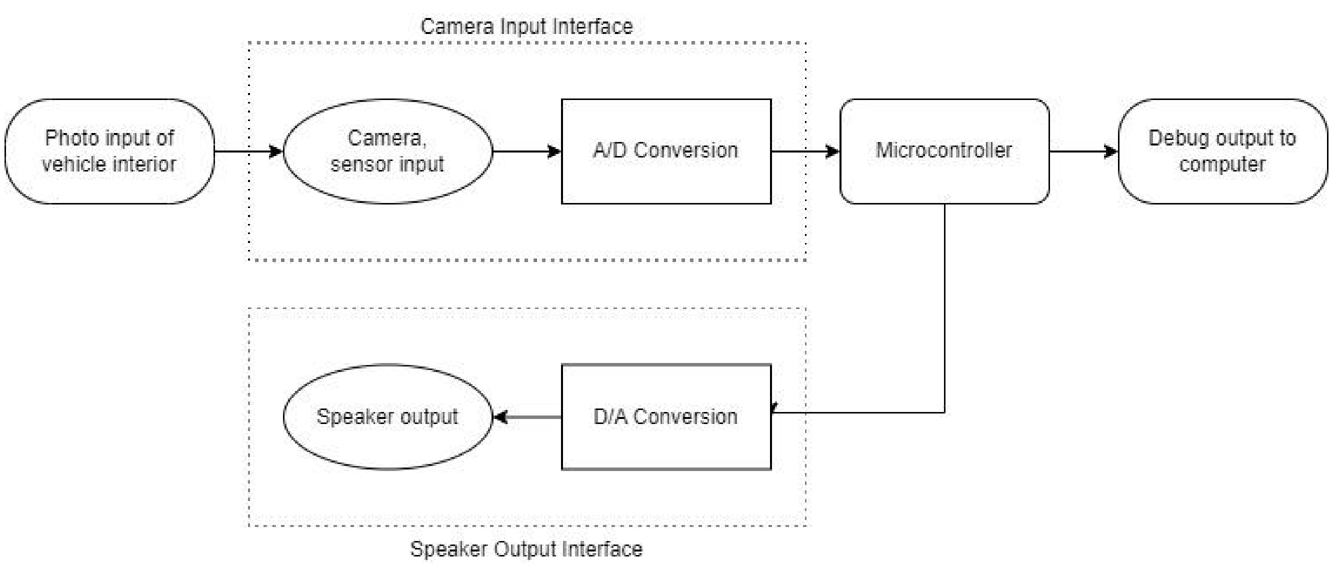
In implementing this project, a significant amount of groundwork had to be laid out. To be able to properly identify driver fatigue levels and impairment, we needed to train an algorithm that properly identifies a face, the eyes, and the level of eye openness. If we look to try to improve the algorithm further, it would help to classify head tilt levels, and eye direction, and identify if a driver is wearing sunglasses. To facilitate the development of the project, we have identified three levels of development, the bronze, silver, and gold levels.

The bronze level would consist of a working model that can properly identify whether or not the eyes are closed. The device at this stage would include all of its hardware components. Being able to identify a face and eyes, the machine learning algorithm then decides if the eyes are open or closed. With that information in hand, a decision can be made on whether the amount of time spent with closed eyes is unacceptable and to send warnings. At this stage, the algorithm can identify two stages: awake, and drowsy.

At the silver level, the device would be able to detect more attributes on a face, including head tilt level and eye direction. The algorithm would be able to classify into three stages: awake, drowsy, and asleep. The drowsy stage would now include distracted driving, where if the eye direction and head tilt levels are away from the road, the device would sound a warning to remind the driver to pay attention to the road.

At the gold level, the model would be able to identify driver drowsiness and distraction with the added feature of being able to perform even if the driver is wearing sunglasses. The algorithm would have to be able to identify if the driver is wearing sunglasses, then rely more on other sources of information aside from the eyes. The algorithm would focus primarily on head tilt levels, noting that if the driver's head tilt drops multiple times within a short time, they may be distracted or fatigued, and if the driver’s tilt drops for an extended period, they may be asleep. The algorithm would respond appropriately with either a shocker for people with hearing difficulties or a loud alarm.

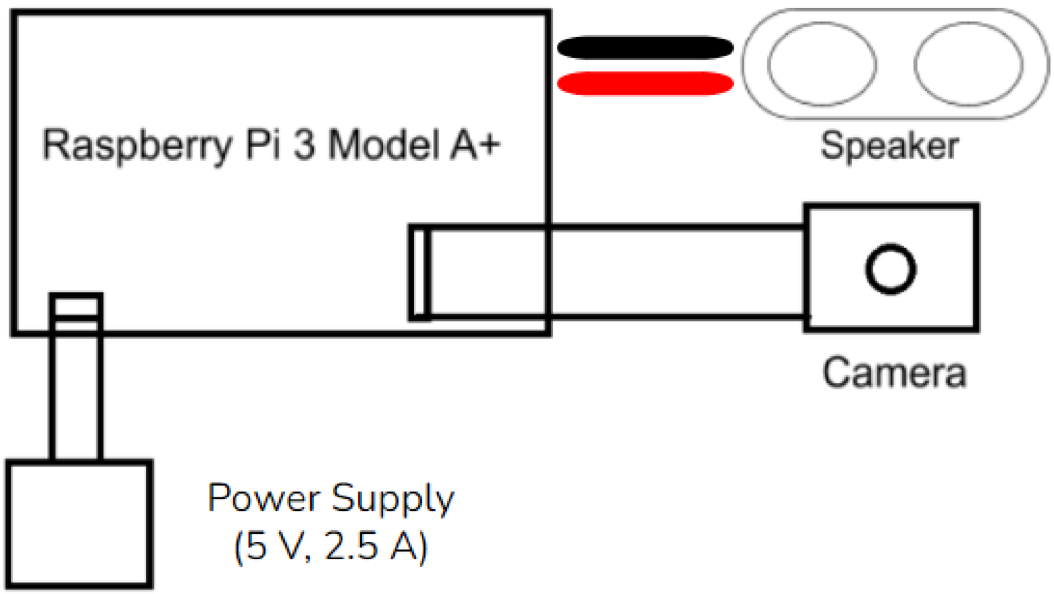
Below is the flow diagram that illustrates the stages of how the system works for detecting drowsiness from collecting real-life data and processing it to output a warning alarm.



ACTUAL IMPLEMENTATION

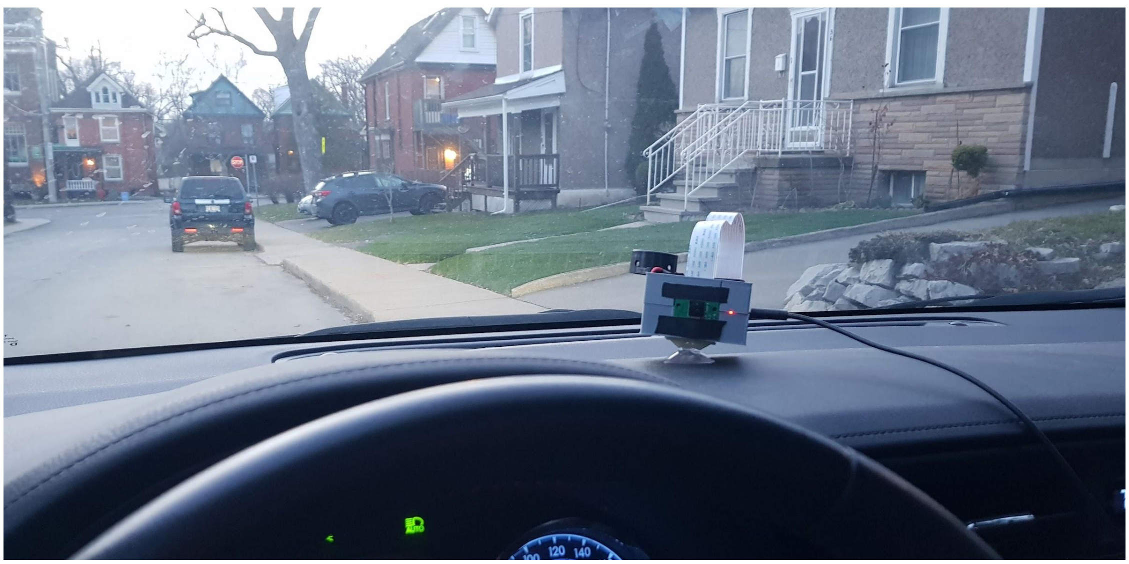
- Design

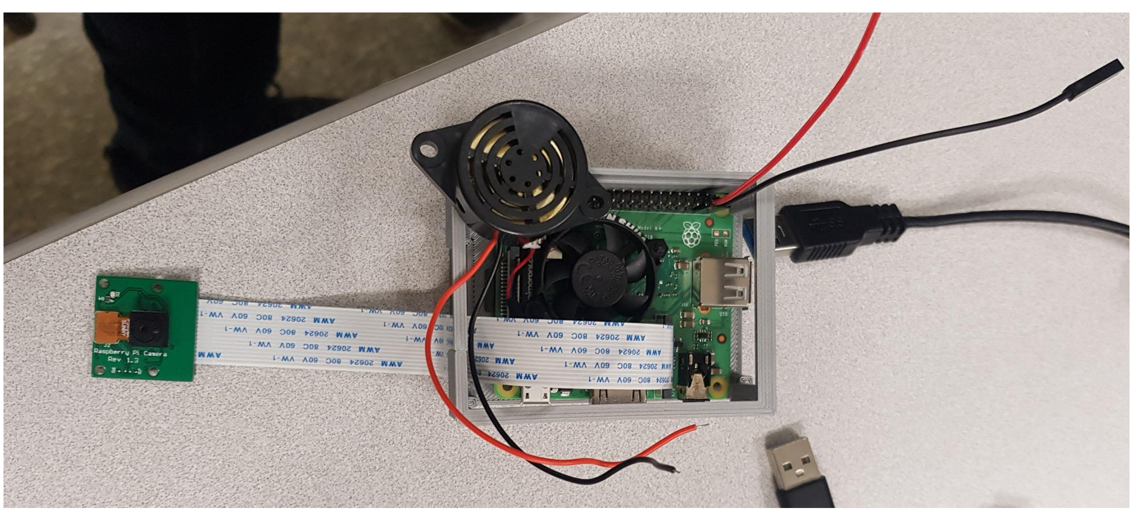
The initial features would be a high-definition camera to scan the face and a fast microcontroller to process the image. Both components are crucial to start processing and returning results. The buzzer would be the next feature to be able to warn the driver if they are fatigued. Although it would be possible to have LEDs to indicate the state, they are insufficient to grasp the driver's attention. Finally, the power source is supplied via USB on the infotainment system in almost every modern car.



Given the design requirements, it was decided to design our circuit around the [Raspberry Pi Model A+]() (1.4 GHz) as it would be able to process all the visual information much faster than an Arduino (16 MHz) at around the same cost. The A+ Raspberry Pi Model does not come with the Power-Over-Ethernet (POE) port like the B+ but it was not required in our design thus A+ was favored. The camera module used was [Pi Camera V 1.3]() with a 5-megapixel resolution and a 1080p; 30fps, or 720p; 60fps transfer rate connected directly to the Raspberry Pi board. The buzzer is wired via the General Purpose Input/Output (GPIO) interface pins that will be used to alert the driver and ensure an awake state when they are detected by the camera as drowsy while driving.

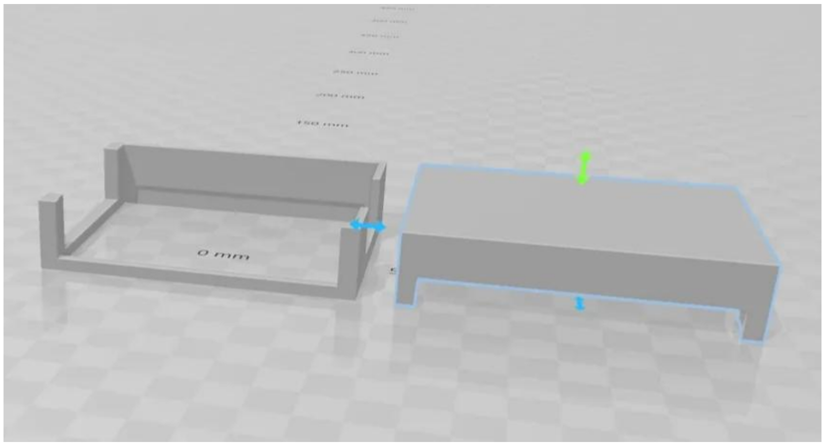
Our philosophy was to design the device to be portable and user-friendly. For example, considerations were made upon picking the best buzzer to accommodate individuals with hearing impairments or for driving in loud traffic. Thus, the buzzer chosen can go up to 100dB adjusted through the interface GPIO pins. Also, the design features the I/O ports of the Pi exposed to be easily accessible for troubleshooting and adding accessories such as a battery power supply and a cooling fan. All the components are housed together in a 3D printed chassis with suction cups attached to the bottom for practical use in a car placed securely on a dashboard.



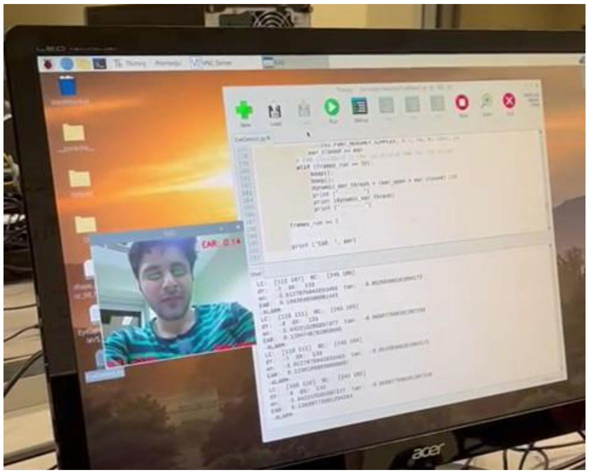


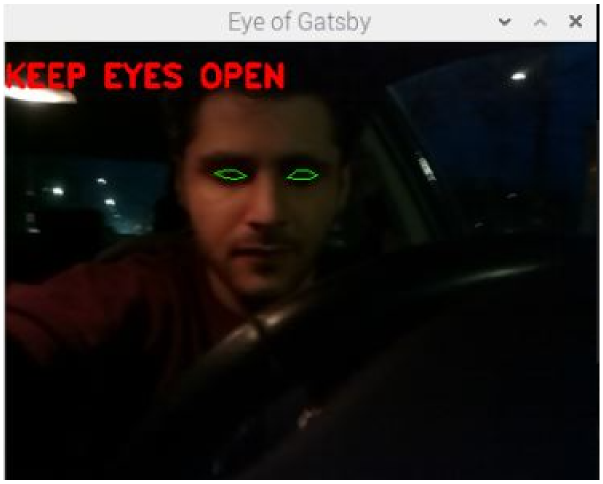
- Modeling/Simulation

Our code did not require simulation. The programming has no need for modeling or simulating aside from its immediate application. However, it required a lot of testing and the modeling was done for the physical aspect of the device which is the 3D printout of the device's chassis modeled on Fusion 360 as STL files that were used to 3D print.



Testing was done in different environments and lighting conditions that would allow for more information to train the machine learning algorithm such as at home, in the lab, and in the car day and night. Fatigue can happen at any time of the day and probably more likely at night, necessitating a system that can work even in low-light conditions. The chosen camera was proven to have an adequate resolution, and the program is able to image process under such conditions. Part of our testing was training the program to calibrate before first-time usage which helps eases eye-tracking.



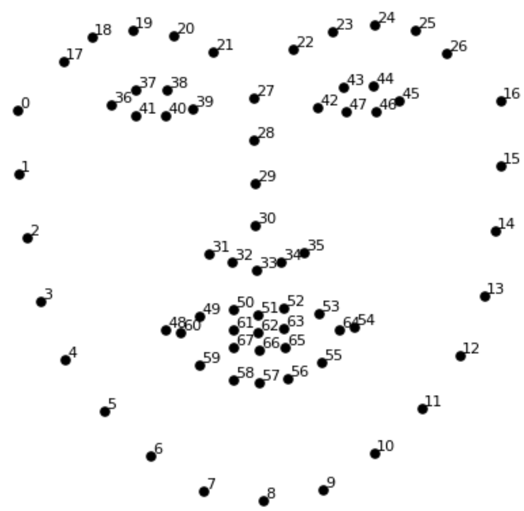


- Implementation

With the camera and speaker attached, the Raspberry Pi-powered, and an optional monitor attached to be able to see the output, we implemented our program to prevent accidents. The device accounted for differences in users' EAR (Eye Aspect Ratio) through calibration. A python script was used for this application that we configured to run on the start-up of the Raspberry Pi.

The device was designed to run the script at startup to allow the device to operate right when turned on, and additionally had the capacity to be operated with a GUI to show users how the device works by connecting it to a display. Once running, our device initially calibrates on a user's face and eyes before continually scanning for drowsiness.

Our coding process began with researching facial detection technologies and implementation with Raspberry Pis. We discovered various resources that could be applied, including OpenCV, Dlib, Haar Cascade, the 5 facial landmark model, the 68 facial landmark model, and some others (1). We quickly excluded many options as we found that they would not run fast enough on our limited hardware. As such, we decided on the 68 facial landmark model, using OpenCV and Dlib as that was our fastest implementation that contained enough accuracy to be able to detect the openness of the eyes while maintaining the fastest scan rate.



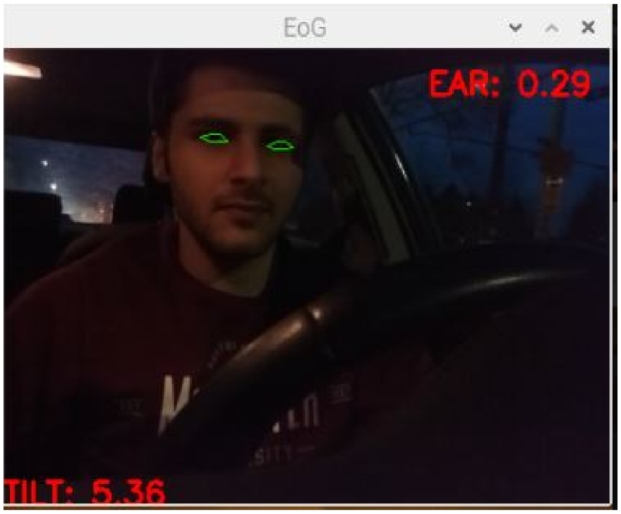
68 Facial Landmarks

Our code is structured with the main loop that handles the facial recognition and alarms, an initialization, and helper functions to help guide flow. In the initialization, we set the sensitivity of the alarms with appropriate timings, initialize the video stream and buzzer, and load the facial predictor. Within the main loop, we read in the image, reduce its size to save processing time, and scan for a face. Upon detecting a face, we first enter a calibration phase.

During the calibration phase, no alarm will go off. A beep informs the user that the calibration is starting, during which they must keep their eyes open. A second beep informs the user that the second phase of the calibration is starting, during which they must keep their eyes open. Two beeps follow, indicating that the calibration is complete and the device is ready to use. To reduce interference from human delays, the calibrations are delayed by 5 frames, then are run for 12 frames each for each eye state, the first 2 frames of which are ignored to account for the human reaction time delay. As such, it plays out as 6 frames to allow caching and device setup, one beep, 2 frames ignored, 10 frames to calibrate the eye open state, one beep, 2 frames ignored, 10 frames to calibrate the eye open state, and then two beeps to end the process.

The main loop, responsible for keeping the driver awake if they appear drowsy, follows the calibration phase. During this phase, the device reads in the EAR, and if the EAR is less than the average between the EAR with eyes opened and the EAR with eyes closed found during calibration, then the device adds to two counters for the amount spent with eyes closed. The first counter is reset if the user opens their eyes, while the second counter is only decremented. The first counter is when the user has their eyes closed consecutively, and the second counter is if the user is having trouble keeping their eyes open, though never keeping them closed long enough consecutively. If the time spent with the eyes closed passes either timing threshold set at startup, then the alarm will go off to alert the user awake. When the alarm goes off, the counters are both reset to allow the user to turn off the alarm once they open their eyes. Otherwise, it would be necessary to keep the eyes open for an extended period consecutively to shut down the alarm, which would be redundant and irritating.

The Eye of Gatsby also detects the rolling tilt of the face. If the face tilts too far to the left or right, then a counter is incremented. If the head is held too long at a deep angle, an alarm will be triggered. This counter is reset if the face is corrected. This implementation was intended to aid in the detection of drowsiness when eyes could not accurately be detected, such as in the case of sunglasses. While the tilt detection works, the facial detection when a user is wearing sunglasses was not accurate enough for this to be used effectively, as the face would usually occasionally not be detected at all if the eyes were not visible. If the eyes were visible, then the open or closed state of the eyes was a more useful indication.



The EAR is calculated by taking the distance from the top of the eye and dividing it by the width of the eye. These distances are found by taking the coordinates of the eye from the 68 facial landmarks. This provides an accurate reading of the state of the eye, regardless of angle or distance to the camera. The tilt is detected by taking the angle of the face through three points off the sides of the detected face. The average of the points is calculated, then using the arctan of the vector between them, the angle from the origin is found which corresponds to the angle of the face.

When counting the time spent with the eyes closed, the device counts the time since the last frame, which accounts for spikes or drops in framerate, removing the need for a steady framerate, allowing framerate to be a helpful addition rather than a strict requirement aside from a minimum. This feature prevents problems occurring from temperature, long-term running operations, and various other factors.

CRITICAL PROBLEMS SOLVED

1. Raspberry Pi

   Initially, the chosen microcontroller was the Arduino Uno for handling the imaging processing, however, it was found that using a Raspberry Pi would have been more efficient and more simple as it uses a fairly standard OPP language such as Python along with its clock speeds of 1.4 GHz compared to the Arduino 8 MHz. The Pi did however have an issue of overheating which was overcome by adding a Fan shim to the CPU to ensure it does not slow down or stop while operating. The Pi also had a low voltage warning which did not impact the performance of the device so was ignored, when the device was connected to the 5V USB port of a vehicle this warning did not appear. Another issue was operating the Pi which was solved by installing Raspberry Pi OS.

2. Slow Facial Detection/Lag

   When first testing and operating the device the Facial Detection was rather slow and laggy, this was fixed by both cooling the CPU of the Pi along with reducing the resolution of the camera which greatly improved the speed of the device but did not decrease the effectiveness of it to detect human faces. Once these solutions were put in place the device did not lag and was able to track a face effectively.

3. Working Together Virtually

Working through covid was difficult for a project like this, it was later found that using a program such as VNC Viewer allowed everyone to remotely connect to the Pi and work on it while communicating through Microsoft Teams. There were many issues with other team members connecting to the Pi, this was resolved by connecting one of our ISPs and getting port forwarding to work, this ensured everyone can access and work on the Pi. Another problem was having this work on McMaster's WiFi network due to it having different encryption which the Pi by default was not able to handle, this was solved by installing a new network manager which allowed us to work on the Pi from our laptops on campus such as in the ITB labs.

4. Finding a Physical/Mechanical Stimulus

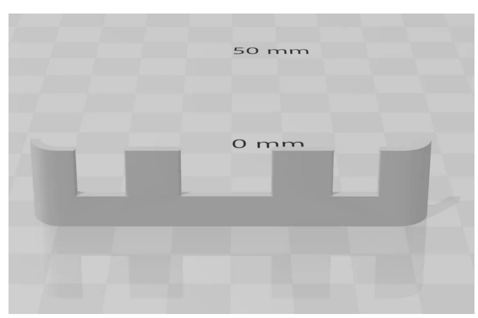
Another problem that we encountered was finding a strong audio stimulus to ensure that even in a noisy environment our speaker was heard. The buzzer we decided to use which functions and can be powered from the Pi's GPIO was a single 100 dB buzzer. The standard noise levels in traffic are generally from 50-90 dB, this buzzer ensures it can be heard even during the busiest of hours. Another issue was finding a physical stimulus, what was suggested and what would have worked would have been using some sort of shocker or a physical tool that would function through a solenoid that once which an electrical current would pass through would also alert the driver, the reason these ideas did not go through was due to time constraints and also safety for the driver.

FUTURE PLANS

The future plans for the device include better implementations and updates to the device. This includes allowing the device to render the frames faster from the microcontroller's camera, such as less OS overhead or a lowered resolution. Another future plan would include implementing better head tilt detection as our device currently only accounts for horizontal tilt. Future implementation could include vertical tilt for situations such as looking down at the phone while driving.

Making the device battery-powered and hopefully having it rechargeable to allow the device to operate even though it is not connected to a USB connection at all times. This could be used to make the device more accessible and usable in more environments than just a vehicle such as for cheap commercial use or for medical applications. This can also allow more voltage for the buzzer as it needs 12 VDC to reach 100dB that can be programmed for extreme cases of drowsiness or non-responsive reaction from the driver to stay awake.

Other ideas brought up would be to upgrade to a Raspberry PI Model 4 which would increase the processing power of the device and ensure the device does not lag while operating. Although our camera worked in most cases, sometimes it did have difficulty if the lighting was very poor, which is where another camera module such as the Raspberry PNight Vision camera module could be incorporated to improve the device to ensure the device functions in more environments.



Due to the nature of the project our chassis which was 3D printed was just a prototype in the future, this would also be improved to be more rigid while still retaining access to the Pi’s IO to debug the device. Along with improving the chassis, improving the mounting mechanism to the vehicle such as using an amount like most commercial GPS instead of suction cups could also be implemented.

CONCLUSION

All in all, we are very proud of our project overall and the deliverables we were able to show in practice. We were able to ensure the device was able to perform on both the bronze and silver levels while the gold level would have been possible but time constraints did not allow this to be feasible. Our device was able to detect when a driver is drowsy from their EAR ratios when the eyes are open or closed from its calibration. This worked from how many consecutive seconds the eyes were closed and raising a flag which then sent a signal from the GPIO to the buzzer to alert the driver. The device would also determine if the driver was drowsy via their horizontal head tilt from the driver's neutral position, if the driver was a certain number of degrees off this tilt again the buzzer would trigger. Due to time constraints we were unable to ensure the device would work through sunglasses, this could have been done by training the pre-trained model but due to the intensity of this task along with the time needed to get this done, this was not possible by the day of the EXPO.

REFERENCES

1- A. Rosebrock, “Facial landmarks with dlib, opencv, and python,” PyImageSearch, 03-Jul-2021. [Online]. Available: https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/. [Accessed: 18-Apr-2022].

2- Å. T, K. G, and K. A, “Spectral analysis of sleep electroencephalography in rotating three-shift work.,” Scandinavian Journal of Work, Environment & Health. [Online]. Available: https://www.sjweh.fi/article/1694. [Accessed: 18-Apr-2022].

3- Bergasa, L.M., Nuevo, J., Sotelo, M.A., Barea, R. and Lopez, M.E “Blinks and saccades as indicators of fatigue in sleepiness warnings: Looking tired?,” Researchgate.net. [Online]. Available: (PDF) Blinks and saccades as indicators of fatigue in sleepiness warnings: Looking tired? (researchgate.net). [Accessed: 19-Apr-2022].

4- D. Cazzoli, C. A. Antoniades, C. Kennard, T. Nyffeler, C. L. Bassetti, and R. M. Müri, “Eye movements discriminate fatigue due to chronotypical factors and time spent on task – a double dissociation,” PLOS ONE. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0087146. [Accessed: 19-Apr-2022].

5- “Distracted driving,” CAA National, 26-Feb-2021. [Online]. Available: https://www.caa.ca/driving-safely/distracted-driving/. [Accessed: 20-Apr-2022].

6- D. Pandey, “Eye aspect ratio(ear) and drowsiness detector using dlib,” Medium, 21-Apr-2021. [Online]. Available: https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706. [Accessed: 20-Apr-2022].

7- “Drowsy driving: Dangers and how to avoid it,” Sleep Foundation, 24-Jun-2021. [Online]. Available: https://www.sleepfoundation.org/drowsy-driving. [Accessed: 17-Apr-2022].

8- “Early cars: Fact sheet for children,” Smithsonian Institution. [Online]. Available: https://www.si.edu/spotlight/early-cars. [Accessed: 19-Apr-2022].

9- J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, “Steering wheel behavior based estimation of fatigue,” Driving Assessment Conference, 23-Jun-2009. [Online]. Available: https://pubs.lib.uiowa.edu/driving/article/id/28097/. [Accessed: 20-Apr-2022].

10- M. B. Kurt, N. Sezgin, M. Akin, G. Kirbas, and M. Bayram, “The ann-based computing of Drowsy Level,” Expert Systems with Applications, 04-Mar-2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417408000675#section-cited-by. [Accessed: 20-Apr-2022].

11- Sun, Yi, et al. “Deepid3: Face Recognition with Very Deep Neural Networks.” ArXiv.org, 3 Feb. 2015, https://arxiv.org/pdf/1502.00873.pdf. [Accessed: 16-Apr-2022]

12- Z. Mardi, S. N. M. Ashtiani, and M. Mikaili, “[PDF] EEG-based drowsiness detection for safe driving using chaotic features and statistical tests: Semantic scholar,” undefined, 01-Jan-1970. [Online]. Available: https://www.semanticscholar.org/paper/EEG-based-Drowsiness-Detection-for-Safe-Driving-and-Mardi-Ashtiani/4ec0886bb1e5b727fe381ee391fe29830ce627aa. [Accessed: 20-Apr-2022].