Method Article

Design, Simulation, Building, and Testing of a Microcontroller-Based AutomaticDrowsiness Detectionand Alert System

ABSTRACT

Aims: The premierpracticalaspect of this researchdriveis to propose,build, simulate, and evaluate anArduino-builtautomatic vehicle driver drowsiness detection and alert system.

Study design: The pivotal role of a reliable braking system in vehicular safety cannot be overstated, particularly considering the escalating frequency of traffic accidents, notably prevalent in Indonesia where human factors take center stage as the primary accident catalyst. An insightful poll underscores physical fatigue or drowsiness while driving as the foremost concern. Acknowledging the nuanced disparities between conventional air systems and electric systems, this research strategically directs its attention toward crafting new system characteristics that mirror the efficiency of the existing systems.

Place and Period of Study: The research actionengaged by the authors in a bunchof two students under the command of a professor as a part of one of hiscoursecapstone projectsfor the Bachelor of Science in Electrical and Electronic Engineering degree at the American International University Bangladesh (AIUB), Dhaka, Bangladesh. The authors performed theirinvestigativetasks at AIUB from June 2023 to January 2024.

Methodology: Recognizing the imperative to fortify vehicular safety, the integration of artificial intelligence emerges as a vital solution to assist drivers in ensuring the safety of individuals both within and outside the vehicle. This focused study, tailored specifically for Electric Vehicles (EVs), centers on the innovative application of object and distance identification methodologies to provide a comprehensive braking action indicator. Leveraging a minicomputer and a sophisticated neural network approach, images captured by a stereo camera undergo meticulous machine learning processes. This facilitates the precise categorization and measurement of distances between objects, with subsequent priority decisions determining the optimal degree of braking action.Employing an intelligent methodology, specifically fuzzy logic, the study demonstrates a successful outcome by constructing a curve while concurrently enhancing the dynamics of the pre-existing system.Moreover, a finely tuned Pulse Width Modulation (PWM) signal for braking, lasting 10 milliseconds, is intricately devised based on the study's discerning results.

Results: The system is simulated in Proteus and tested in real time to check its functionality. The outcomes of the test were very satisfactory.

Conclusion: This comprehensive approach ensures the seamless replacement of components while concurrently elevating the overall performance of the braking system.

Keywords: Braking, Drowsiness,Microcontroller, Raspberry Pi, Arduino Uno, Alert, Vehicle, Pulse Width Modulation.

1. INTRODUCTION

Over the years, a vehicle that moves people from one location to another has undergone fast development. According to data on commercial four-wheel vehicles, nearly 80 million units were sold in 2018 [1]. Naturally, the larger number of owned vehicles has an impact. Accidents are just one of the drawbacks of vehicle use. More than 1.3 million individuals per year pass away because of an automobile accident, according to the World Health Organization [2]. According to data on highway accidents in Jakarta, most accidents are caused by people who are fatigued or sleepy [3]. To avoid this issue, a driving assistance gadget is required. Additionally, this sophisticated driver assistance for braking can be built upon to create an autonomous car. There are various types of commercial vehicle braking systems. The one that is usually used is friction. Hydraulic actuators are typically used in lightweight automobiles and are used to mechanically create friction action when the brake pedal is depressed. Use pneumatic actuators for heavy-duty vehicles as well. This system is an experiment in moving the braking actuator with an electrical signal. Artificial Intelligence (AI)-based advanced vehicles have been the subject of research. Martin has created computational perception utilizing a visual and laser scanner [4]. Wang has developed a braking mechanism employing support vector machines [5]. The goal of Heimberger's development of a three-dimensional reconstruction by vision for auto parking was to create a low-cost intelligent system for vehicles [6]. Applying an air actuator technology to electric buses has disadvantages. The capacity of the energy storage would be reduced, and the energy consumption of the vehicle would increase first. Additionally, many conversion processes result in losses [4]. As a result, studies of an electrical brake actuator as a substitute system. Fast trains, magnetic hoist cranes, wheelchairs, and aerospace vehicles are only a few examples of brake assemblies for which this approach has been used [6-7].

Pneumatic systems, on the other hand, have compressible qualities as opposed to conventional electric actuators, which might result in rapid movement during braking action. Considering driving safety, it is undesirable. Therefore, the service braking component of the braking actuator is the sole subject of this study. The objective of this study is to build a process or algorithm that mimics the behavior of a normal pneumatic system utilizing an electric actuator as an alternate system. Therefore, the goals are to integrate artificial intelligence technology and gain the proper control for smooth braking. Additionally, using the technique to produce PWM signals using the Arduino Uno microcontroller as the controlled magnetic solenoid's input [8]. In this study, we suggested the Autonomous Emergency Braking System (AEBS), a low-cost braking system. When the human driver is not prepared to apply the brakes, the AEBS takes control of the vehicle. Luxury cars typically have this active braking safety feature installed. In this work, the created AEBS uses a DC motor as an actuator to pull the brake pedal and an ultrasonic sensing device to gauge the gap between the electric vehicle and the obstacle [9]. The technology will provide a warning stage of alarm sounds and LCDsbefore autonomous braking operation. In this study, modified on-off control is used to control the AEBS. The electric car prototype will have the proposed AEBS installed for low-speed automated braking.

2. LITERATURE REVIEW

A system that uses computer vision techniques to detect driver fatigue and drowsiness. The system processes images captured by a webcam to analyze eye movements and blink rates and generates an alert if the driver is found to be drowsy. The system can be implemented as a mobile application or integrated with a car for automatic speed control. The software requirements include Python, OpenCV, NumPy, SciPy, and Pygmy. The future scope of the system includes the use of artificial intelligence algorithms to improve processing speed and

accuracy. The system aims to improve driver safety by addressing driver fatigue as a significant factor in road accidents [1].

A system that uses image processing techniques to detect drowsiness in drivers. The system analyzes the driver's eye movements and facial expressions captured by a webcam to detect signs of drowsiness. If the eyes of the driver remainshut for a specifictime, the system triggers an audible alarm to wake up the driver. If the driver does not wake up, the system sends text messages and emails to their family members. The system uses OpenCV and Dlib libraries for real-time image processing and facial feature detection and incorporates the concept of the Eye Aspect Ratio (EAR) to determine drowsiness[2]. The journal discusses various methods and techniques for driver sleepinessrecognition utilizing machine learning and computer vision. It highlights the application of machine learning methods, such as deep learning and Convolutional Neural Networks (CNN), to analyze facial features, eye movements, and head positions to determine the drowsiness level of a driver. Several studies have been conducted to assess the effectiveness of these methods, with one study using a CNN-based spatiotemporal approach to check the alertness of a driver in real time, showing that temporal information significantly improved classification accuracy. Other studies focused on detecting distracted driving, such as cell phone usage while driving, using CNN architectures and data augmentation techniques to accurately classify distracted drivers[3]. The paper proposes a system that aims to detect drowsiness in drivers and monitor their health status in real-time. The system utilizes various sensors, including an eye blink sensor, a heart rate sensor, and a temperature sensor, to detect driver drowsiness and monitor their health. An Arduino microcontroller is employed to process sensor data and control system actions, integrating GPS and GSM modules to transmit emergency alerts to designated contacts. The system also integrates methods based on image processing, head tilting frequency, face recognition, and eye blinking analysis. The authors identify the need for a system that can detect drowsiness and notify drivers promptly, preventing accidents caused by this issue[4].

The authors present a widespreadalertingmethod for intelligent cars to avoid accidents triggered by drowsy drivers. The system utilizes video stream processing, facial recognition algorithms, and various sensors to detect driver drowsiness and collisions. It incorporates IoT, GSM, and GPS modules,cloud servers, Arduino Uno, Raspberry Pi, sensors, blink count, and image processing techniques. These are used to make the system more compact and user-friendly [5, 10-12]. The system alerts the driver through a buzzer, sends alert messages to family members or authorities via SMS, and displays information on an LCD panel [11]. The severity of the collision and its location are also tracked and communicated. The system aims to provide timely warnings and alerts to prevent accidents and ensure driver safety.In contrast to the papers, this paper aims to improve upon the concept of AI-based driver sleep detection by showing Proteus-based simulation [13] and adding functionalities such as auto-braking portrayed via a servo motor, car seat, and steering wheel vibration to wake up the driver portrayed using a vibration motor, vehicle speed recording all in an affordable cost.

3. WORKING METHODOLOGY AND MODELING

To accomplish the goal of detecting a driver's face and eyes and subsequently breaking and stopping the car when the driver falls asleep, OpenCV was used.OpenCV is an open-source library dealing with computer vision and machine learning to provide a common frame for such applications for accelerating the development of commercial products using machine learning [6]. Functions from the OpenCV library were programmed into Raspberry Pi to obtain face and eye information and perform functions of stopping or starting the gear motor, manipulating the servo motor, activating the vibration motor,and turning LEDs on and off.

The speed and RPM of the gear motor were then measured using an IR obstacle sensor and displayed on an LCD screen using an Arduino Uno R3.

Based on the LEDs, there are 3 situations. When the yellow LED is on, the face is detected that means there is a driver in the seat. While the yellow LED is on, if the green LED is on indicating eye detection, it means the driver is in the seat and awake, and the wheel is rotating symbolizing the car being driven. The servo motor is in an upright position indicating an unpressed brake pedal of a car. The LCD monitor displays the RPM of the wheel as well as the speed of the vehicle.If the yellow LED is on and the green LED is off, the red LED turns on and indicates an emergency where the driver is in the seat but asleep due to no eyes being detected. The vibration motor installed in the car's seat and steering wheel turns on and the gear motors turn off to bring the car to rest, the LCD screen updates accordingly.

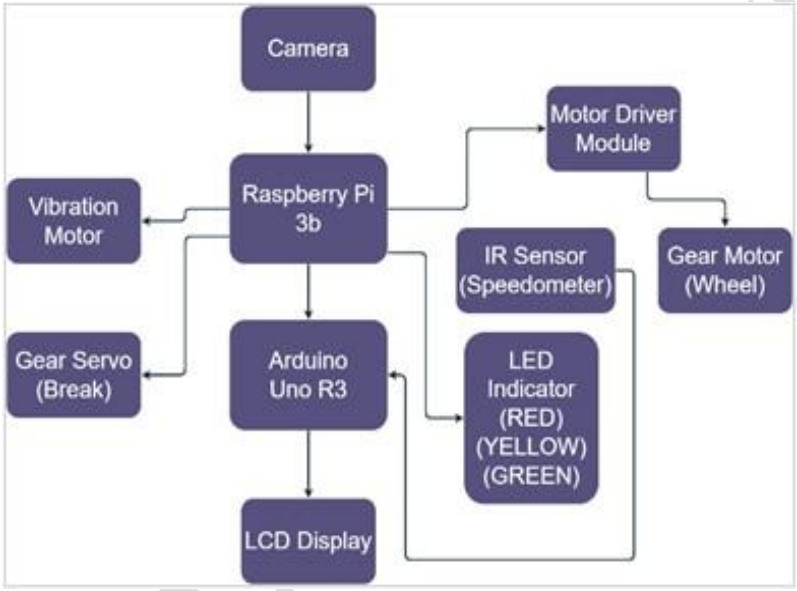

Fig. 1.Block diagram of the proposed driver drowsiness detection system.

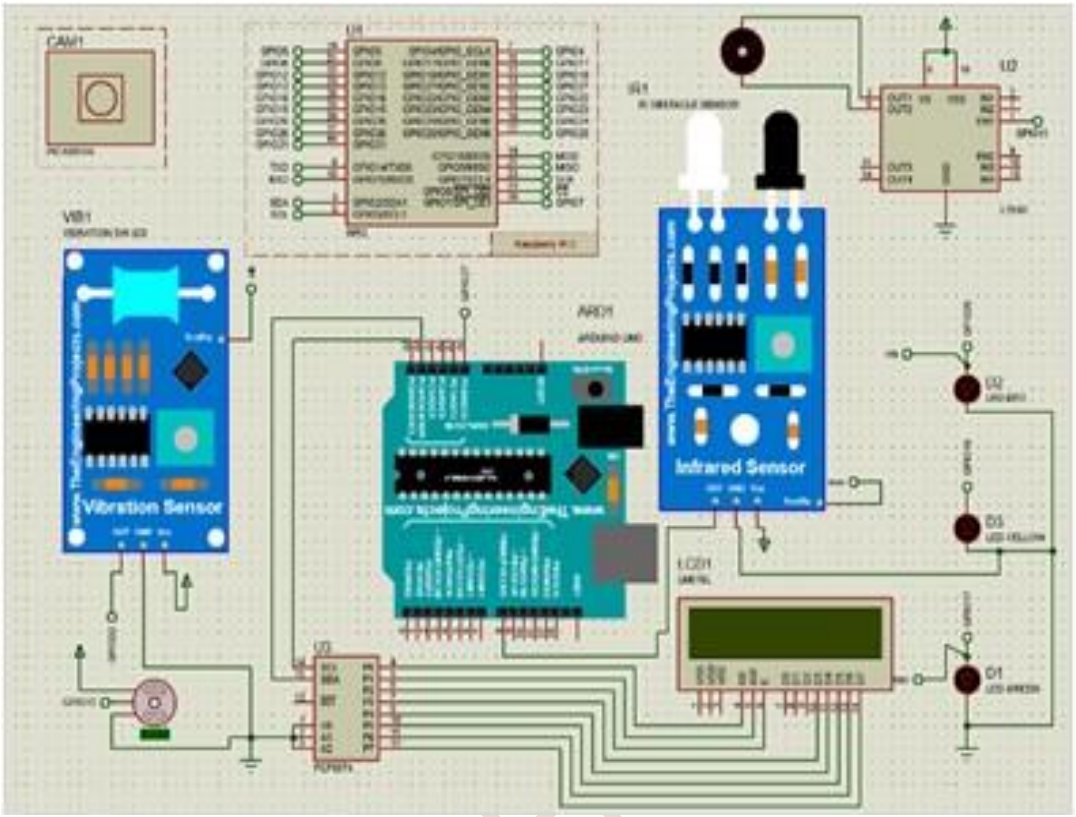

Fig.2.Schematic diagram of the proposed system drawn using Proteus.

4. PROGRAM DEVELOPMENT

The program was developed using Raspberry Pi and Arduino Uno. In the next two sub-sections, the program development processes using these two boardsare described.

4.1 Program Development for Raspberry Pi

First, Pythonprogramming requires various appropriate libraries, such as numpy, cv2, os, and RPi.GPIO, time, etc. are imported into the main.py file in the Raspberry Pi. Here, numpy is employed to work with arrays[14]. cv2, a module import name for OpenCV Python, is a cross-platform library that canresolve all computer vision-related problems. cv2 provides essential functions, such as creating color masks, resizing, and rotating images or videos.os module in Python provides functions for interacting with the operating system [15]. RPi.GPIOmeans the 40 input/output pins on the Raspberry Pi board for connectingthe external devices and hence sending or receivinginput/output signals.These pins are called General Purpose Input/Output (GPIO) because they can be programmed for different functions. The ‘time’elementpermits working with various time instances in Python so that we can get the current time, pause the program during its execution, etc. [16].

Next, pins were allocated for specific functions and appropriate modes were set using various functions, like GPIO.setmode(), GPIO.setwarnings(), GPIO.setup(), etc. After that, the servo motor was set to start operation at 50Hz frequency with a duty cycle of 5[16]. This was done to emulate the realistic situation of a car brake, which is constantly pushed and released using the function GPIO.PWM().

The functioncv2.CascadeClassifier()is used to makehaarcascade_frontalface_default.xml and haarcascade_eye.xml pre-trainedclassifiers available in OpenCV used for face and eye recognition. There are four different stages under which this recognition procedure occurs. For example, Haar-feature selection,Integral Image creation, AdaBoost Training, and Cascade classifier. Using a Haar-like feature,some effectivefeaturesrequired for recognizing an objectwere extracted. To reduce the required time substantially for implementing this process, an Integral Image creation technique was used in the second stage. To obtain the best features among the different extracted effective features, the AdaBoost Training was used by combining several weak classifiers into one strong classifier linearly. To boost the computation speed significantly and improve the computation process efficiency, the Cascade classifier is used as a fourth stage in this procedure by combining more complex classifiers [17].

After that,the video was streamed into the Raspberry Pi using the camera and the cv2 function, cv2.VideoCapture(0). By changing the parameter value of this function from 0 to other values, potentially multiple cameras can be used, each assigned to a specific number. Here, 0 refers to the connected Raspberry Pi camera. Then in a while loop, frames captured by the cameraare read into the program using the cap.read() function and are saved in a variable. This variable value and cv2.COLOR_BGR2GRAY parameters are used in the cv2.cvtColor() function to convert a color image into a greyscale image and saveit in another variable.This second variable is used in face_cascade.detectMultiScale() function along with the coordinates of the boundary of the rectangle (x,y,w,h) for detected faces.The method accepts parameters, scaleFactor which determines the size of the rectangle.The higher the scaleFactor, the fewer the number of steps, as such the detection becomes faster, but objects may be missing. By default, this value is 1.3. The higher the values of the “minNeighbors”, the fewer the number of false positives, therefore, the less error in face detection; here it was set to 5[17].

In the beginning of the process, the yellow LED,which signified the face detection,is set to LOW as the face has not been detected yet. When a face is detected, a blue rectangle of 2mm thickness is drawn aroundthe periphery of that detected face by employing the rectangle method of the cv2 sectionthrough an iteration processusing afor loop across all the detected faces[17]. The rectangle method takes the boundary values returned by the face detection method. Here, the color and thickness of the rectangle are also set. The region of interest (ROI) within the rectangle was extracted in both grayscale and color using a cv2.rectangle() function. Since the face is detected, the yellow LED is turned on now, and it indicates thata driver is sitting on the seat.

Like the face detection method, the eye detection method was used to detect eyes within the grayscale ROI by using the eye_cascade.detectMultiScale() function. Since no eyes are detected at this point, the vibrator is turned on, Arduino receives no signal, the green LED is turned off, the wheel motor is not spinning, and the servo assumes the breaking position only the red LED remains on indicating an emergency by using several GPIO.output() functions. Besides, the servo duty cycle was changed to 5% only indicating the absence of the car braking system by utilizing a pwmChangeDutyCycle() function [17].

Like the face detection method, with the boundary values returned to draw rectangles around detected eyes, green rectangles of 2mm thickness were drawn around the detected eyes by using a cv2.rectangle() function. Later, the vibrator was turned off, the Arduino received a signal,the green LED and the wheel motor were turned on, and the red LED was turned off by using several GPIO.output() functions.Besides, the servo duty cycle was

changed to 10% indicating the absence of braking by using a pwmChangeDutyCycle() function [17].

The resulting image with the rectangles drawn around the detected faces and eyes was displayed in a window using the cv2.imshow() function. The code waited for 30ms for a key press using the cv2.waitKey(). If the ‘ESC’ key was pressed, the loop was exited.

After exiting the loop, some cleanup operations were performed, such as the GPIO pins being cleaned up using theGPIO.cleanup() function, the PWM object was stopped using thepwm.stop() function, and the video capture item was released using the cap.release() function. Finally, all OpenCV windows were destroyed using the cv2.destroyAllWindows() function. The complete flow chart of this program is shown in Fig. 3.

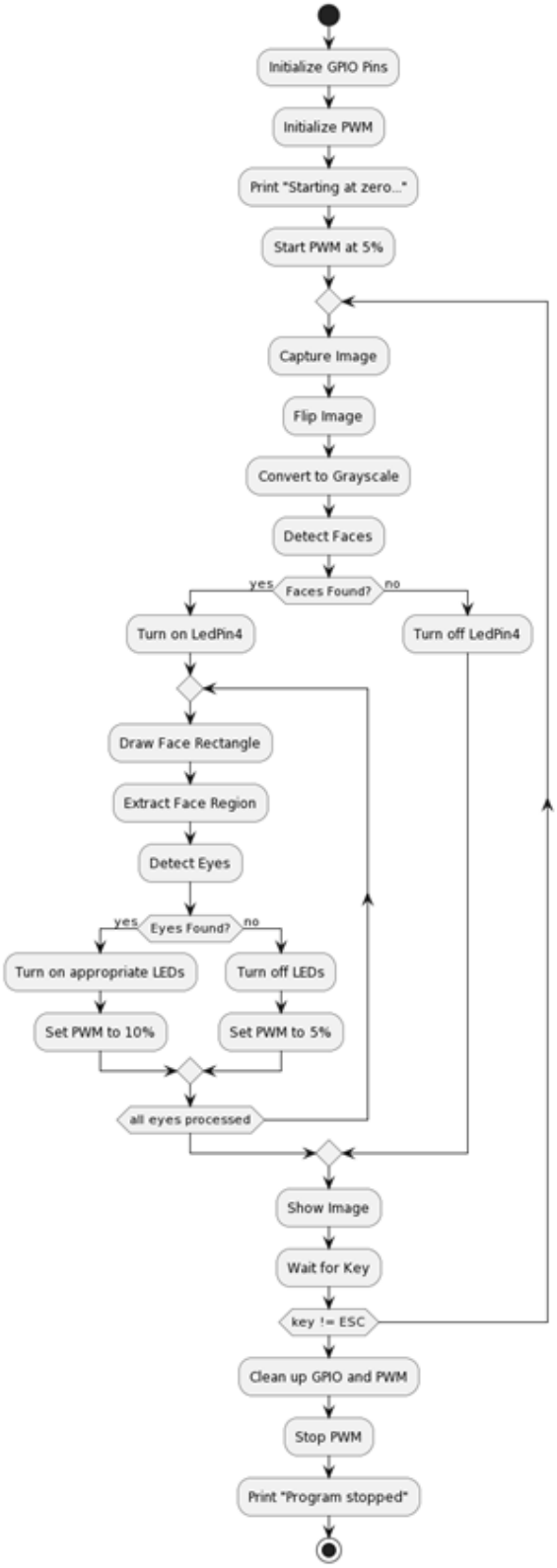

Fig.3.Flow chart of the Raspberry Pi program for the proposed system.

4.2Program Development forArduino Uno

At first, to develop the program using an Arduino Uno development board, the header file required for the LCD module is included using a #include<> function. Here, we used the Inter-Integrated Circuit or I2C protocol for the LCD to communicate with the Arduino Uno. The header file LiquidCrystal_I2C has a library function called an lcd() function to control the LCD. An address of 0x3Fis used for theLCD as an instance of LiquidCrystal_I2Cclass and the object has 16 columns and 2 rows. These are inserted as the parameters of the lcd() function.

After that, a few pins and variables were initialized with some values required for this program development. We used a Hall effect sensor that was connected to pin 8, the sample time was set to 1000, indicating that the speed of the wheel was measured over a period of 1000 milliseconds. The constant,maxspeedvariable was set to 10000, indicating the maximum speed of the wheel.The variable volume pin was set to A0, indicating that the signal from the Raspberry Pi controlling the PWM signal'sduty cycle was connected to the analog input pin, A0. The variable,read_ADC was used to store the ADC reading from the Raspberry Pi. The variable pwm_ pin was set to 11, indicating that the PWM output is available on pin 11. There are twovariables,duty_cycle and duty_ cycle _lcd that were used to store the PWM signal'sduty cycle in different contexts along the code.

The setup() function initialized the input and output pins, as well as the LCD. The Hall effect sensor, potentiometer, and PWM output were configured as inputs and outputs, respectively. The LCD was initialized, its backlight was turned on, and its size was set to 16 columns and 2 rows to display a total of 32 characters with 16 characters in each row. A welcome message was displayed on the LCD for 2000 milliseconds before being cleared.

The getSpeed() function measured the speed of a wheel using a Hall effect sensor. The function sampled the state of the Hall effect sensor for a period specified by the constant,sampleTime (in this case, 1000 milliseconds). The variable count was used to count the number of times the Hall effect sensor was triggered throughout this period. The variable,kflag was used to keep track of whether the Hall effect sensor had been triggered since the last time it was reset.

The function used a while loop to sample the state of the Hall effect sensor for the specified period. The loop checked whether the current time was less than or equal to the sampleTime, and if so, it continued to sample the state of the Hall effect sensor. If the state of the Hall effect sensor was HIGH, indicating that a magnet was near, kflag was set to HIGH. If the state of the Hall effect sensor was LOW and kflag was HIGH, indicating that a magnet had just passed by, the count was incremented and kflag was reset to LOW.

After the sampling period had elapsed, the function calculated the speed of the wheel in km/h was computed using the formula of equation (1) and was converted to an integer number, which is stored in a variable, named count2speed.

$$count2speed = \frac{30000}{sampleTime} \times count \qquad (1)$$

Equation (1) converts the number of times the Hall effect sensor was triggered during the sampling period into a speed in km/h. The calculated speed was then returned by the function.The loop() function is the main function of the Arduino program and executes repeatedly. It measures the speed of the wheel using the getSpeed() function, reads the value of the potentiometer connected to volume_pin using the analogRead() function, and maps this value to a duty cycle for the PWM signal using the map() function. The duty cycle

was then sent to the output pwm_pin using either analogWrite() or digitalWrite() function, depending on whether the duty cycle was greater than 0 or not.

The function also updated the LCD screen data with information about the current speed of the wheel and the duty cycle of the PWM signal. It sets the cursor position on the LCD screen, printed text, and values, and uses formatting characters such as % and km/h to display units. The function ended with a short delay of 10 milliseconds before repeating.

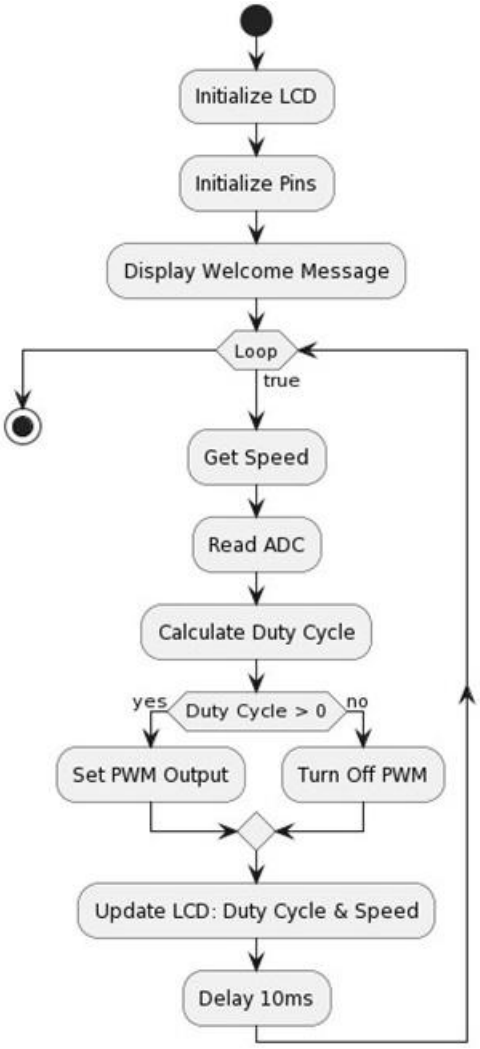

Fig.4.Flow chart of the assembly language program for Arduino Uno.

5. HARDWARE DESCRIPTION

5.1 Raspberry Pi 3B

In this task, we used a Raspberry Pi 3B single-board computer with a quad-core ARM Cortex-A53 processor, 4GB of RAM, Wi-Fi, Bluetooth connectivity, and various ports for peripherals. It wasselected for our workbecause of its small size and low power consumption.

5.2 Arduino Uno R3
In this work, an Arduino Uno R3 microcontroller board is used that has an ATmega328P chip with 14 digital input/output pins, 6 analog input/output pins, a USB connection, a power jack, and a reset button. We wrote a program for this board in Arduino IDE.

5.3 Raspberry Pi Camera Module
In this work, a Raspberry Pi Camera Module with the Raspberry Pi 3B computerwas used. It has a 5 MP sensor and can capture still images and high-definition videos. The camera module is connected to the Raspberry Pi 3B board using a ribbon-type cable.

5.4 16x2 Serial LCD Module
In this work, we utilized one LCD I2C-type display unit to display several types of messages.This unit is connected to the Arduino Uno R3 board using several wires utilizing the I2C serial communication protocol.

5.5 MG90s Micro Metal Servo
In this task, we used a Gear Servo Mini, a specialized servo motor to rotate it accurately to a predetermined angle in response to input signals. Because of its built-in potentiometer, this can provide positional feedback and thusenable precise alignment of mechanical parts. As a result, we can make sure that the motor's angle of rotation aligns with the desired angle.

5.6 DC Vibration Motor Module
In this work, we used a DC Vibration Motor Module that generates vibrations when powered by the DC power supply. It can generate alert signals for the system.

5.7 L293D Motor Driver Module
In this task, we employed an electronic motor driver module, L293D to controlthe DC motor with precise speed and correct direction of rotation.

5.8 DC Gear Motor and Wheel
In this work, a DC motor with a gearbox and wheel having increased torque and reduced speedwas employed. Its wheel rotates around the central axle. It facilitates movement by providing traction and support for the whole system.

5.9 LED Traffic Light Module
In this task, an LED traffic light signal module system consisting of a set of LED lights arranged in a specific pattern was utilized. The module includes red, yellow, and green lights as a prototype of the traffic light signaling system.

5.10 IR Sensor Module
In this task, an electronic infrared (IR) Sensor Module was used for identifying objects and sensing motion.An IR emitter releases infrared radiation. An IR receiver detects and transforms reflected infrared radiation into an electrical signal for the microcontroller board.

After combining all these modules using jumpers, cables, and wires, the whole hardware system is developed. This complete setup is shown in Fig. 5.

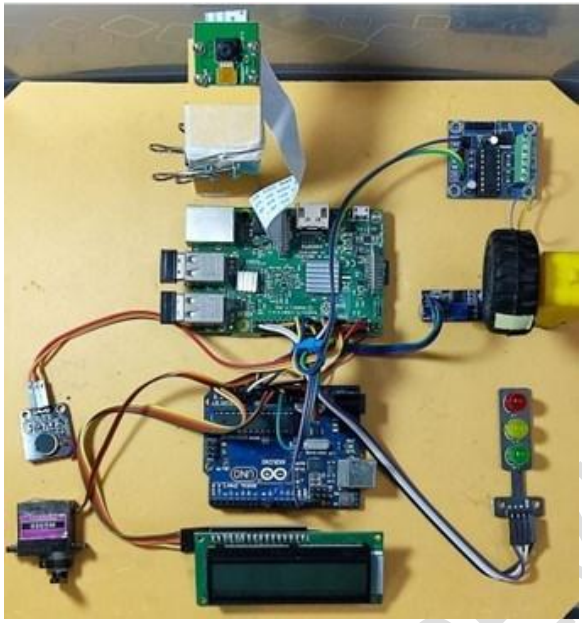

Fig. 5. Hardware setup of the complete system.

6. RESULTS AND DISCUSSIONS

Figures6-8 show the simulation resultsof the proposed system obtained by using the Proteus simulator.Figure 6 shows that the faces and eyes of the driver are detected, yellow and green LEDs are turned on, the red LED is turned off, and as such the motor is rotating. The LCD screen displays this message.

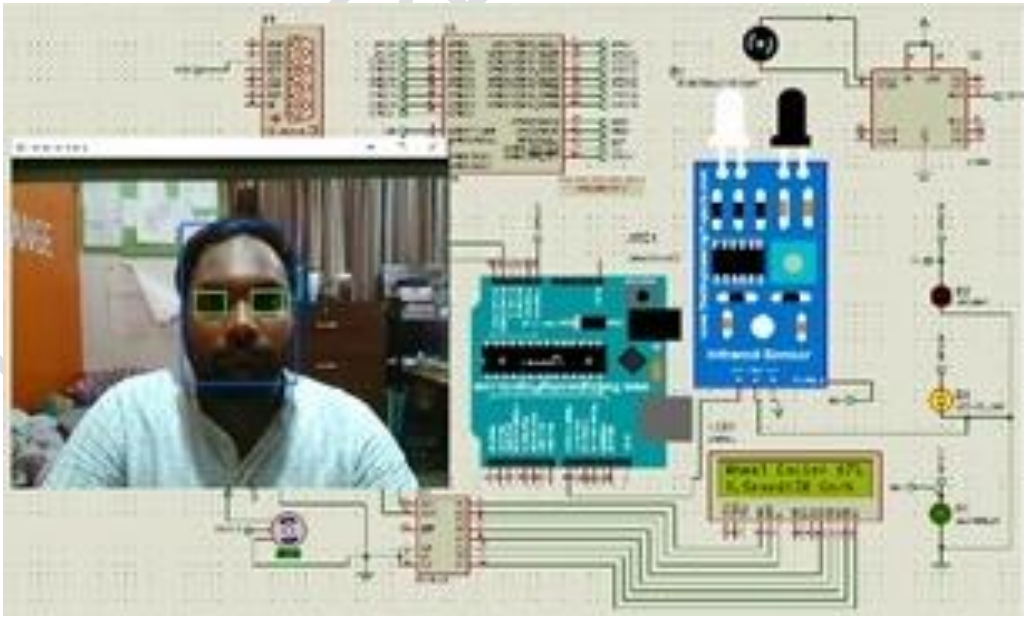

Fig.6.The face and eyes of the driver are detected, the Yellow and Green LEDs are on, the Red LED is off. The wheel motor is rotating, and the servo motor is at 90º.

Figure 7 shows that the face of the driver is detected but eyes are not detected, yellow and red LEDs are turned on, but the green LED is turned off, and as such the motor is not rotating. The LCD screen displays this message.

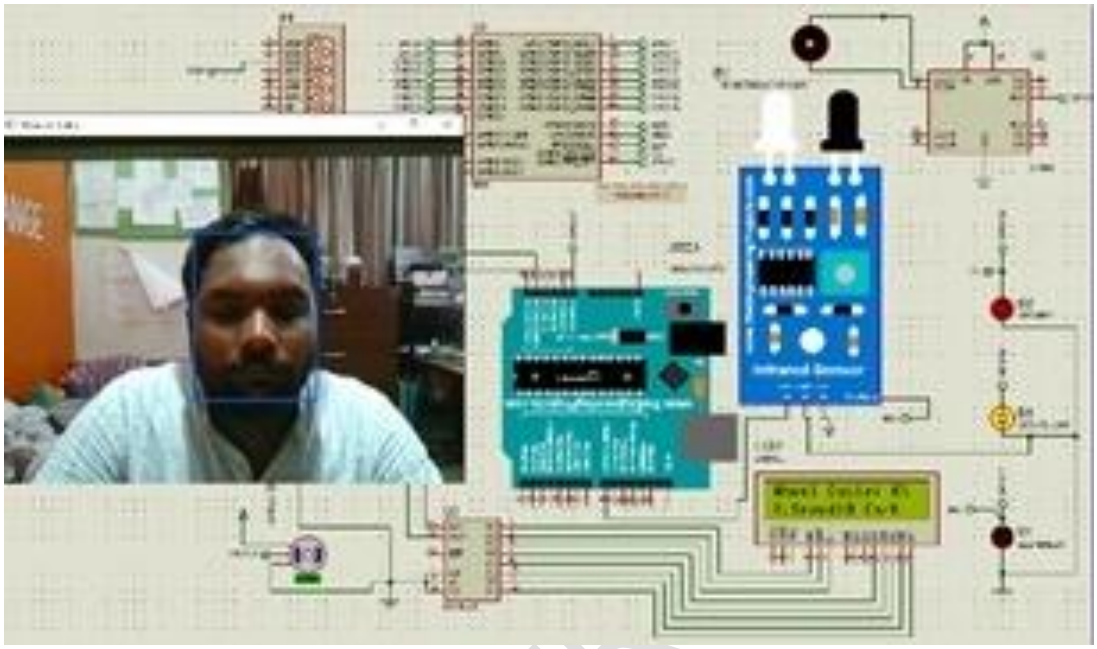

Fig.7.The face of the driver is detected but the eyes are not detected. Yellow LED and Red LED are on, Green LED is off. The wheel motor is not rotating, and the servo motor is at 180º.

Figure 8 shows that the face and eyes of the driver are not detected, the red LED is turned on, but the yellow and green LEDs are turned off, and as such the motor is not rotating. The LCD screen displays this message.

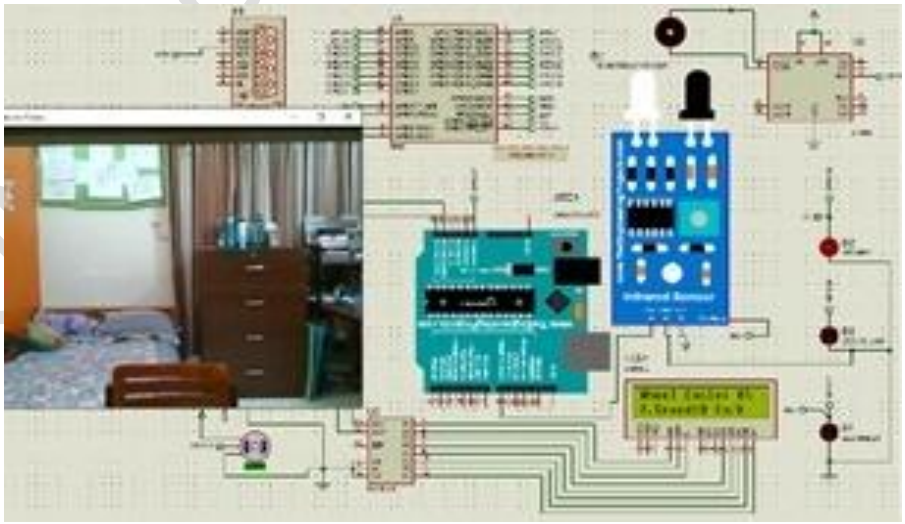

Fig.8.The face and eyes are not detected, the Red LED is on, and the Yellow and Green LEDs are off. The wheel motor is not rotating, and the servo motor is at 180º.

Figures9-11demonstrate the real-time experimental results under the same conditions used to obtain the simulation results of the proposed system. Figure 9 shows that the faces and eyes of the driver are detected, yellow and green LEDs are turned on, the red LED is turned off, and as such the motor is rotating at 30 km/h. The LCD screen displays this message.

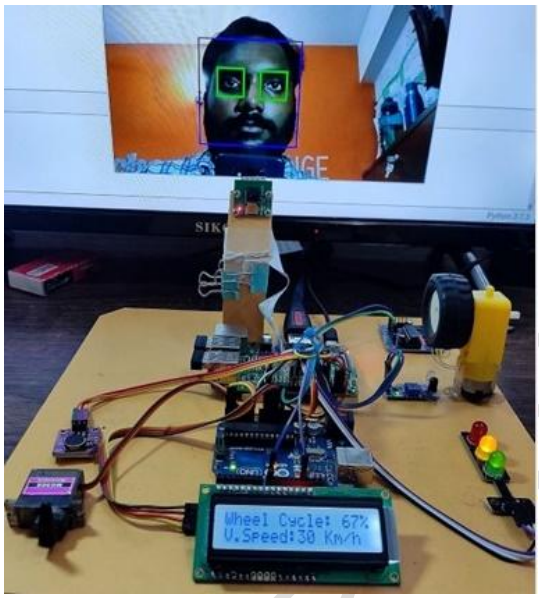

Fig.9.The face and eyes of the driver are detected, the Yellow LED and Green LED are on, the Red LED is off. The wheel motor is rotating, and the servo motor is at 90º.

Similarly, Fig. 10 shows that the face of the driver is detected but eyes are not detected, yellow and red LEDs are turned on, but the green LED is turned off, and as such the motor is not rotating, it shows zero speed on the LCD screen.

Fig. 10. The face of the driver is detected but the eyes are not detected. The Yellow and Red LEDs are on, and the Green LED is off. The wheel motor is not rotating, and the servo motor is at 180º.

Finally, Fig. 11 shows that the face and eyes of the driver are not detected, the red LED is turned on, but yellow and green LEDs are turned off, and as such the motor is not rotating, its speed is zero which is displayed on the LCD screen.

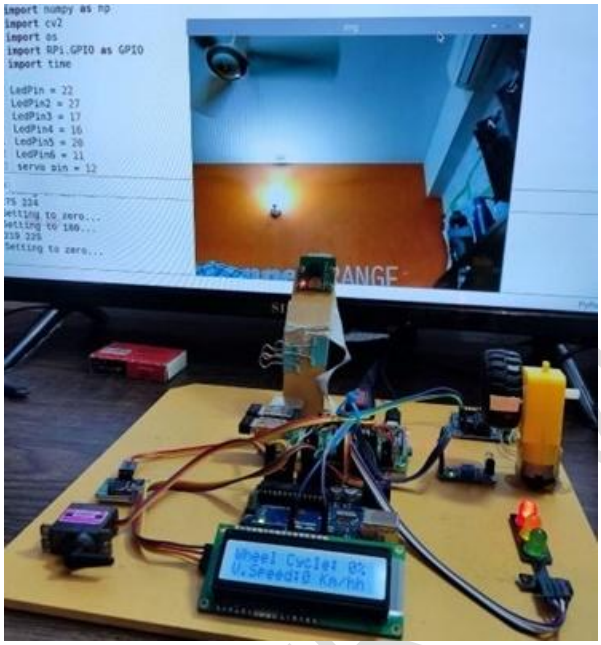

Fig. 11. The face and eyes are not detected, the Red LED is on, Yellow and Green LEDs are off. The wheel motor is not rotating, and the servo motor is at 180º.

Table 1 shows the breakdown of prices of various parts used in this task. The entire cost necessary for this work is BDT22,185 (Bangladeshi Taka twenty-two thousand one hundred and eighty-five only), approximately US$201.69 (US Dollar two hundred one point six nine).

Table 1.Cost estimation ofa driver's drowsiness detection system.

Sl. No.	Items	Quantity	Unit Price (BDT)	Price (BDT)	Price (US$)*
1	Raspberry Pi 3 Model B	1	19,000	19,000	172.73
2	Raspberry Camera Module	1	960	960	8.73
3	Arduino Uno R3	1	840	840	7.64
4	16x2 Serial LCD Module	1	340	340	3.09
5	MG90s Micro Metal Servo	1	250	250	2.27
6	DC Vibration Motor Module	1	220	220	2.00
7	L293D Motor Driver Module	1	215	215	1.95
8	DC gear motor and Wheel	1	180	180	1.64
9	LED Light Module	1	120	120	1.09
10	IR Sensor Module	1	60	60	0.55
Total Prices in BDT and US$, respectively				**22,185**	**201.69**

** Costs are given based on prices of the components in Bangladesh in Bangladeshi Taka (BDTK). But it may vary depending on the country of purchase and dollar rate. In general, 1 US$ = 110 BDT.*

According to the driver drowsiness detection system market survey data, the market size of this product is going to grow cumulatively at an annual compound rate of 11.8%. The current market value of this product was estimated at US$8 billion in 2023. It is projected that this may go above US$24.4 billion in 2033 if this growth rate persists over a period of the next 10 years from 2024 to 2033 [18]. Hence, we must carry out research extensively in this field.

7. CONCLUSION

In conclusion, the development of the automobile operator's drowsiness detection and alert system merges hardware components and computer vision to increase driver safety and reduce the likelihood of accidents caused by the drowsiness of drivers. The system can alert the driver when necessary and take the appropriate actions when required. An elaborate safety net is built through the collaboration of Arduino's knowledge and skills for RPM monitoring and PWM management by face and eye detection employing the Raspberry Pi 3B microcontroller board. A notable feature is that the LCD screen functions as a simple user interface and provides real-time information on the motor's performance. This combination of advanced computer vision and exact hardware control not only highlights the promise of multidisciplinary problem-solving techniques but also establishes a viable course for resolving concrete real-world problems.

There are some limitations in our system. For example, using OpenCV's cascades for face and eye detection may not always yield accurate results, which might result in false positives or missing detections. This can lead to inaccurate sleepiness detection and erroneous alarms. The performance of the system may be impacted by different lighting conditions. The accuracy of face and eye detection may be impacted by various lighting conditions and angles, which might result in inaccurate detection. Because the Raspberry Pi and Arduino run separately, it might be difficult to ensure precise synchronization between the camera frames, face/eye identification, motor control, and sensor readings. An incorrect understanding of the driver's condition might result from inaccurate synchronization. The alarms from the system could not consider outside variables like medication, fatigue, or medical issues that could have an impact on the driver's mood, and hence can yield incorrect warnings.

A potential expansion of this initiative involves transforming it into a mobile application, curbing hardware expenses, and broadening its accessibility to drivers to avert accidents.The processing speed could be augmented by incorporating Artificial Intelligence techniques, consequently enhancing the precision of the system's algorithms. Messaging services could be integrated, notifying a designated mobile number when driver drowsiness is detected. The inclusion of a GPS module stands as another potential enhancement, permitting the monitoring of the driver's location in conjunction with the drowsiness detection system. Exploring the integration of health monitoring sensors represents a prospect, introducing a holistic dimension to the system by tracking the driver's health parameters.

CONSENT

There is no consent required for this publication except for ours.

ETHICAL APPROVAL

There is no ethical approval requirement for this task.

REFERENCES

1. Ramani MVSR, Tejaswee A, Gupta PSTB, Harikrishna K, Sri GU. (2020). Driver Fatigue and Drowsiness Detection System. International Journal of Engineering Applied Sciences and Technology.2455-2143. 5(3), 231-236 Link: *https://www.ijeast.com/papers/231-236,Tesma503,IJEAST.pdf*.
2.Titare S, Chinchghare S, Hande KN. (2021). Driver Drowsiness Detection and Alert System. International Journal of Scientific Research in Computer Science, Engineering, and Information Technology. 7(3). 583-588. DOI: https://ijsrcseit.com/paper/CSEIT2173171.pdf.
3. MS, BP, SM, Hegde SM, and NHP. (2022). Driver Drowsiness Detection System. International Journal for Research in Applied Science & Engineering Technology (IJRASET).10(5), 129. DOI: https://www.ijraset.com/best-journal/driver-drowsiness-detection-system.
4. Laxmi R, Rafi SKM, Rao PPR, Kumar BL, Narayana NVL. (2022). Automatic Driver Drowsiness Alert and Health Monitoring System. International Journal of Engineering, Technology and Management Sciences (IJETMS). 6(4),58-61.DOI: 10.46647/ijetms.2022.v06si01.011.
5. Kumar AJS. Sanjana P, Sanjay N, Sanjay KY, Kodgi SU. (2022).Smart Alert System for Driver's Drowsiness Detection System. International Journal of Computer Applications.184(24), 21-26, DOI:https://www.ijcaonline.org/archives/volume184/number24/kumar-2022-ijca-922282.pdf
6. About OpenCV. Accessed on12 August 2023. [Online].
7. Python Time Module.GeeksforGeeks. Accessed on 12August 2023. [Online].
8. Russel MK, Bhuyan MH. (2012). Microcontroller Based DC Motor Speed Control Using PWM Technique. Proceedings of the IEEE International Conference on Electrical, Computer and Telecommunication Engineering (ICECTE), Rajshahi, Bangladesh, pp. 519-522.
9. Titlee R. Bhuyan MH. (2016). Design, implementation and testing of ultrasonic high precision contactless distance measurement system using microcontroller. Southeast University Journal of Science and Engineering (SEUJSE), p-ISSN: 1999-1630. 10(2), 6-11.
10. Bhuyan MH, Ali MA, Khan SA, Islam MR, Islam T, Akter J. (2023). Design and implementation of solar power and an IoT-Based pisciculture management system. Journal of Engineering Research and Reports, ISSN: 2582-2926. 24(2), 15-27.
11. Paul NK. Saha D. Biswas K. Akter S. Islam RT. Bhuyan MH. (2023). Smart Trash Collection System – An IoT and Microcontroller-Based Scheme. Journal of Engineering Research and Reports, 24(11), 1-13. https://doi.org/10.9734/jerr/2023/v24i11849.
12. Bhuyan MH, Sheikh M. (2021). Designing, implementing, and testing of a microcontrollerand IoT-based pulse oximeter device. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), 16(5):series-2, 38-48, e-ISSN: 2278-1676, p-ISSN: 2320-3331. https://doi.org/10.9734/jerr/2023/v24i2799.
13. Bhuyan MH, Hasan M. (2020). Design and Simulation of Heartbeat Measurement System using Arduino Microcontroller in Proteus. International Journal of Biomedical and Biological Engineering (IJBBE). 14(10). e-ISSN: 1307-6892, 350-357.
14. NumPy Introduction. W3schools. Accessed on 12October2023. [Online].
15. OS Module in Python with Examples. GeeksforGeeks. Accessed on 25August2023. [Online Resources].

16.Sicad J. Raspberry Pi: PWM in GPIO (Python). Radish Logic. Accessed on 28July2023. [Online Resources].
17. Face detection using Cascade Classifier using OpenCV-Python. GeeksforGeeks.Accessed on 20 September 2023. [Online Resources].
18. Market Research on Drowsiness Detection System. https://market.us/report/driver-drowsiness-detection-system-market/,Accessed on 2 December 2023. [Online Resources].