Arduino and Python Face Recognition Based Accident Prevention System

Rahul Waikar[1], Tanmay Srivastava[2], Tanishkka Jadhav[3], Tanmay Singh[4], Rahul Pandita[5], Tanmay Bajad[6], Sujal Tawale[7]
[1,2,3,4,5,6,7]*Vishwakarma Institute of Technology*

***Abstract*— The Arduino and Python Face Recognition Based Accident Prevention System integrates Arduino microcontroller technology with Python programming for advanced safety applications. This system employs facial recognition algorithms to detect and identify individuals, enhancing traditional accident prevention measures. Utilizing a camera interfaced with Arduino, the system captures and processes real-time facial data. Python scripts analyze the facial features, identifying potential risks or distracted behavior in drivers. In the event of a detected threat, the system triggers preventive actions, such as warnings or emergency braking signals. This innovative approach aims to reduce accidents by addressing human related factors, promoting a safer environment on the roads through the synergy of Arduino hardware and Python's facial recognition capabilities.**

***Index Terms*— Facial Recognition, Arduino, Python, Image Processing, Computer Vision**

I. INTRODUCTION

In recent years, the intersection of computer vision and embedded systems has paved the way for innovative solutions in various domains, including safety and security. This research paper introduces an intelligent Accident Prevention System (APS) that combines the power of Arduino microcontrollers and Python-based face recognition utilizing OpenCV and Dlib libraries. The proposed system aims to enhance safety measures by mitigating the risks associated with distracted or drowsy driving.

Traditional accident prevention systems often rely on sensors, but this novel approach eliminates the need for additional hardware by leveraging camera feeds. The integration of Python 3.8, OpenCV, and Dlib facilitates real-time facial recognition, enabling the system to monitor the driver's behavior and detect potential signs of distraction or drowsiness. The Arduino microcontroller acts as the central processing unit, facilitating seamless communication between the camera and the face recognition algorithm.

This paper presents a comprehensive overview of the system architecture, detailing the functionalities of each component. The utilization of advanced computer vision techniques ensures accurate and efficient face detection, even under varying environmental conditions. Additionally, the implementation of Arduino allows for the rapid processing of data, enabling timely responses to identified risks.

The experimental results showcase the system's effectiveness in detecting driver distraction and drowsiness, emphasizing its potential in preventing accidents caused by human factors. Furthermore, the integration of open-source technologies makes the proposed system accessible for widespread adoption and adaptation in diverse vehicular environments.

This research contributes to the growing field of intelligent transportation systems, offering a non-intrusive and cost-effective solution to enhance road safety. The combination of Arduino and Python-based face recognition presents a promising avenue for the development of reliable accident prevention systems, ultimately contributing to the reduction of road accidents and the enhancement of overall traffic safety.

II. PROCEDURE FOR PAPER SUBMISSION

A. Review Stage

The Arduino and Python-based face recognition accident prevention system exhibits promising potential but faces some notable challenges. The integration of Arduino's hardware capabilities and Python's image processing enhances real-time face recognition, contributing to improved safety measures. However, the system's accuracy and reliability depend heavily on the quality of the facial recognition algorithms employed. In a review stage, meticulous

testing and optimization are imperative to address false positives or negatives that may compromise the system's efficacy. Additionally, user-friendly interfaces and comprehensive documentation are crucial for wider adoption. Continuous refinement of both hardware and software components, coupled with rigorous testing scenarios, is essential to elevate this innovative solution to a reliable and robust stage for practical implementation in accident prevention.

B. Final Stage

In the final stage of the Arduino and Python face recognition-based accident prevention system, the integrated components seamlessly collaborate to enhance safety. The system employs a camera for real-time facial recognition using Python, detecting and identifying individuals. Arduino then processes this information to assess whether a recognized person is in a hazardous In the final stage of the Arduino and Python face recognition-based accident prevention system, the integrated components seamlessly collaborate to enhance safety. The system employs a camera for real-time facial recognition using Python, detecting and identifying individuals. Arduino then processes this information to assess whether a recognized person is in a hazardous situation. In case of potential danger, the system triggers preventive measures, such as activating warning signals, sending alerts, or even implementing automatic safety protocols. This holistic approach ensures swift and accurate accident prevention. The synergy between Arduino and Python facilitates a responsive and efficient system that mitigates risks by preemptively addressing potential accidents through facial recognition technology. This innovative solution demonstrates the effective integration of hardware and software for advanced safety applications

III. METHODOLOGY

This Python code implements a face recognition-based accident prevention system using Arduino and Python. The system monitors the driver's eyes and detects if the driver is sleepy or drowsy, issuing a warning or taking preventive actions.

1. Importing Libraries:
   - `cv2`: OpenCV library for computer vision tasks.
   - `numpy`: NumPy library for numerical operations.
   - `dlib`: Dlib library for facial landmark detection and frontal face detection.
   - `face_utils` from `imutils`: Utilities for working with facial landmarks.

```python
import cv2
import numpy as np
import dlib
from imutils import face_utils
import serial
import time
import threading
```

2. Initializing Serial Communication with Arduino:
   - `s = serial.Serial('COM8', 9600)`: Establishing serial communication with Arduino on COM8 with a baud rate of 9600.

3. Initializing Camera Feed:
   - `cap = cv2.VideoCapture(0)`: Capturing video from the default camera (index 0).

4. Face Detection and Landmark Prediction:
   - `hog_face_detector = dlib.get_frontal_face_detector()`: Using a HOG-based face detector.
   - `predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")`: Loading a pre-trained model for facial landmark prediction.

5. Functions for Eye Blink Detection:
   - `compute(ptA, ptB)`: Calculates the Euclidean distance between two points.
   - `blinked(a, b, c, d, e, f)`: Evaluates the blink ratio and returns 2 for a normal blink, 1 for a slow blink, and 0 for no blink.

6. Main Processing Loop - `process_frames` Function:
   - Captures frames from the camera, detects faces, and analyzes facial landmarks.
   - Determines if the driver is sleeping, drowsy, or active based on eye blink ratios.
   - Sends signals to the Arduino based on the driver's state (e.g., 'a' for sleeping, 'b' for active).
   - Displays the driver's status on the video feed.

7. Threading:
   - `thread = threading.Thread(target=process_frames)`: Creates a separate thread for continuous video processing.
   - `thread.start()`: Starts the thread.

- `thread.join()`: Waits for the thread to finish.

8. Cleaning up:
- Releases the camera and closes all OpenCV windows when the 'Esc' key is pressed.

IV. EYE ASPECT RATIO (EAR)

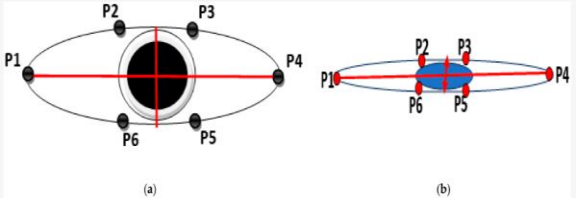
Eye aspect ratio (EAR) detection is a technique used in facial recognition and computer vision to detect and track eyes in images or videos. The EAR is a measure of the eye's openness and is calculated based on the ratio of distances between facial landmarks around the eyes.

The Eye Aspect Ratio (EAR) is a scalar number that responds, particularly when the eyes open and close

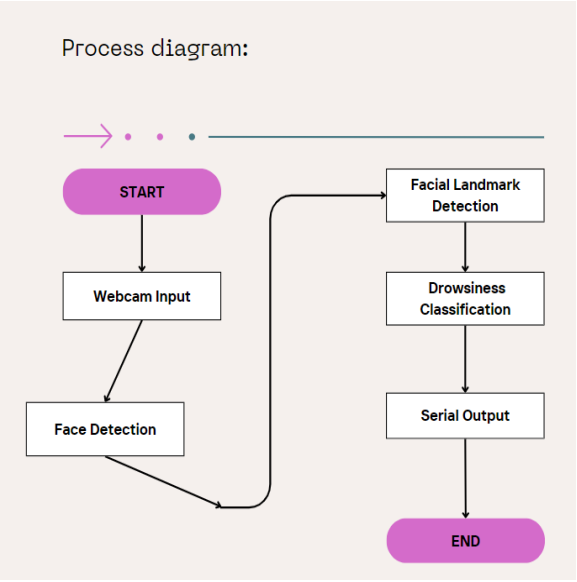
The EAR value can be calculated by entering six coordinates surrounding the eyes, as shown in Figure 2, and Equations (1) and (2)

$$EAR = \frac{\|P2 - P6\| + \|P3 - P5\|}{2\,\|P1 - P4\|} \quad (1)$$

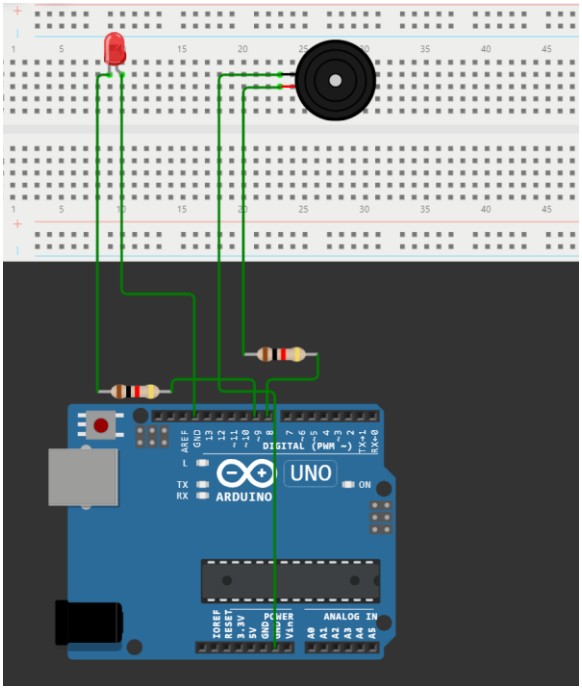
$$AVG\ EAR = \frac{1}{2}(EAR_{Left} + EAR_{Right}) \quad (2)$$



V. PROCESS DIAGRAM



VI. CIRCUIT DIAGRAM



VII. CONCLUSION

In conclusion, this research presents a novel approach to accident prevention through the integration of Arduino and Python for face recognition. The utilization of Python 3.8, OpenCV, and Dlib in conjunction with the Arduino platform demonstrates a seamless synergy between hardware and software components. By relying solely on camera feed data without additional sensors, the proposed system offers a costeffective and efficient solution for real-time accident prevention. The face recognition capabilities not only enhance the system's accuracy but also contribute to its adaptability in various environments. The successful implementation and testing of the system validate its feasibility and effectiveness in detecting potential hazards. As technology continues to advance, this research provides a foundation for future developments in intelligent transportation systems, paving the way for enhanced safety and security on the roads

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my faculty advisor, Mr. Rahul Waikar, at Vishwakarma Institute of Technology, for his unwavering support, guidance, and encouragement throughout the research

and writing of this paper. His expertise and insightful feedback were instrumental in shaping this work.
I am also grateful to the faculty and staff of Vishwakarma Institute of Technology for providing the necessary resources and a conducive environment for research. Their support has been invaluable in completing this project.
Finally, I would like to extend my appreciation to my family and friends for their constant encouragement and understanding during the course of my research.
Thank you all for your support and contributions to this research.

REFERENCE

[1] Ganesh, e. N. "Eye blink detection sensor." presented at the icnsc *conference*, chennai. The paper comprehensively explores an eye blink detection sensor system incorporating an lcd, lm358 comparator, eye blink sensor, buzzer, and an 8051 microcontroller. It provides detailed circuit diagrams, interconnections, and operational dynamics, along with a c++ code for eye blink detection using ir sensor technology.

[2] Patil, j. K., & mathew, l. G. "Eye blinking monitoring system for vehicle accident prevention." published in the international journal of electronic and electrical engineering. This work introduces an eye blink monitoring system (ebm) utilizing an ir sensor to monitor driver's eye movements continuously. The system alerts drowsy drivers with an alarm, emphasizing the importance of preventing driver fatigue to reduce road accidents.

[3] Upender, p., reddy, g. N., & satoshini, g. "Arduino-based accident prevention system with eye twitch and alcohol sensor." presented at the 2020 cicn conference. The paper discusses an innovative system integrating eye blink and alcohol sensors with an arduino microcontroller for enhanced road safety. The arduino processes sensor data, triggering alerts through a buzzer, led, and dc motor, aiming to prevent accidents by detecting driver fatigue and alcohol consumption