International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Accident Avoidance using Machine Learning and Arduino

Sowmya Sunkara*, Pavan Kumar C[1], Abhishek Krishna Puranik [2], Mohan M [3], Kiran Kumar KR[4]

*Department of Electronics and communications

BMS College Of Engineering

Abstract: **Safety and speed of a car goes hand in hand. In today's society we need to implement and We should Keep on upgrading the safety on the road. Basically, the speed of the vehicle is the prima facia issue in India and elsewhere in the world were extreme speed causes accidents. Hence, we need to create a solution where the car automatically reduces speed after seeing the speed limit board for that particular flyover or the underpass and also detect and report the drowsiness.**

Keywords-Arduino UNO, CNN model architecture

1.Introduction

In a diverse world which we live, everything is interconnected in society. We need homes to live, cars and other form of transports to travel to either our offices, colleges, outstation trips, malls, restaurants etc.

According to the statistics by NFHS In India they were about 100 million registered private vehicles (2018) and it is rapidly growing proportionally at the rate of 7.5% and there are over 600 million+ Over 50% still use bicycles, bikes & scooters. Then there are also 65-70 million commercial trucks, buses, containers.

There is basically an infrastructure pressure across India especially in the major metropolitan cities of India. Hence the govt. in the past 3 decades have subsequently increased the budget every year on road building across the nation and within the cities.

In the year 2023, Govt would be investing 2.7lakh crore rupees for the development of the road highway infrastructure. Even though the govt.is making the efforts to build a robust infrastructure. The public is misusing the highways, freeways, flyovers, underpasses by over speeding and causing accidents. Road safety continues to be a major developmental issue, a public health concern and a leading cause of death and injury across the world. At least one out of 10 people killed on roads across the world is from India, according to the World Health Organization. The cost of road accidents is borne not only by the victims and their family, but by the economy as a whole in terms of untimely deaths, injuries, disabilities and loss of potential income. It is indeed a matter of great concern that despite the continuing efforts of the Government in this regard and our commitments for halving fatalities we have not been able to register significant progress on this front.

During the year 2021, a total number of 4,12,432 road accidents have been reported in the country, claiming 1,53,972 lives and causing injuries to 3,84,448 persons. Unfortunately, the worst affected age group in Road accidents is 18-45 years, which accounts for about 67 percent of total accidental deaths.

In order to solve this over speeding. We come with an interesting solution that is with using Arduino uno, dc motors, l298n motor, we will be implementing these all where the cars front camera will detect a speed limit board and however the speed of the car it will automatically bring the car down to the given speed for that particular stretch.

Drowsy driving is the dangerous combination of driving when sleepy. This usually happens when a driver has not slept enough, but it can also happen because of untreated sleep disorders or shift work. Prescription and over-the-counter medications can also cause drowsiness, and alcohol can interact with sleepiness to increase both impairment and drowsiness.

In India according to ministry of transport there are 40% accidents in India due to drowsiness in one form or the another.

II.Proposed solution

The proposed solution is that:

To detect speed limit signs the CNN algorithm is used which is very accurate in image classification. When the speed limit board is detected, the Arduino will adjust speed accordingly. The proposed system will give solution for the above addressed problems by implementing:

1. Accurate algorithm to analyses the driver behaviour and identify signs of drowsiness in real time.
2. A non-intrusive monitoring without using any bio-sensors, that collect data without causing discomfort or distraction to the driver.
3. A real-time alert and speed control mechanism to promptly notify the driver when drowsiness or speed limit zone is detected using audible and visual indications.

Drowsiness can be detected by monitoring the driver through continuous video stream with a mobile or camera. Our system will detect the drowsiness in real-time using the concept of facial landmarks and eye aspect ratio. Upon detection of drowsiness the ECU (Arduino UNO) will

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

effectively reduce the speed of the vehicle to avoid catastrophic results.

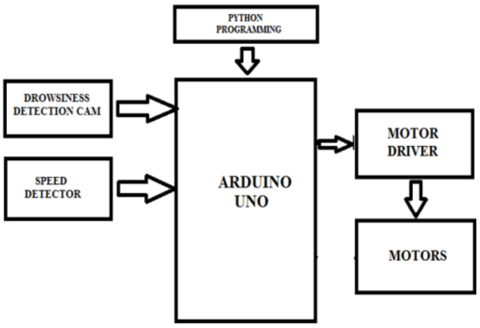

Fig 1: Block Diagram

III: Working methodology

The project utilizes the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which contains a diverse range of traffic sign images, including various speed limit signs. The image dataset is consists of more than 50,000 pictures of various traffic signs(speed limit, crossing, traffic signals, etc.) Around 43 different classes are present in the dataset for image classification.

Workflow: the below 4 steps are followed to build our classification model:

1) Dataset exploration

2) CNN model building

3) Model training and validation

4) Model testing

Dataset Exploration: The data set contains 43 subfolders labelled from 0 to 42, each representing a distinct class. But we are using only 9 subfolders, each representing 0 to 8 distinct class. We utilize the OS module to iterate through all the images within these subfolders, along with their corresponding classes and labels. In order to store the contents of the images into an array, we make use of the PIL library.

CNN Model Building:

The model architecture is as follows:

• 2 Conv2D layer (filter=32, kernel size=(5,5), activation=”relu”)

• MaxPool2D layer (pool_size=(2,2))

• Dropout layer (rate=0.25)

• 2 Conv2D layer (filter=64, kernel_size=(3,3), activation=”relu”)

• MaxPool2D layer (pool_size=(2,2))

• Dropout layer (rate=0.25)

• Dense layer (43 nodes, activation=” softmax”)

The following steps are followed:

1) Start by adding the layers in a specific order: two convolutional layers, one pooling layer, a dropout layer, a flattening layer, a dense layer, another dropout layer, and a final dense layer. 2) The convolutional layer applies a specified number of filters to the original image, performing a convolution operation and generating a feature map.

3) The ReLU (Rectified Linear Unit) function is used to transform the feature map, converting negative values to zero while keeping positive values unchanged, resulting in a rectified feature map.

4) The pooling layer takes the rectified feature map and performs a down-sampling operation, such as Max Pooling or average pooling, to reduce the dimensionality of the image.

5) The flattening layer is responsible for converting the input feature map into a 1-dimensional array.

6) The dropout layer helps prevent overfitting by randomly setting some of the input neurons to zero during the training process. The dense layer connects all the outputs from the preceding layer to its neurons and performs matrix-vector multiplication, generating an m-dimensional vector. It's important that the dimensions of the preceding layer's output match the dense layer's input.

7) Once the layers are added, the model needs to be compiled, which involves defining the loss function and applying optimization techniques. In this case, the loss function is set "sparse_categorical_crossentropy," suitable for a multiclass classification problem where each image belongs to exactly one class. The "Adam optimizer" is used for optimization

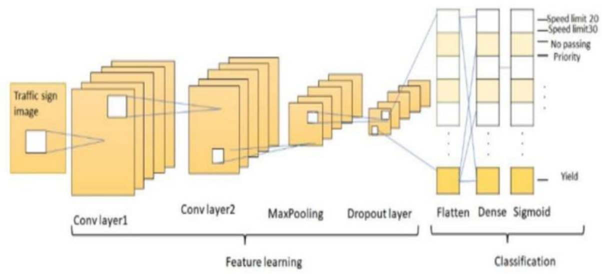

Fig 4.2: CNN Model Architecture

Fig 2.CNN Model Architecture

Model Training and Validation:

The next step involves training the model using the training dataset. The pre-processed images from the training dataset are passed through the model during this phase.

To train our model, we will use the model.fit() method that works well after the successful building of model architecture. 80% of images are used to train the model and the remaining 20% is used for validation.

Model Testing: Finally, the trained model is used to make predictions on the test data. The output includes the predicted traffic sign name and its corresponding class ID. We have a dataset with a folder named "label" containing a main comma-separated file called "label.csv". This file contains image paths and their corresponding class labels. To extract the image paths and labels, we can utilize the panda's library in Python. Afterwards, we need to resize the images to dimensions of 30x30 pixels in order to make predictions using the model. We will create a NumPy array filled with the image data. To evaluate how accurately the model predicts the labels, we will import the accuracy score function from the sklearn. metrics library. Finally, we will save our trained model using the Keras model.save() method. Given below in fig3.is the working of CNN model

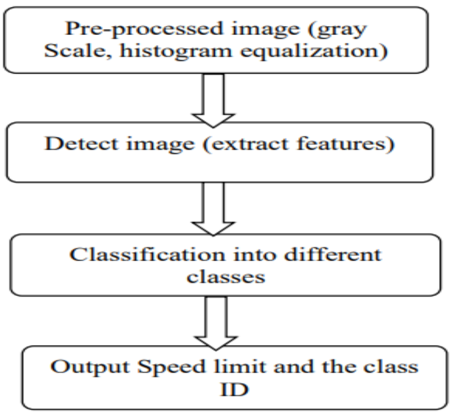

Fig 3. Basic flow of CNN model

Now we will discuss about the Drowsiness detection-

1) The first step of the project involves accessing the webcam to capture video frames. These frames are typically recorded at a rate of 30 frames per second, and we will process one frame every 1/30th of a second. Through a continuous loop, each frame is subjected to various processes aimed at detecting facial features.

2) Once a frame is obtained, our program employs image-processing techniques to identify specific regions of interest, such as the face and the Euclidean distance between eyes. This involves applying a series of functions to modify the frame in a manner that facilitates easy detection of features like the eyes, mouth, nose, and so on.

The Processing steps are:

Resizing:

The initial step involves flipping the image over the y-axis and then resizing it. The purpose of resizing is to adjust the image to a specific resolution according to the project's requirements. In this case, the new resolution set is 640x480.

BGR to ray: To achieve more accurate results in detecting different parts of the face and hand, the image needs to be converted from RGB to grayscale format. This conversion is necessary as the subsequent data processing techniques rely on grayscale images. Converting the image to grayscale allows for easier identification of facial features.

Detection and Prediction of Facial Features: The project utilizes a prebuilt model (Dlib model) for face detection and feature prediction. Dlib model uses HOG algorithm for facial feature Extraction and SVM for classification. This model provides the necessary functionality for locating faces within an image. The "detector ()" function is used to detect the face, while the "predictor ()" function helps identify 68 specific points on the face. These points represent various facial landmarks such as the eyes, mouth, etc. The resulting coordinates are stored in arrays for further processing, enabling the creation of boundaries or contours around specific facial regions.

Eye Aspect Ratio: Once the facial contours are established, reference ratios are utilized to monitor and interpret movements within specific regions of the face. By comparing changes in the 2D coordinates, actions such as blinking or yawing can be detected. This system relies on the Dlib prebuilt model, which provides fast and accurate face detection and landmark prediction. Specifically, the Eye-Aspect-Ratio (EAR) is used to detect blinking/winking or closing of eyes.

Fig 4. 68 Facial landmarks

This project revolves around the accurate prediction of facial landmarks for a given face, offering a wide range of potential applications. These landmarks can be utilized for various purposes, such as detecting eye blinks in videos or predicting the subject's emotions. The possibilities and outcomes associated with facial landmarks are vast and fascinating. Dlib's prebuilt model, which implements an efficient face detection algorithm, enables us to precisely predict 68 2D facial landmarks while maintaining speed.

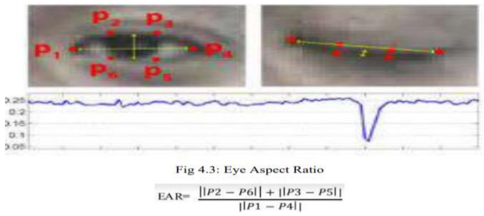

Fig 5. Eye Aspect Ratio

The graph clearly demonstrates a significant decrease in the EAR (Eye Aspect Ratio) value when the eye is closed. This observation can serve as a reliable trigger for performing drowsiness detection. When the eye is closed for more than the threshold time (i.e., EAR drops below threshold for longer time) then a warning like alarm is generated to warn about the drowsiness. The working of EAR is given in **Fig5.**

IV. Results and discussions-

In the **Fig 6** below is the hardware implemented protocol here we will be seeing the built-up car model which will be first running in a preset 50-80km/h speed. Then according to the speed limit board e.g.: 20km/h the webcam which is here enabled we will see that the speed limit captured sends the data via Bluetooth HC-05 module to the Arduino by using main library pyfirmata. Pyfiramata turns the Arduino into a data acquisition card controlled by Python. So by using pyfirmata we will be controlling the Arduino and then from Arduino the data is sent to the L298N motor where it will act according to the data being received by the Arduino.

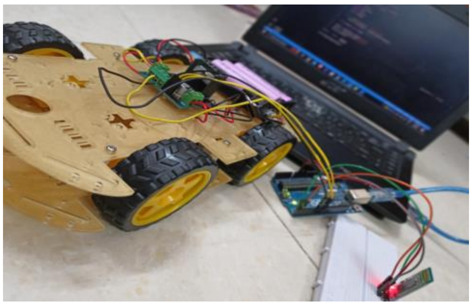

Fig 6. Hardware implemented model car

Fig 7. Speed limit board

The speed limit sign board- Here the speed limit board as shown in **Fig 7.** is detected and then it reduces the car speed from 50km/h to 20 km/h speed. Components Being used for the hardware implementation-

Arduino UNO- The Arduino UNO is a standard board of Arduino, which is based on an ATmega328P microcontroller. It is easier to use than other types of Arduino Boards. The Arduino UNO Board, with the specification of pins.

Motor driver L298N: The L298N Motor Driver module consists of an L298 IC Dual H-bridge, 5V Voltage Regulator, resistors, capacitor, Power LED, 5V jumper. 2 DC motor output pins, 12-volt external motor power supply, motor direction control pins (IN1, IN2, IN3, IN4), motor output enable pins (ENA, ENB), and a heat sink.

Bluetooth HC-05 module- HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration. This Bluetooth module is based on Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with a complete 2.4GHz radio transceiver and baseband.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

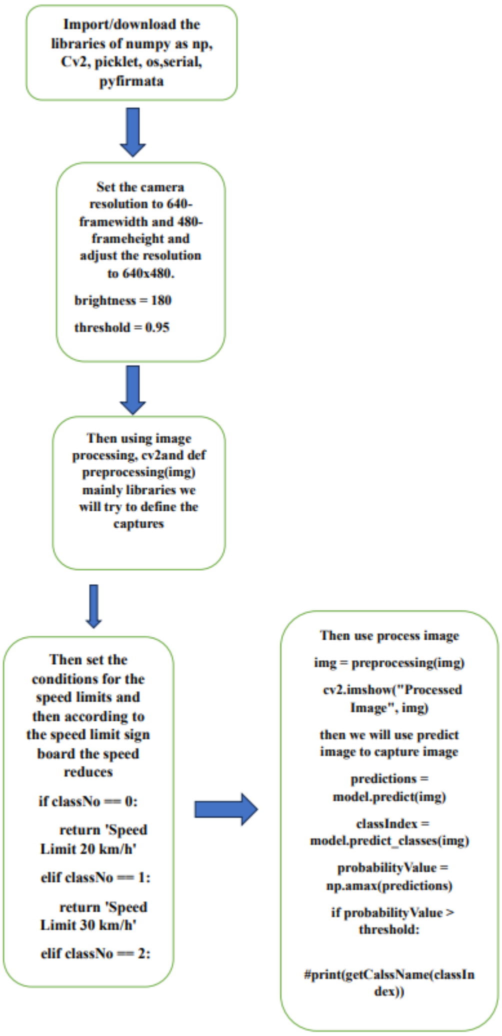

Fig 8. The Flowchart working of speed detection and reduction

Drowsiness- After the drowsiness is detected by the number of seconds the persons eyes are closed then it reduces the speed and eventually stops. The drowsiness process is being defined in the above working methodology. Here according to the threshold set for the seconds the drowsiness detection process is activated as given below the threshold is set at 0.25s

```
def eye_aspect_ratio(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

thresh = 0.25
```

V.Conclusions

The project on drowsiness detection and speed control using Dlib and concepts of facial landmarks and EAR (Eye Aspect Ratio) has successfully achieved its objectives of enhancing road safety by addressing the issue of drowsy driving. By continuously monitoring the EAR, the system could detect signs of drowsiness in real-time. The integration of speed control mechanisms into the system further contributed to its effectiveness. When drowsiness was detected, the system automatically adjusted the vehicle's speed to a safer level, reducing the risk of accidents caused by slower reaction times or loss of control due to driver fatigue. By utilizing CNN algorithm, Arduino microcontroller, DC motor, and a driver IC, the project achieves two main objectives: detecting speed limit signs and controlling the speed of a vehicle accordingly. The DC motor, connected to the wheels or throttle mechanism of the vehicle, allows for precise speed control. This ensures that the vehicle adheres to the appropriate speed restrictions, enhancing safety and compliance with traffic regulations. However, it is important to note that the project's success depends on various factors such as the accuracy of the CNN algorithm, the quality of image processing, and the reliability of the hardware components. Continuous testing, refinement, and updates are necessary to ensure optimal performance and real-world viability.

VI. Future scope

• One possible way to expand the scope of the project is by incorporating mouth feature extraction to detect drowsiness in drivers. By analysing patterns of yawning, it becomes possible to identify when a driver is exhibiting signs of drowsiness. If the driver yawns multiple times in quick succession, it could indicate that they are becoming sleepy.

• Along with speed limit board detection, the project can be updated to detect other traffic signs and alert the driver.

• Furthermore, integrating the system with other advanced driver assistance systems (ADAS) and vehicle-to-vehicle communication could create a more comprehensive and interconnected road safety ecosystem

VII. Literature survey

[1] T. Vesselenyi , S. Mocal, "Driver drowsiness detection using ANN image processing" 2017, IOP Publishing.

This study is regarding the possibility to develop a drowsiness detection system for car drivers based on three types of methods: EEG and EOG signal processing and driver image analysis. The EEG (Electroencephalography) method monitors the brain activity through a sensor placed on a specific part of the scalp. The EOG (Electrooculography) method tracks the eye movements by measuring the signals from the muscles which are acting on the eye. The eye image analysis can monitor the opened or the closed state of the eye. To use EOG signals acquisitioned from 3 sensors (EOG1, EOG2, EOG3). After preprocessing, four types of different signals were identified.

[2] Yuvraj Suryawanshi, Sushma Agrawal, "Driver Drowsiness Detection System based on LBP and Haar Algorithm", 2020, IEEE.

In recent years, drowsiness has emerged as a significant cause of road accidents. The primary goal of this system in mitigating road accidents is to detect driver drowsiness in real-time using video capture and face detection. Once drowsiness is detected through camera analysis, an alarm is triggered. This detection relies on monitoring head position and eye blinking as key features.

[3] Ankit Kumar, Sandeep Kumar, "Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning", 2021, IEEE.This research paper introduces a novel approach that utilizes machine learning techniques for real-time image segmentation and drowsiness detection. The study focuses on implementing an emotion detection method that utilizes Support Vector Machines (SVM) and facial expressions. Additionally, the paper showcases the application of various libraries including NumPy, OpenCV, Media Pipe, and TensorFlow for drowsiness detection.

[4] Dakshata Ramesh Dhote, Pragati Ramesh Lakde, "Automatic Speed Control of Vehicle by using RFID Technology", 2021, IJRESM.

This study focuses on implementing a natural control system for vehicles in speed-limited areas such as schools and hospital zones. This research proposes a solution for regulating vehicle speed within specified limits in restricted zones without relying on driver intervention. To achieve this, a RFID system is utilized. The vehicle is equipped with an RFID reader, while RFID tags are placed within the restricted zones.

[5] Muhammad Saif Basit, Usman Ahmad, "Driver Drowsiness Detection with Region-of-Interest Selection Based Spatio-Temporal Deep Convolutional-LSTM",2022, IEEE This study proposes an automatic region-of-interest selection based stacked spatio-temporal convolution-long short-term memory (ConvLSTM) drowsiness detection neural network for an invehicle surveillance and security system. Haar Cascade classifiers are used to select the region-ofinterest on the human face.

VIII. References

REFERENCES [1] T. Vesselenyi, S. Moca, A. Rus, T. Mitran and B. Tătaru, "Driver drowsiness detection using ANN image processing", IOP Conference Series: Materials Science and Engineering, Volume 252, 8–10 November 2017, Pitesti, Romania, Page 1-2

[2] Yuvraj Suryawanshi, Sushma Agrawal, "Driver Drowsiness Detection System based on LBP and Haar Algorithm", IEEE- 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 07-09 October 2020,

[3] Ayman Altameem, Ankit Kumar; Ramesh Chandra Poonia, Sandeep Kumar, Abdul Khader Jilani Saudaga, "Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning", IEEE Journal, Volume-9, 30 November 2021,

[4] Dakshata Ramesh Dhote, Pragati Ramesh Lakde, Anjali Sudhakar Pachare, Mukesh Ankush Meharkure, "Automatic Speed Control of Vehicle by using RFID Technology", International Journal of Research in Engineering, Science and Management, Volume 4, Issue 7, July 2021, Page 268-270.

[5] Muhammad Saif Basit, Usman Ahmad, Jameel Ahmad, Khalid Ijaz, Syed Farooq Ali, "Driver Drowsiness Detection with Region-of-Interest Selection Based Spatio-Temporal Deep ConvolutionalLSTM", IEEE-16th International Conference on Open-Source Systems and Technologies (ICOSST),

[6] https://www.analyticsvidhya.com/blog/2021/12/traffic-signs-recognition-using-cnn-and-keras-inpython/

[7]

https://docs.arduino.cc/hardware/uno-rev3

[8]
https://www.youtube.com/watch?v=SWaYRyi0TTs&t=157s