Machine Learning System for Detection of Driver Drowsiness

Aadil Yaqoob Gakhed
Student, *Dept. of ECE*
National Institute of Engineering
Mysuru, India

Aishvarya C M
Student, *Dept. of ECE*
National Institute of Engineering
Mysuru, India

Pramod H
Student, *Dept. of ECE*
National Institute of Engineering
Mysuru, India

Pramod N
Student, *Dept. of ECE*
National Institute of Engineering
Mysuru, India

Kavitha S S
Assistant Professor, *Dept. of ECE*
Mysuru, India

Abstract—Vehicle accident is one of the major causes of death in the world of which around 1.3 million people die every year. Of these, at least 20 - 30 percent are caused due to exhaustion of drivers. The involvement of fatigue as a cause is generally grossly underestimated which can be a major reason for contributing in road accidents. Majority of the accidents in recent times are caused because of distraction or drowsiness of drivers. Lack of sleep or distractions like phone calls, talking with the passengers, etc. lead to road accidents. Fatigue combined with bad infrastructure in developing countries like India is a recipe for disaster. To prevent such accidents we propose a system that alerts the driver if the driver gets diverted or feels drowsy.

Machine learning trained models for facial landmarks detection are used with the help of Python libraries in the current approach. Need for such driver drowsiness detection systems which is a real time solution for one of the major threats for drivers as well as passengers has also been recently discussed in the Indian Parliament. This whole system will be deployed on a portable hardware which includes a microcontroller, camera for live face image capture and alarms for alerting the driver which can be easily installed in any vehicle for use.

Keywords—Fatigue drivers, road accidents, Machine Learning, Python libraries, microcontroller, face and eyes detection.

I. INTRODUCTION

Drivers generally tend to overlook drowsiness while driving, but its share in the causes of accidents is high [1]. Drowsiness is neglected by everyone, there is no law to punish drowsy drivers nor any devices to detect drowsiness in use. The percentage of drowsy drivers causing accidents is increasing [2]. When a driver is sleepy all it needs is to alert the driver whereas other implementations like shutting down the engine can cause a different accident altogether.

Current drowsiness detection systems monitoring the driver’s conditions include Electroencephalography (EEG) and Electrooculography (EOG) [3]. EEG signal shows the activities of brain tissues and its functional state to the system which determines the drowsiness of the driver while EOG is the electrical signal generated by eye movements which is used to detect drowsiness conditions. Technologies like SmartEye [4] employ the movement of the driver’s eyes and the position of the driver’s head to determine the level of their fatigue.

The systems like EEG, EOG are really expensive and time consuming [5]. Owing to easy installation and low cost, non-contact methods are preferred for fatigue detection. A drowsiness detection system which uses a camera placed in front of the driver is more suitable to be used [6]. Most of the similar systems exist, which makes use of pre-trained haar- cascades. But the drawback of using such pre-trained classifiers for any detection purpose is that we never know how training of such classifiers can be done, or how the dataset has to be prepared and how to use different classifier parameters while training.

In this paper we build our own custom haar-cascade classifiers using suitable dataset. Lighting intensity and tilting of the driver's head are the problems that may occur during detection of eyes and mouth region [7]. We will be using high resolution webcam to increase the accuracy and to create an efficient system. This system can be used in any car/vehicle as the camera can be fixed on the roof without disturbing the driver's line of sight and without equipping the driver’s body with devices.

II. LITERATURE SURVEY

Author	Technical Paper	Year	Inference	Limitations
R. P. Balandon, R. F. Ahmad, M. N. Mohamad Saad and A. S. Malik	"A Review on EEG based Automatic Sleepiness Detection System for Drowsiness"	2018	Contact based systems using EEG, EOG technologies can be used to detect driver drowsiness	Uncomfortable to wear during driving and involves complex computation.
B. Rohini, D. M. Pavuluri, N. K. L.S., S. V. and N. Mohankumar	"Technology Maneuvering in Smart Vehicles for Safe Commute"	2020	Smart eye technology for drivers can be used as a measure to detect driver drowsiness	expensive and time consuming
M. Vamsi and K.P Soman	"In-Vehicle Occupancy Detection And Classification Using Machine Learning"	2020	Haar cascade generation can be done using linux commands or using AWS	Existing Haar cascade generation techniques are time consuming and outdated
Anis-Ul Islam Rafid, Amit Raha Niloy, Atiqul Islam Chowdhury, Nusrat Sharmin	"A Brief Review on Different Driver's Drowsiness Detection Techniques"	2020	Hardware drowsiness detection systems using IRIS sensors	Most of the existing systems use sensors for drowsiness detection which are found to be less efficient
G. Chen, L. Hong, J. Dong, P. Liu, J. Conradt and A. Knoll	"EDDD: Event-Based Drowsiness Driving Detection Through Facial Motion Analysis With Neuromorphic Vision Sensor"	2020	Neuromorphic vision sensors, such as Dynamic Vision Sensors (DVS) can be used for drowsiness detection	DVS do not acquire full images at a fixed frame rate affecting accuracy of the system

III. METHODOLOGY

Object Detection using Haar feature-based cascade classifiers is found to be an effective method. It is basically a machine learning based approach, which uses Haar algorithm, which involves training a cascade function from a lot of positive and negative images. It is then used to detect objects in other images. For this project we have created our own custom haar-cascade classifiers using Cascade Trainer GUI (a tool designed by Amin Ahmadi) [8] to detect face and eyes in any given image. We have used a dataset with positive and negative samples for training purpose. Cascade Trainer GUI is a program that can be used to train, test and improve cascade classifier models. It uses a graphical user interface to set the parameters and make it easy to use OpenCV tools for training and testing classifiers. By creating custom haar-cascade classifiers we are able to re-train the classifier after each detection stage, so that we can increase accuracy of the classifier in detection work.

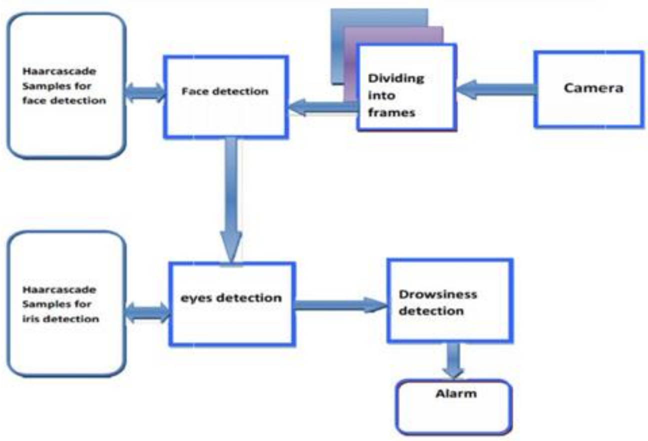

Fig. 1. Proposed System Architecture

Fig. 1. Showcases the various important blocks in the proposed system and their interaction. It can be seen that the system consists of 5 distinct modules. Functions of each module is described as follows [9]:

1. Video Acquisition: Video acquisition is achieved by using a video camera. Video acquisition is obtaining the live video feed of the automobile driver, based on which real-time drowsiness detection is done

2. Dividing into Frames: This module involves taking live video as its input and convert it into a series of frames/ images, which are then processed.

3. Face Detection and Eyes Detection: The face detection method takes one frame at a time, and in each and every frame it tries to detect the face of the automobile driver. Once face detection is done, the face data is fed to facial landmark detection model, which represents facial landmarks as 68 co-ordinate points. This is achieved by making use of a set of haar-cascade samples.

4. Facial landmark detection: The landmarks (*Fig. 2*) that we are interested in are eyes and mouth which can be detected using co-ordinate points.

5. Drowsiness Detection: After detecting the eyes and mouth of the automobile driver, the drowsiness detection function detects if the automobile driver is drowsy or not, by taking into consideration the state of the eyes and the mouth, *i.e.*, open or closed and the blink rate, yawning respectively.

MODEL TRAINING:

Cascade Trainer GUI is a program that can be used to train, test and improve cascade classifier models. Numerous parameters like iterations, pre-calculation buffer size, etc., have to be set for customizing the classifier training. After all the parameters are set, model training can be initiated by just clicking Start button. Here we will work with face detection, to generate face detecting model, which will be further used to obtain face landmarks.

Datasets required for training were taken from the website www.kaggle.com.

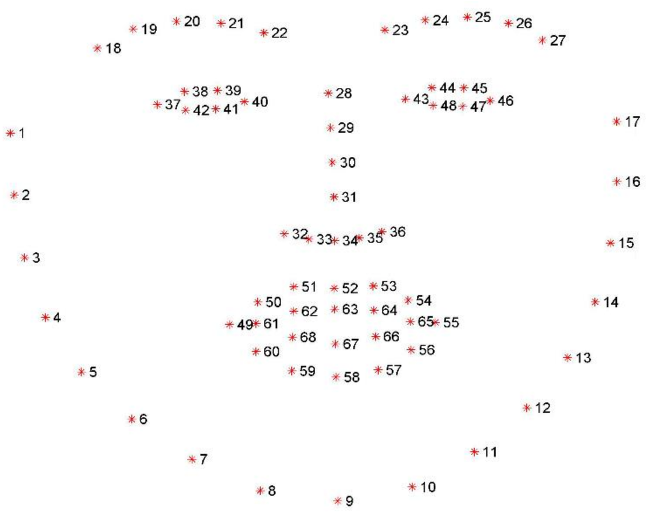

Fig. 2. Facial landmark detection

Fig. 2. Showcases the Face Landmark Detection algorithm. It is an implementation of the Ensemble of Regression Trees (ERT) algorithm, presented in 2014 by Kazemi and Sullivan [10].

UNDERSTANDING THE TRAINING OPTIONS:

The training options are a set of parameters that defines the characteristics of the trained model. These parameters can be properly fine-tuned in order to get the desired behavior of the generated model, more or less. These parameters affect the speed, accuracy and size of the model generated. Some of the most important parameters are:

Tree Depth — Tree Depth specifies the depth of the trees used in each of the cascades. Tree depth basically represents the “capacity” of the model. An optimal value (in terms of accuracy) of tree depth is 4. Instead a value of 3 is also a good tradeoff between accuracy and size of the model.

Cascade Depth — Cascade depth represents the number of cascades used to train the model. This parameter either affects the size or accuracy of a model. A good value of cascade depth is about 10-12, instead a value of 15 is also a perfect balance of a reasonable model-size and maximum accuracy.

Feature Pool Size — Feature pool size denotes the number of pixels used to generate the features for the different random trees at each cascade. Larger the amount of pixels, more robust and accurate the algorithm. But the execution will be slower. A value of 400 will achieve a great accuracy.

Num Test Splits — Num test splits represents the number of split features sampled at each node. This parameter is responsible for selecting the best features at each cascade during the training process. The parameter affects the training speed as well as the model accuracy. The default value of this parameter is 20.

GETTING THE DATA:

In order to replicate Dlib results, we have to utilize the annotations and images inside the iBug 300W dataset (available here). We are interested in training a model that is able to localize only the landmarks of the left and right eye and mouth. To do this, we have to edit the iBug training annotations by selecting only the relevant points. When the training is done, we can evaluate the model accuracy by invoking the corresponding functions. By properly fine-tuning the training options is possible to customize the training process in such a way that satisfies the constraints of the system we are developing.

This technique utilizes simple and fast features (pixel intensities differences) to directly estimate the landmark positions [11]. These estimated positions are subsequently refined with an iterative process done by the help of cascade of regressors. These regressors produce a new estimate from the previous one, trying to reduce the error of alignment of the estimated points at each of the iterations. The algorithm is fast, which in fact takes about 1–3ms (on desktop platform) to detect (align) a set of 68 landmarks on a given face.

DROWSINESS DETECTION:

Face detected by the model, is fed into Landmark detection model which represents face data using 68 points. Using these landmarks, eyes, mouth regions can be detected as well as their instantaneous movements.

Following cases are objectives of this project:

a. If eyes of drivers are closed for a threshold period of time then it is considered that driver is feeling sleepy and corresponding audio alarm is used to make the driver aware
b. If the mouth of driver remains open for the certain period of time then it is considered that driver is yawning and corresponding suggestion are provided to the driver to overcome drowsiness.
c. It driver don't keep eyes on the road then it is observed using facial landmarks and the corresponding alarm is used to make the driver aware

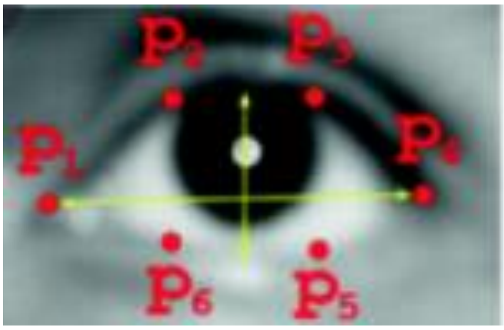

Fig. 3. Eye landmarks

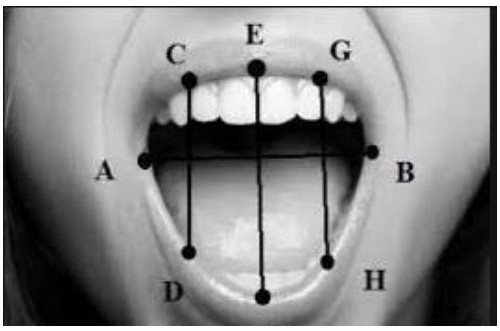

Fig. 4. Mouth landmarks

By formulating suitable expression using eye co-ordinates, we can detect if persons eye is closed or open for a duration of time. This expression is called the EAR (Eye Aspect Ratio).

$$\text{EAR} = \frac{|P2 - P6| + |P3 - P5|}{2\,|P1 - P4|}$$

If EAR < Threshold value set for a certain period of time then we can say eyes are closed, and suitable alert will be given. Similarly, by calculating the mean distance between the upper lip and lower lip using the mouth co-ordinates we can detect yawning, that distance is called as Lip distance. It is the parameter used to determine if the subject's mouth is open. If the lip distance calculated from the frame is above lip distance threshold, the subject is determined to be yawning. An alarm is raised if the subject has yawned more than the set boundary value consecutively. Small openings that in reality are construed as a result of talking, eating are ignored

IV. RESULTS

The designed system successfully detects face from an image and alerts if a person is feeling drowsy. System is also proved to be accurate and is able to detect drowsiness conditions even if the person is wearing glasses and in low light as well.

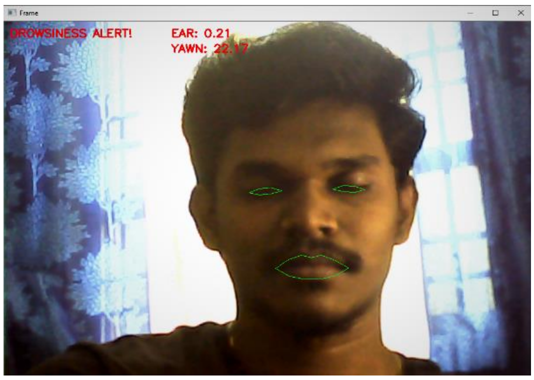

Fig. 5. Eye Closure Detection

Fig. 6. Yawn Detection

V. CONCLUSIONS AND FUTURE WORK

The approach used, successfully detects a drowsy driver and can be implemented in real time in any vehicle. It is also very adaptable, as it is a non-contact based approach. For now, software implementation of the project has been completed. In Future, All this functionality will be implemented on a hardware, with the help of raspberry pi. An audio interfacing will be used to provide audio feedback to the user and a small LED screen can also be used to display the message. With all these implementations we are building an efficient embedded system which can be deployed on any four wheeler vehicle.

VI. REFERENCES

[1] Dinesh Mohan, Geetam Tiwari, Kavi Bhalla, "Road Safety in India: Status Report 2020", IIT Delhi, 2020.

[2] Dr. Jayaprakash G. Hugar, Dr. Mirza Muhammad Naseer, Dr. Abu Waris, Muhammad Ajmal Khan, "Road Traffic Accident Research in India: A Scientometric study from 1977 to 2020", e-journal, University of Nebraska, 2020.

[3] T. Vesselenyi, "Driver Drowsiness Detection using ANN image processing", IOP conf. ser.:Mater.Sci. Eng. 252 012097, 2017.

[4] Dr. Akhtar Khalil, "Smart Eye for Driver (SED)", T-TRIID, My Smart Remote Ltd., 2018.

[5] Rodney Petrus Balandong, "A Review on EEG based Automatic Sleepiness Detection System for Drowsiness", IEEE journal, 2018.

[6] A. Subbaro, K. Sahithya, "Driver Drowsiness Detection System for Vehicle Safety", IJITEE, 2019.

[7] Ratna Yustiawati, "Analysing of different features using haar cascade classifier", IEEE journal, 2016.

[8] Dr. Prof. Suhas S. Patil, Mr. Tejas R, "Building Custom HAAR-Cascade Classifier for face Detection", IJERT, 2019.

[9] Paul Viola and Michael Jone, "Rapid Object Detection using a Boosted Cascade of Simple Features", Cambridge, 2001.

[10] Kazemi, Vahid & Sullivan, Josephine. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. 10.13140/2.1.1212.2243

[11] Kim H-W, Kim H-J, Rho S, Hwang E. Augmented EMTCNN: A Fast and Accurate Facial Landmark Detection Network. *Applied Sciences*. 2020; 10(7):2253.