International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 28 Issue 7 – JULY 2021.

Driver Drowsiness Detection

Chinmaya Bari[#1] and Kevin Tailor[#2]

[#]*Department of CS & IT, N.B.MEHTA SCIENCE COLLEGE,BORDI,India*

***Abstract*— In recent years driver fatigue is one of the major causes of vehicle accidents in the world. A direct way of measuring driver fatigue is measuring the state of the driver i.e. drowsiness. So it is very important to detect the drowsiness of the driver to save life and property. This project is aimed towards developing a prototype of drowsiness detection system. This system is a real time system which captures image continuously and measures the state of the eye according to the specified algorithm and gives warning if required. Though there are several methods for measuring the drowsiness but this approach is completely non-intrusive which does not affect the driver in any way, hence giving the exact condition of the driver. For detection of drowsiness the per closure value of eye and mouth is considered. So when the closure of eye exceeds a certain amount then the driver is identified to be sleepy. For implementing this system several OpenCv libraries are used including Haar-cascade. The entire system is implemented using Python.**

***Index Terms*—Drowsiness dection, opencv, python**

I. Introduction

Driver exhaustion is a significant variable in an expansive number of vehicle accidents. Late insights, assess that yearly 1,200 deaths and 76,000 injuries can be credited to weariness related accidents. Road Accidents in Sri Lanka cause financial losses worth around Rs.9.34 billion every year. It can be seen there are around 2,400 road accidents consistently which is one death per every four hours. It has been figured around 20% of car crashes with driver fatalities are due to driver's drowsiness. It was uncovered that driving execution quickly drop with expanded tiredness which result in making more than 20% of all vehicle accidents. Less attention heads the driver to being distracted and the likelihood of street accident goes high. Drowsiness related accidents have all the earmarks of being more serious, because of the higher speeds involved distraction and the driver being not able to take any avoiding activity, or even brake, before the accident. The improvement of innovations for recognizing or preventing tiredness of the driver is a significant test in the field of accident preventing systems. Because of the danger that that drowsiness presents on the road, strategies need to be created for checking its influences. Loss of the awareness because of the tiredness causes a few changes in the human's body and activities. These side effects and parameters empower us to effectively measure the drowsiness level. Different strategies for drowsiness identification can be partitioned into two general classifications. The techniques in the first gathering recognize the level of the tiredness focused around the physiological changes in the body. Eye status, speech properties, time interval between two yawning, head position, sitting carriage, heart rate, and brain signals are simply a couple of illustrations of the strategies in the first classification. Drowsiness additionally brings about some changes in the driving style. Techniques in the second category estimate the driver's drowsiness level by following these progressions. Steering angle, distance from the following vehicle, lateral position of the vehicle, longitudinal speed, longitudinal speeding up, and lane departure are utilized as a part of the technique of the second classification.

II. Project Analysis

Efforts reported in literature have focused on all three categories of drowsiness detection systems. Here we present a survey of literature on non intrusive detection using computer vision, have presented a detection system based on edge detection and exploiting the symmetry of facial features for extracting the eyes. The state of the eyes is determined as open or closed by taking the Hough transform for circles and comparing the intersection of the Hough transform and the edge image with a threshold. The state of drowsiness is then determined by using Percentage of Eyelid Closure(PERCLOS)- a scientifically associated measure of drowsiness associated with slow eye closure. have presented two drowsiness detection methods. In the first method they develop a camera by exploiting the fact that the retina reflects different amount of infrared light at different frequencies.Two images of the drivers face are taken at fixed wavelengths. The difference of this images is used to measure percentage eye closure. The second method although in its infancy uses a neural network to predict PERCLOS by finding the right combinations of driver performance variables. I have built a system for detecting microsleep. The system uses a remotely placed camera with near infra-red illumination to acquire the video. Haar object detection algorithm is used to detect a face. The eyes Region of interest is detected using anthropomorphic parameters. Eye closure is detected by taking ratio of the closed portion of the eye to the average height of the open portion. Under the light of what has been mentioned above, methods for drowsy detection have involved detection of face, eyes and(or) facial features. III. PROPOSED APPROACH: CONV-LSTM The problem of detecting drowsiness is that it is difficult to tell from a single frame if the person is blinking or falling asleep. In order to overcome this problem, we introduce our method Conv-LSTM, which comprises of two submodels: the CNN model for feature extraction and LSTM for interpreting the features across consecutive frames. The procedure for drowsiness detection is thus as follows: First, we extract

significant CNN features from the video frames. Then features representing the sequence of the action (Alert or a Drowsy Driver) for a certain time interval (fixed number of frames) are fed to the LSTM as an input. Finally, a softmax layer is used to predict drowsiness/alertness of the entire video sequence.[15]. Figure(1) below explains the flow of our model. Fig. 1. Flow Diagram for Conv-LSTM A. Dataset Collection Videos of eight subjects (6 males and 2 females) imitating signs of alertness and drowsiness were recorded under ambient recording conditions. During the recording of the videos, the subjects were asked to perform certain actions to imitate drowsiness such as slow eyelid closure, and droopy eyes followed by a quick recovery of head posture to imitate micro-sleep. In order to imitate alertness, the subjects were asked to gaze in different directions with/without head movement. The dataset consists of 16 Training and 3 testing videos, both containing classes: Alert-Eyes and DrowsyEyes. Videos were recorded with a CMOS front web-camera 1280x720p at 30fps with a flicker reduction of 50 Hz. B. Face ROI Detection and Eye Detection module We use Viola-Jones Haar-Feature based Cascade Classifiers for face detection. In order to avoid false positives, we first detect the face Region of Interest(fROI) and then apply eye detection on this ROI to obtain a rectangular localized patch containing a pair of eyes. After detecting the face and eyes in the first frame, we track them using CAMShift (Continuously Adaptive Mean-shift). Below figures demonstrate detection of closed as well as open eyes. Fig. 2. Alert-Eye detection Fig. 3. Drowsy-Eye detection C. Convolutional Neural Network (Inception-v3) module We manually created an image dataset for feature extraction. Two classes were made with approximately 120 images each for Alert-Eyes and Drowsy-Eyes. To extract significant visual features from these images, we use Convolutional Neural Networks (CNNs), which are state-of-the-art for image classification and feature extraction.We adapted a pretrained model, Inception-v3[12], which is trained on the Image-Net Dataset comprising of 1000 classes for Large Scale Visual Recognition Challenge(2012)[10]. Using transfer learning we retrain the final layer of this model on our dataset with Tensorflow[11]. At the completion of 4000 training steps, our model reported an accuracy of 96.5% on the validation set. Then, we ran each frame(image) of every video through Inception model and saved the output from the final pooling layer (pool-3:0). This results in a 2048-Dimensional vector of features, which we passed to the sequential neural models. Finally, we convert these extracted features into sequences of extracted features

III. ALGORITHM

There are various algorithms used for facial recognition. Some of them are as follows:

- Eigen faces
- Fisher faces
- Local binary patterns histograms

A. EIGEN FACES

This method is a statistical plan. The characteristic which influences the images is derived by this algorithm. The whole recognition method will depend on the training database that will be provided. The images from two different classes are not treated individually.

B. FISHER FACES

Fisher faces algorithm also follows a progressive approach just like the Eigen faces. This method is an alteration of Eigen faces so it uses the same principal Components Analysis. The major conversion is that the fisher faces considers the classes. As mentioned previously, the Eigen faces does not differentiate between the two pictures from two differed classes while training. The total average affects each picture. A Fisher face employs Linear Discriminant Analysis for distinguishing between pictures from a different class.

C. LOCAL BINARY PATTERNS HISTOGRAMS

This method needs the gray scale pictures for dealing with the training part. This algorithm in comparison to other algorithms is not a holistic approach.

1) PARAMETERS:

LBPH uses the following parameters:

Radius:

Generally 1 is set as a radius for the circular local binary pattern which denotes the radius around the central pixel.

Neighbors:

The number of sample points surrounding the central pixel which is generally 8.The computational cost will increase with increase in number of sample points.

Grid X:

The number of cells along the horizontal direction is represented as Grid X. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

Grid Y:

The number of cells along the vertical direction is represented as Grid Y. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

D. ALGORITHM TRAINING:

For the training purpose of the dataset of the facial images of the people to be recognized along with the unique ID is required so that the presented approach will utilize the provided information for perceiving an input image and providing the output. Same images require same ID.

E. COMPUTATION OF THE ALGORITHM:

The intermediate image with improved facial characteristics which corresponds to the original image is created in the first step. Based on the parameters provided, sliding window theory is used in order to achieve so.

Facial image is converted into gray scale. A 3x3 pixels window is taken which can also be expressed as a 3x3 matrix which contains the intensity of each pixel (0-255). After this

we consider the central value of the matrix which we take as the threshold. This value defines the new values obtained from the 8 neighbors. A new binary value is set for each neighbor of the central value. For the values equal to or greater than the threshold value 1 will be the output otherwise 0 will be the output. Only binary values will be present in the matrix and the concatenation is performed at each position to get new values at each position. Then the conversion of this binary value into a decimal value is done which is made the central value of the matrix. It is a pixel of the actual image. As the process is completed, we get a new image which serves as the better characteristics of the original image.

F. EXTRACTION OF HISTOGRAM:

The image obtained in the previous step uses the Grid X and Grid Y parameters and the image is split into multiple grids. Based on the image the histogram can be extracted as below:

The image is in gray scale and each histogram will consist of only 256 positions (0-255) which symbolizes the existences of each pixel intensity.

After this each histogram is created and a new and bigger histogram is done. Let us suppose that there are 8x8 grids, then there will be 16.384 positions in total in the final histogram. Ultimately the histogram signifies the features of the actual image.

G. THE FACE RECOGNITION:

The training of the algorithm is done. For finding the image which is same as the input image, the two histograms are compared and the image corresponding to the nearest histogram is returned. Different approaches are used for the calculation of distance between the two histograms. Here we use the Euclidean distance based on the formula:

Hence the result of this method is the ID of the image which has the nearest histogram. It should return the distance calculated in the form of 'confidence'. Then the threshold and the 'confidence' can be used to automatically evaluate if the image is correctly recognized. If the confidence is less than the given threshold value, it implies that the image has been well recognized by the algorithm

H. ADVANTAGES OF USING LBPH ALGORITHM:

1. It is one of the simplest algorithms for face recognition.
2. The local features of the images can be characterized by this algorithm.
3. Using this algorithm, considerable results can be obtained.
4. Open CV library is used to implement LBPH algorithm.

IV. BLOCK DIAGRAM

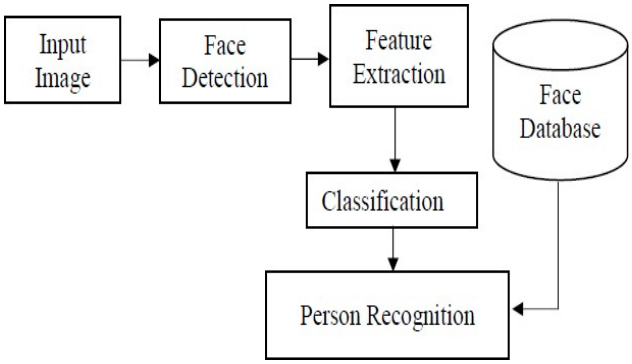

Figure 1: Block Diagram

V. IMPLEMENTATION RESULTS

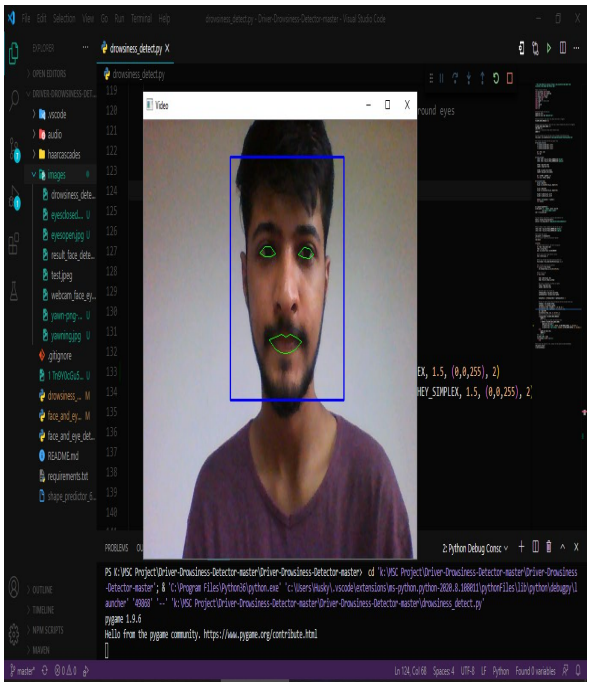

Figure 2: Our system is detecting the facial point such as face, eyes and mouth.

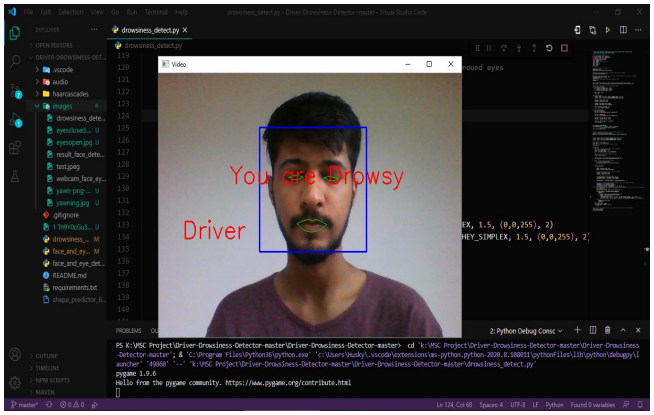

Figure 3: System will alarm the driver if the closure value of eye is below threshold value. Resulting in alarm which can aware the driver that they are drowsy.

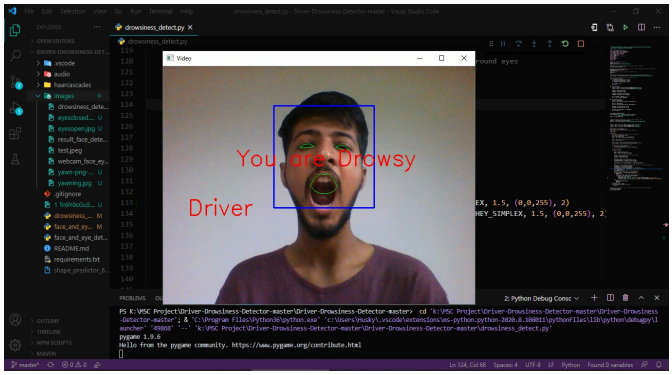

Figure 4: System will alarm the driver if the closure value of mouth is below threshold value. If the driver yawns then we will inform him to take some rest. Resulting in alarm which can aware the driver that they are drowsy.

VI. CONCLUSION

This project proposes a drowsiness and intrusion detection system based on driver behavior. The role of the system is to detect facial landmark from images that are collected while the person is driving the vehicle by a camera module attached to the vehicle and deliver the obtained data to the trained model to identify the driver's state. Once the collected data is detected to be showing signs of drowsiness the person will be alerted using the speakers in the vehicle so that the person can stop the vehicle to avoid any accidents due to his drowsy state. The system also includes GPS tracking of car and alerts on mobile app regarding car movement. The admin of the system can monitor the sessions of the drivers and incase of any events such as person ignoring the drowsiness alerts or accidents can take proper actions according to situation. According to the experimental results, the size of the used model is small while having the accuracy rate of 81 percentage. Hence, it can be integrated into advanced driver-assistance systems, the Driver drowsiness detection system, and mobile applications. However, there is still space for the performance improvement. The further work will focus on detecting the distraction and yawning of the driver. Also accelerometer is to be incorporated to track the speed of the car, condition of different faces. So we need to make sure that only the driver face come within the range of the camera. Also the speed of detection reduces because of operation on multiple faces.

REFERENCES

[1] COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.
[2] USE OF THE HOUGH TRANSFORMATION TO DETECT LINES AND CURVES IN PICTURE; R. DUDA AND P. E. HART.
[3] JAIN, "FACE DETECTION IN COLOR IMAGES; R. L. HSU, M. ABDEL-MOTTALEB, AND A. K. JAIN.
[4] OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI.
[5] http://ncrb.gov.in/StatPublications/ADSI/ADSI2015/chapter1A%20traffic %20accidents.pdf
[6] http://www.jotr.in/text.asp?2013/6/1/1/118718
[7] http://dlib.net/face_landmark_detection_ex.cpp.html
[8] https://arxiv.org/abs/1904.07312
[9] https://pypi.org/project/opencv-python/
[10] https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e
[11] https://neurohive.io/en/popular-networks/vgg16/