A Project Report

on

DROWSY DRIVER ALERT SYSTEM

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. N. Partheeban
Assistant Professor
Department of Computer Science and Engineering

Submitted By

19SCSE1010908 – HIMANSHU JOSHI

19SCSE1010531 – LALIT KUMAR

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **" DROWSY DRIVER ALERT SYSTEM "** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **SEPTEMBER-2021 to DECEMBER-2021**, under the supervision of **Mr. N. Partheeban, Assistant Professor**, **Department of Computer Scienceand Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

19SCSE1010908 – HIMANSHU JOSHI

19SCSE1010531 – LALIT KUMAR

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

Mr.N. Partheeban,

(Assistant Professor)

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **19SCSE1010908 – HIMANSHU JOSHI, 19SCSE1010531 – LALIT KUMAR** has been held on _____and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

ABSTRACT

Drowsy driving is the risky mix of driving and tiredness or weariness. This usually happens when a driver has not slept enough, but it can also happen because of untreated sleep, disorders, medications, drinking alcohol, or shift work. An expected 1 out of 25 grown-up drivers report having nodded off while driving in the past 30 days. Even if the driver has not consumed alcohol, studies have shown the effects of driving while sleep-deprived are similar to those of drunk driving. These effects include impaired attention and coordination, slower reaction time, and poor judgement. So, to prevent these accidents, we'll build a system using Python, OpenCV, and Keras which will alert the driver when he feels sleepy. Drowsy detection is a security innovation that can forestall accidents that are brought about by drivers who fell asleep while driving.

In this Python project, we will utilize OpenCV for gathering the pictures from webcam and feed them into a Deep Learning model which will group whether the individual's eyes are 'Open' or 'Shut'.

Keywords : drowsy driving, sleep, model, OpenCV, detection

Table of Contents

Introduction

With the expanding speed of life and work the individuals have begun compromising with the thing they need most, to proficiently work in a given circumstance i.e. satisfactory rest and rest to be dynamic while doing an assignment. Driver's laziness is an extremely mindful thing and has as of now prompted a great deal of setbacksWith the assistance of current innovation and continuous filtering frameworks utilizing cameras we can forestall significant accidents out and about by alarming vehicle driver who is feeling sluggish through a drowsy driver alert system.

A driver who nods off at the worst possible time fails to keep a grip on the vehicle, an activity which frequently brings about an accident with either another vehicle or fixed articles. To forestall these overwhelming mishaps, the condition of sluggishness of the driver ought to be observed. The accompanying measures have been utilized generally for observing languor:

- Vehicle-based measures : A number of measurements, including deviations from path position, development of the guiding wheel, strain on the speed increase pedal, and so on, are continually observed and any adjustment of these that passes a predetermined boundary shows an altogether expanded likelihood that the driver is sluggish .

- Social measures : The conduct of the driver, including yawning, eye conclusion, eye squinting, head present, and so on, is observed through a camera and the driver is cautioned if any of these languor manifestations are recognized.

- Physiological measures : The relationship between's physiological signs (electrocardiogram (ECG), electromyogram (EMG), electrooculogram (EoG) and electroencephalogram (EEG)) and driver languor has been contemplated by numerous specialists.

The aim of this program will be to design a framework which will, with nearness, recognize the condition of the driver's eyes for example regardless of whether they are open or shut continuously. The guideline behind this relationship is that the condition of somebody's eyes can demonstrate the indications of their weakness and furthermore prevent accidents brought about by it by sounding an alert, preferably loud enough to take the waterway back to detects and keep away from the impending accident. This works by first setting up the area containing the eyes, all together to do as such first the entire face is searched and set up then a Region of Interest is selected i.e. the area containing the eyes. Then, at that point, the system decides if the eyes are shut or open.

FORMULATION

To prevent the accidents, we will build a system using Python, OpenCV and Keras which will alert the driver when he feels sleepy.

To make this model, we've taken our dataset from the Kaggle.com and isolated them into their particular marks Open or Closed. The information was physically cleaned by eliminating the undesirable pictures which were excessive for building the model. The information contains around 7000 pictures of people groups eyes under various lighting conditions. In the wake of preparing the model on our dataset, we have connected the last loads and model design record models/cnnCat2.h5.

Now, you can use this model to classify if a person's eye is open or closed.

The stage-wise method of our model is :

Stage 1 : Take picture as input from a camera.

Stage 2 : Detect the face in the picture and make a Region of Interest (ROI).

Stage 3 : Detect the eyes from ROI and put it into the classifier.

Stage 4 : Classifier will classify whether eyes are open or shut.

Stage 5 : Calculate score to check whether the individual is sleepy.

Tools and Technology Used

The Computer's Vision

PC's vision is the difference in data from a still, or camcorder into either a depiction or another decision. Every single such changes are performed to accomplish a specific objective. A Computer acquires a cross segment of numbers from a camera or from the circle, and that's all there is to it. Generally, there is no worked in model affirmation or customized control of focus and hole, no cross-relationship with significant stretches of involvement. By and large, vision systems are still sensibly simple.

- **OpenCV (face and eye detection)** : OpenCV is the huge open-source library for the PC vision, machine learning, and image processing and presently it assumes a significant part progressively activity which is vital in the present frameworks. By utilizing it, one can process pictures and recordings to identify items, faces, or in any event, handwriting of a human. At the point when it coordinated with different libraries, for example, NumPy, python is equipped for handling the OpenCV array structure for analysis. To Identify image pattern and its different highlights we use vector space and perform numerical procedure on these elements.

  OpenCV is coded with streamlined C and can take work with multicore processors. In the event that we want dynamically customize improvement using Intel models. These involve low-level timetables in various algorithmic locales which are smoothed out. OpenCV along these lines utilizes the IPP library, at runtime assuming that library is presented.

  Why use OpenCV ?

  i. Explicit OpenCV was planned for picture dealing. Each design and construction of the data is imagined in the Picture Processing Plan. Then, at that point, Matlab, is exceptionally ordinary.
  ii. Speedy Matlab is just excessively moderate.
  iii. Productive Matlab utilizes an unnecessary measure of framework resources. With OpenCV, we can remove so a lot as 10mb RAM for continuous application.

- **Keras (to build the classification model)** : Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

- **TensorFlow (keras uses TensorFlow as backend)** : TensorFlow is an end to end open source stage for machine learning. It has a far reaching, adaptable environment of instruments, libraries and community resources that allows specialists to push the cutting edge in ML and engineers effectively assemble and send ML controlled applications.

  Why use TensorFlow ?

  i. Easy model building
  ii. Robust ML production anywhere
  iii. Powerful experimentation for research

- **Pygame (to play alarm sound) :** Sounds generally come in two major forms: Either "ambient" noise or as results of player actions. With PyGame, you get two choices: Music or Sounds. Music will just play in the background when you call it to, and sounds will play at any time you call them to play.

LITERATURE SURVEY

The danger, risk, and regularly grievous aftereffects of drowsy driving are disturbing. Drowsy driving is the risky mix of driving and sluggishness or weariness. This typically happens when a driver has not rested enough, yet it can likewise happen due to untreated rest issues, drugs, drinking liquor, or shift work.

Nobody knows the specific second when rest comes over their body. Nodding off at the worst possible time is obviously perilous yet being drowsy influences your capacity to drive securely regardless of whether you don't nod off.

Anybody can encounter the impacts of inadequate rest, yet certain people are more inclined to lazy driving. These include :

• Business drivers who work tow trucks, semi-trucks, trailers, and other enormous vehicles.

• Shift laborers who work around evening time or whose movements are somewhat long.

• Individuals with rest problems that cause rest misfortune as well as rest confused relaxing.

• The people who take meds that cause sleepiness.

Furthermore, certain driving conditions represent a higher danger of languid driving-related mishaps. A large portion of these impacts happen either in the late evening or between the long periods of 12 am and 6 am. Both of these occasions match with regular energy plunges constrained by your circadian clock, an inside timekeeping framework that directs rest, craving, and other real cycles.

RELATED WORK

In [1], Manu (2016) depicts a useful technique for drowsiness acknowledgment by three portrayed stages. These three phases are facial features disclosure using Viola Jones, the eye following and yawning revelation. At the point when the face is recognized, the system is made light invariant by dividing the skin part alone and considering only the chromatic portions to excuse most of the non-face picture establishments reliant upon skin concealing. The accompanying of eyes and yawning acknowledgment are done by relationship coefficient design organizing.

In [2], Belal et al. (2013) proposed a module for Advanced Driver Assistance System (ADAS) to diminish the amount of disasters due to driver's shortcoming and thus increase the transportation prosperity. This system oversees modified driver sluggishness distinguishing proof ward on visual information and Artificial Intelligence.

In [3], Jang et al. (2018) presented an original drowsiness recognizable proof computation using a camera near the dashboard. The proposed computation recognizes the driver's face in the image and checks the achievements in the face region. To perceive the face, the proposed computation uses an AdaBoost classifier dependent on the Modified Census Transform features.

In [4], Feng et al. (2019) inferred a steady driving drowsiness acknowledgment computation that contemplates the singular differentiations of driver. A significant fell convolutional neural framework was worked to perceive the face region, which avoids the issue of powerless accuracy brought about by counterfeit component extraction. In view of the Dlib tool kit, the achievements of front facing driver facial in a packaging are found. As demonstrated by the eye's achievements, another limit, called Eyes Aspect Ratio, is familiar with survey the sluggishness of driver.

In [5], Shibo and Xiaojie (2013) worked on the Histograms of Oriented Gradients which are used to address the edge of the information of pictures. In order to follow logically, they utilized establishment derivation area with Histograms of Oriented Inclinations, which achieves the vital precision and satisfies ceaseless interest.

In [6], Bruno et al. (2019) fostered a drowsiness level acknowledgment structure that directions picture taking care of with the usage of Raspberry Pi3, OpenCV library, and sensors.

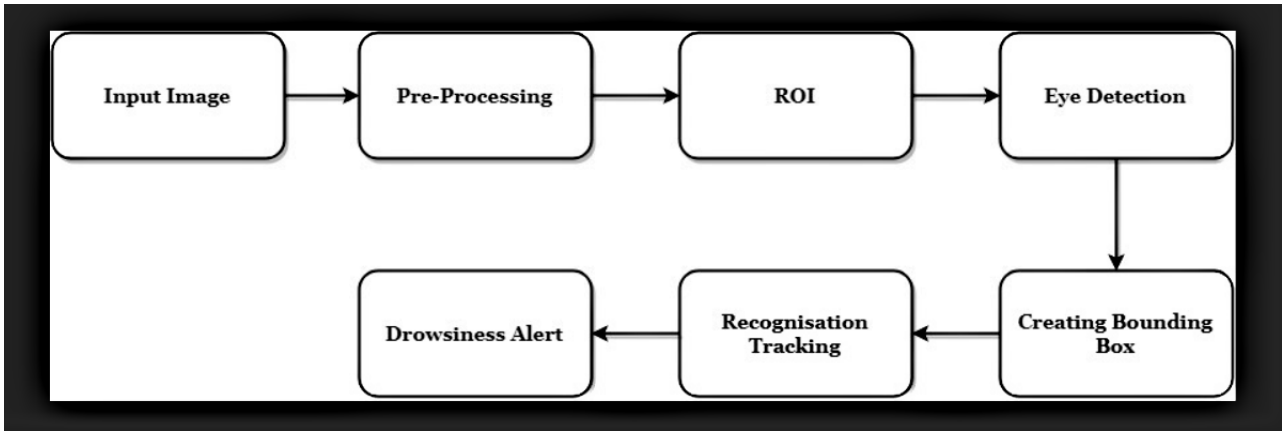
In [7], Anirban et al. (2018) introduced a mobile phone based structure for the sleepiness in vehicle drivers. The proposed framework uses three-stage tiredness acknowledgment. The primary stage uses the degree of eyelid end (PERCLOS) gained through pictures got by the front camera with a changed eye state request method. The structure uses near infrared lighting for edifying the quintessence of the driver during night-driving. The ensuing development utilizes the voiced to the unvoiced extent got from the talk data from the mouthpiece. A last check stage is used as a contact response inside a predefined freedom to announce the driver as drained and thus stable an alert.

SCOPE OF OUR WORK

There are different ways to deal with further develop the road security for a vehicle driver. It should be seen that one of the massive standard methodologies presented in past sensible investigates depends in the improvement of state of the art driver help systems. These prosperity structures license to lessen road incidents and outfit better cooperation and responsibility with a driver. Some typical occasions of driver security progressions for this kind of structures are vehicle sway avoiding system, way keep accomplice, driver laziness and interference seeing what's more, alerted. General usage of such systems can be portrayed as a certain plan of consecutive orders thusly: noticing driver lead, state of the vehicle or road situation by using particular intrinsic aide devices, including short and long reach radars, lasers, lidars, video transfer cameras to see the ecological factors; unending examination of readings from sensors and choosing unsafe conditions while driving; alerted driver about seen risky in-hotel and road conditions; and assuming responsibility for the vehicle on the off chance that driver reaction isn't satisfactory or missing. Right presently, driver security systems overwhelmingly rely upon data assembled from different in-vehicle sensors.

PROJECT DESIGN

Most of them in a few ways relate to highlights of the eye (ordinarily reflections from the eye) inside a video picture of the driver. The first point of this venture was to utilize the retinal reflection as a implies to finding the eyes on the confront, and after that utilizing the nonappearance of this reflection as a way of identifying when the eyes are closed. Applying this calculation on sequential video outlines may help within the calculation of eye closure period. Eye closure period for lazy drivers are longer than ordinary blinking. It is additionally exceptionally small longer time may result in extreme crash. So, we'll caution the driver immediately as closed eye is identified.

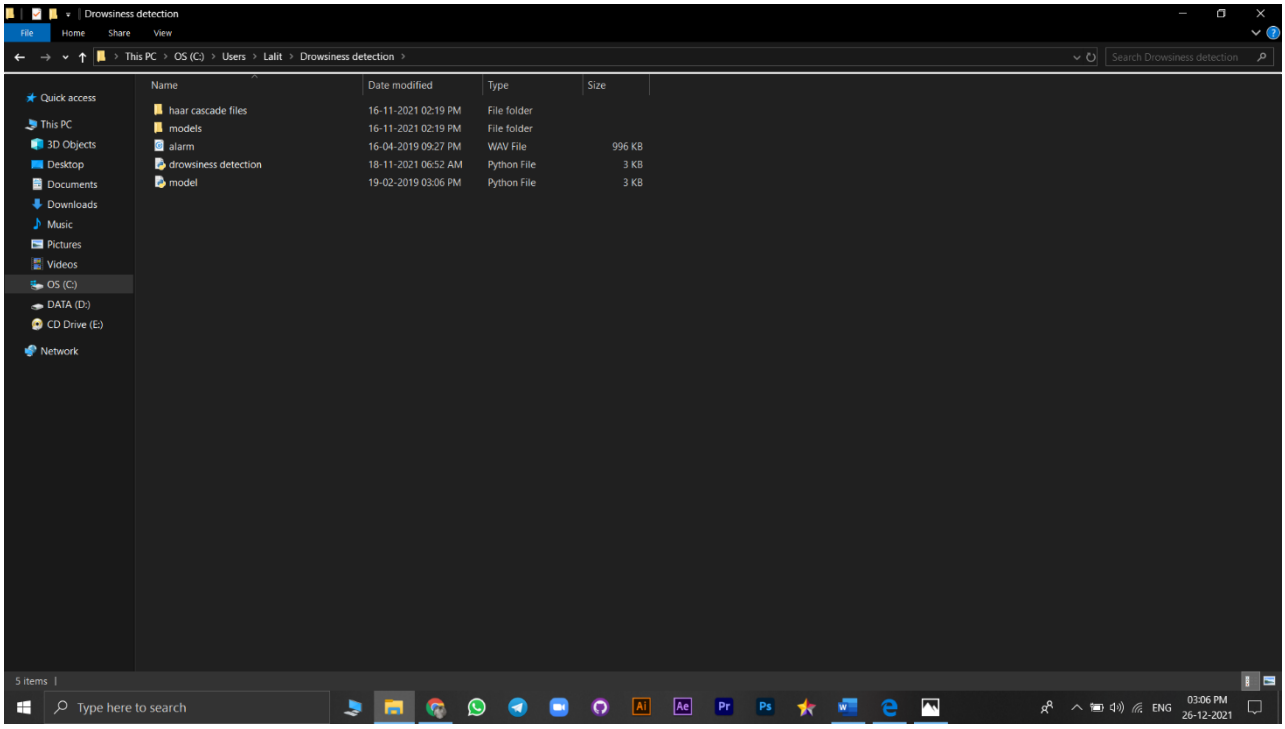


DATASET

The data was manually cleaned by removing the unwanted images which were not necessary for building the model. The data comprises around 7000 images of people's eyes under different lighting conditions. After training the model on our dataset, we have attached the final weights and model architecture file "models/cnnCat2.h5".

Now, you can use this model to classify if a person's eye is open or closed.

FILES



The haar cascade files comprises of the xml documents that are expected to distinguish objects from the picture. For our situation, we are distinguishing the face and eyes of the individual.

• The models envelope contains our model record cnnCat2.h5 which was prepared on convolutional neural organizations.

• We have a sound bite alarm.wav which is played when the individual is feeling lazy.

• Model.py record contains the program through which we assembled our order model via preparing on our dataset. You could see the execution of convolutional neural organization in this document.

• Sleepiness detection.py is the primary document of our task. To begin the recognition system, we need to run this record.

WORKING

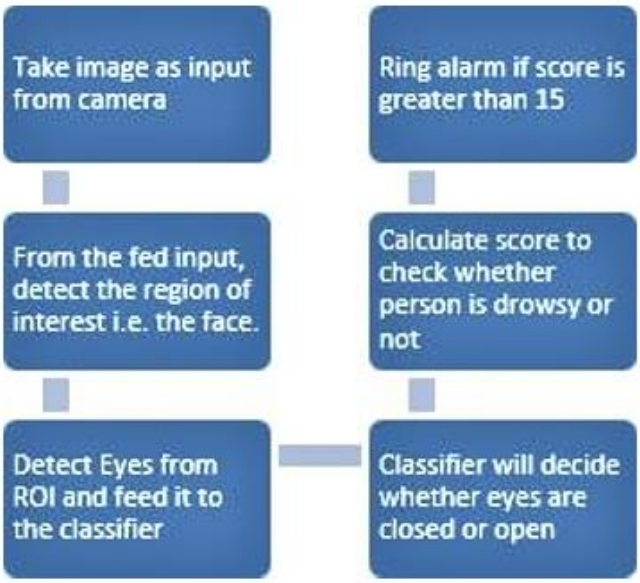


Fig. 1 - Systematic Flowchart of the algorithm

Algorithm

Step 1 : Take picture as input from a camera.

With the assistance of a camera, each frame is caught persistently, for this, it is ideal to make and infinite circle by the method given by OpenCV i.e., cv2.VideoCapture(0), each frame is caught and put away in a variable named outline.

Step 2 : Detect the face in the picture and make a Region of Interest (ROI).

To detect the face in the picture, we need to convert the picture into grayscale as the OpenCV algorithm for object location takes gray pictures in the input. We needn't bother with colour data to detect the objects. We will utilize haar cascade classifier to recognize faces. This line is utilized to set our classifier face = cv2.CascadeClassifier(' way to our haar course xml record'). Then, at that point, we play out the detection utilizing faces = face.detectMultiScale(gray). It returns an array with x,y directions, and height, the width of the boundary box of the item. Presently we can emphasize over the faces and draw boundary boxes for each face.

```
for (x,y,w,h) in faces:
cv2.rectangle(frame, (x,y), (x+w, y+h), (100,100,100), 1
)
```

Step 3 : Detect the eyes from ROI and put it into the classifier.

Presently, a similar strategy as above is applied to separate eyes from the face. Cascade Classifier is utilized to initially recognize left eye and afterward the right eye utilizing left_eye = leye.detectMultiScale(gray). The data of left eye is taken care of to l_eye variable and the data of right eye is taken care of to r_eye variable.

```
l_eye = frame[ y : y+h, x : x+w ]
```

The information of l_eye is taken care of to CNN classifier which will secure whether or not the eye is shut, comparable interaction is applied to r_eye.

Step 4 : Classifier will classify whether eyes are open or shut.

We are using CNN classifier for predicting the eye status. To take care of our picture into the model, we need to perform certain actions that the model necessities the right aspects to begin with. To start with, we convert the shading picture into grayscale utilizing r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY). Then, at that point, we resize the picture to 24*24 pixels as our model was prepared on 24*24 pixel pictures cv2.resize(r_eye, (24,24)). We standardize our information for better combination r_eye = r_eye/255 (All qualities will be between 0-1). Grow the aspects to take care of into our classifier. We stacked our model utilizing model = load_model('models/cnnCat2.h5') . Presently we predict each eye with our model lpred = model.predict_classes(l_eye). If the worth of lpred[0] = 1, it expresses that eyes are open, assuming worth of lpred[0] = 0, it expresses that eyes are shut.

Step 5 : Calculate score to check whether the individual is sleepy.

The score is fundamentally a worth we will use to decide how long the individual has shut his eyes. So assuming that both the eyes are shut, we will continue to build score and when eyes are open, we decline the score. We are drawing the result on the screen utilizing cv2.putText() function which will show constant status of the individual.

An threeshold is characterized for instance assuming score becomes more prominent than 15 that implies the individual's eyes are shut for a long period of time. This is the point at which we beep the alarm utilizing sound.play()

```
1. cv2.putText(frame, “Open”, (10, height-20), font, 1,
   (255,255,255), 1, cv2.LINE_AA )
```

SOURCE CODE

```
import cv2
import os
from keras.models import load_model
import numpy as np
from pygame import mixer
import time
mixer.init()
sound = mixer.Sound('alarm.wav')
face = cv2.CascadeClassifier('haar cascade files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier('haar cascade files\haarcascade_lefteye_2splits.xml')
reye = cv2.CascadeClassifier('haar cascade files\haarcascade_righteye_2splits.xml')
lbl=['Close','Open']
model = load_model('models/cnncat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]
while(True):
ret, frame = cap.read()
height,width = frame.shape[:2]
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
left_eye = leye.detectMultiScale(gray)
right_eye = reye.detectMultiScale(gray)
cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) , thickness=cv2.FILLED )
for (x,y,w,h) in faces:
cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )
for (x,y,w,h) in right_eye:
r_eye=frame[y:y+h,x:x+w]
count=count+1
r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
r_eye = cv2.resize(r_eye,(24,24))
r_eye= r_eye/255
r_eye= r_eye.reshape(24,24,-1)
r_eye = np.expand_dims(r_eye,axis=0)
rpred = model.predict_classes(r_eye)
if(rpred[0]==1):
```

```
lbl='Open'
if(rpred[0]==0):
lbl='Closed'
break
for (x,y,w,h) in left_eye:
l_eye=frame[y:y+h,x:x+w]
count=count+1
l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
l_eye = cv2.resize(l_eye,(24,24))
l_eye= l_eye/255
l_eye=l_eye.reshape(24,24,-1)
l_eye = np.expand_dims(l_eye,axis=0)
lpred = model.predict_classes(l_eye)
if(lpred[0]==1):
lbl='Open'
if(lpred[0]==0):
lbl='Closed'
break
if(rpred[0]==0 and lpred[0]==0):
score=score+1
cv2.putText(frame,"Closed",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
# if(rpred[0]==1 or lpred[0]==1):
else:
score=score-1
cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
if(score<0):
score=0
cv2.putText(frame,'Score:'+str(score),(100,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
if(score>15):
#person is feeling sleepy so we beep the alarm
cv2.imwrite(os.path.join(path,'image.jpg'),frame)
try:
sound.play()
except: # isplaying = False
pass
if(thicc<16):
thicc= thicc+2
else:
thicc=thicc-2
if(thicc<2):
thicc=2
cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
```

```
cv2.imshow('frame',frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
```

ANALYSIS

Convolutional Neural Network

In deep learning, CNN is broadly used to perform picture classification, object detection, picture recognition, face recognition and a few different errands identified with image handling. A CNN normally comprises of an information layer, a result layer and hidden layers. The quantity of hidden layers differs as per the complexity of the model. For the most part, in deep learning CNN models, each casing is handled by means of a progression of convolutional layers with channels then they are handled through enactment capacities to play out an arrangement of the casings with probabilistic qualities, in genuine practice, these vales lie somewhere in the range of 0 and 1.

In the proposed model, a similar sort of convolutional neural network is utilized. A few test runs were made utilizing different hidden layers until the right number of hidden layers was observed which gave the most accuracy. Initial, two secret layers were picked which gave the exactness of around half, which isn't exactly good. Then, at that point, to work on the precision, one more hidden layer was added which expanded the accuracy nearly twice. It is clear from the graphs introduced below :

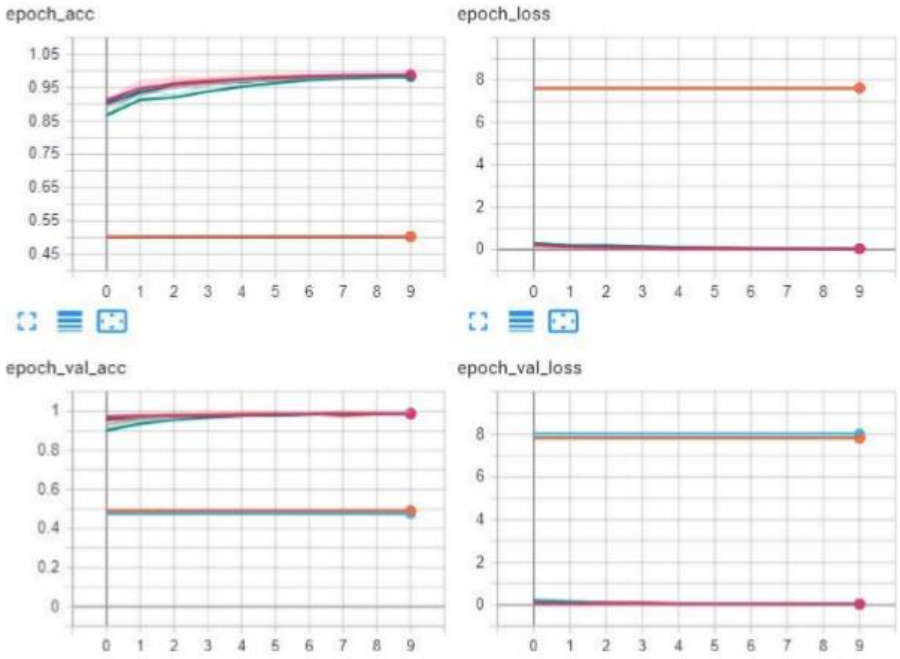


Fig. Report of analysis using 3 convolutional and 2 dense layers

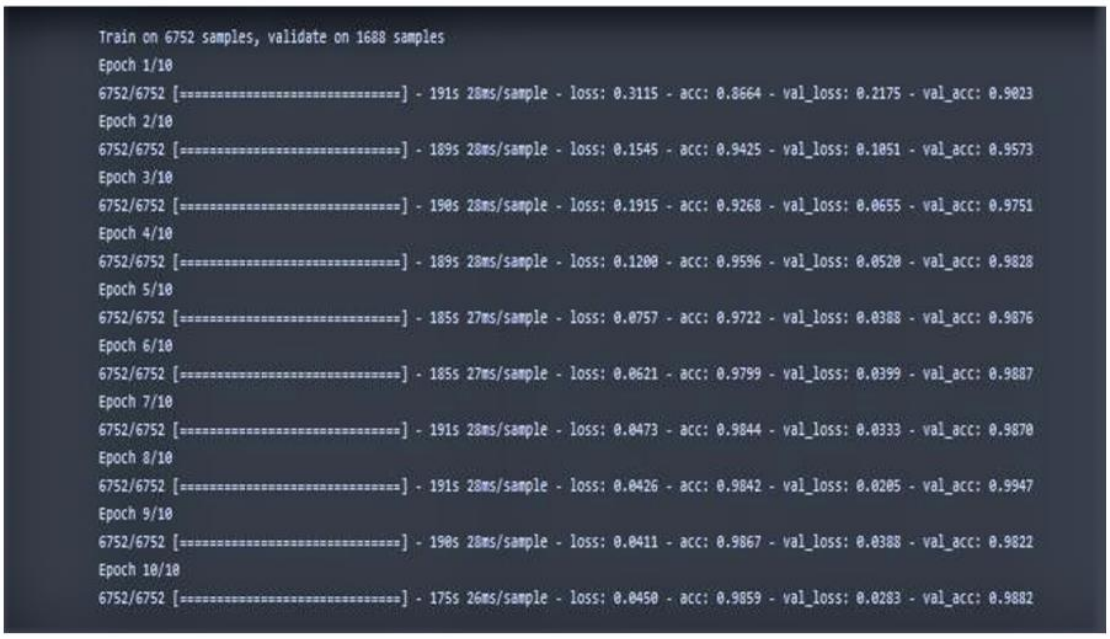


Fig. Report of data fitting

After choosing the ideal number of convolutional layers, the next stage is to choose the most fitting optimiser. A few optimisers to be specific Adagrad(orange), adam(blue), nadam(red) adamax(light blue) were attempted and their relating accu-racies were noted. Adamax performed better and thus, was utilized. The following are the diagrams for the performance of Adamax Optimiser :

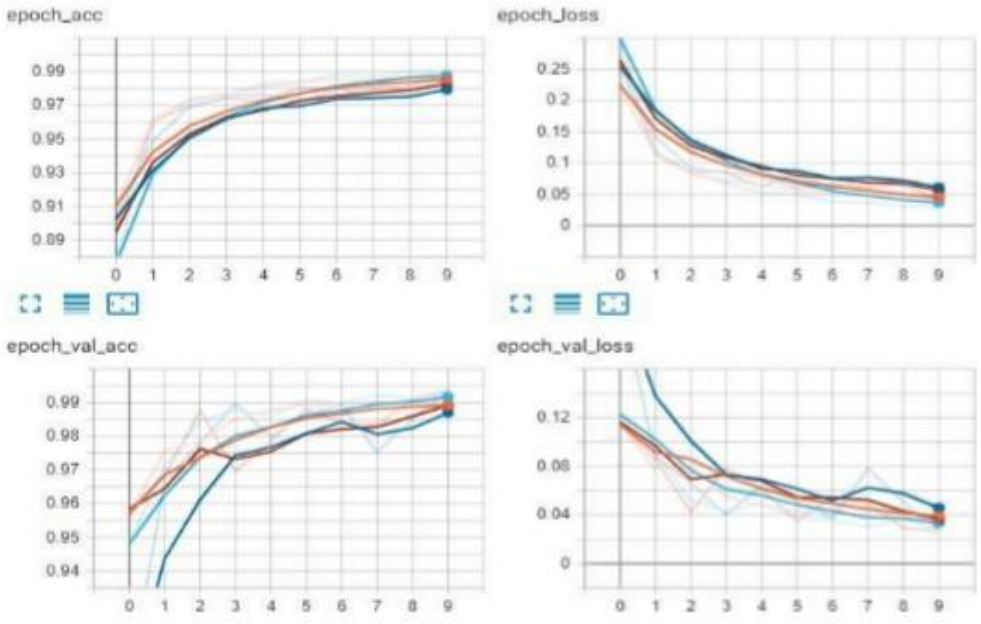


Fig. Graphs for the performance of Adamax Optimiser

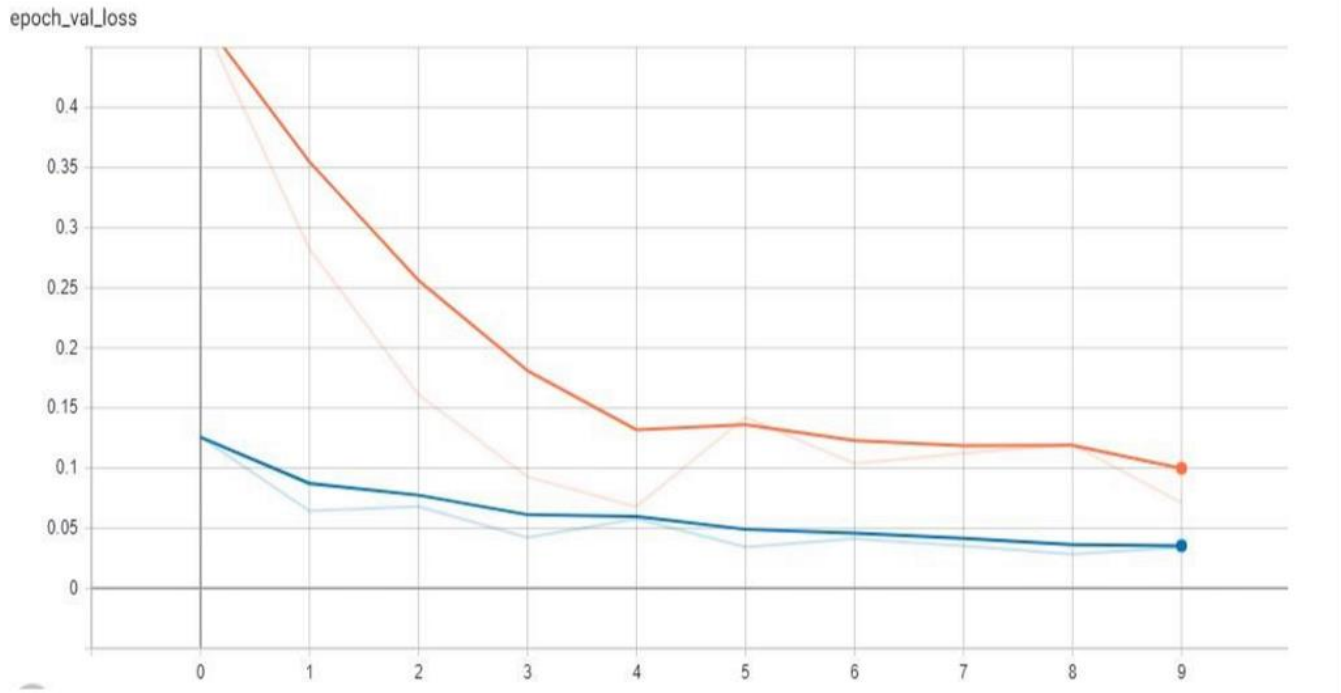


Fig. Validation loss between VGG vs The Proposed Model
(Blue line depicts proposed model and Red line depicts VGG)

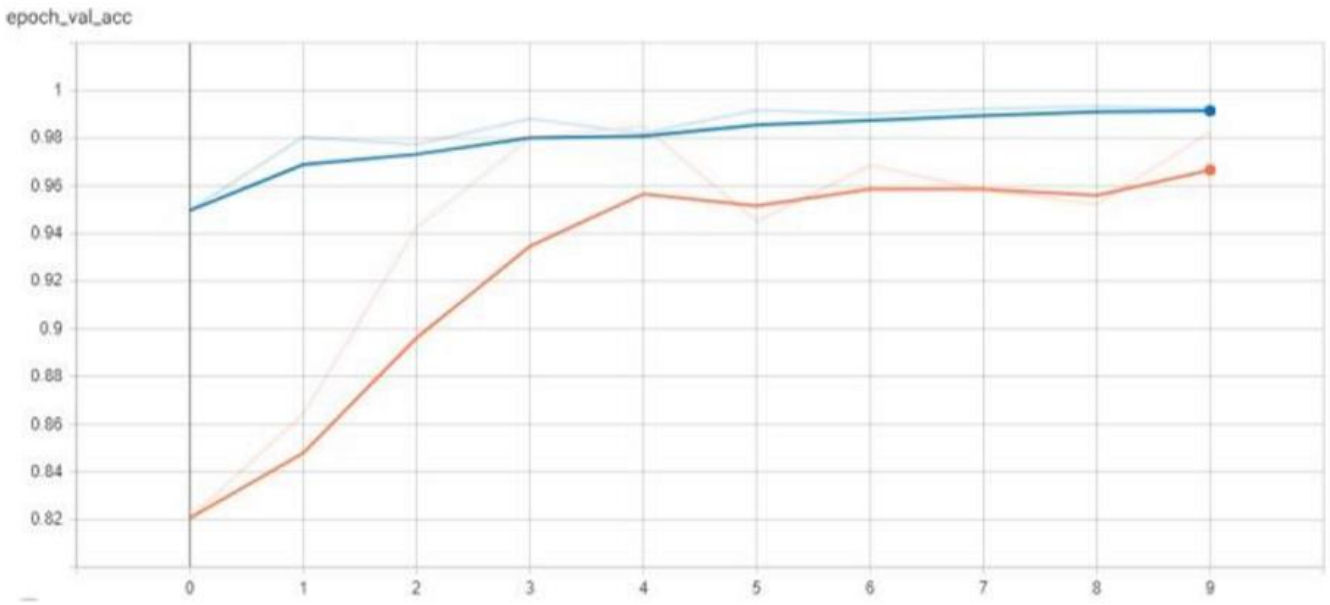


Fig. Validation Accuracy between VGG vs The Proposed Model
(Blue line depicts proposed model and Red line depicts VGG)

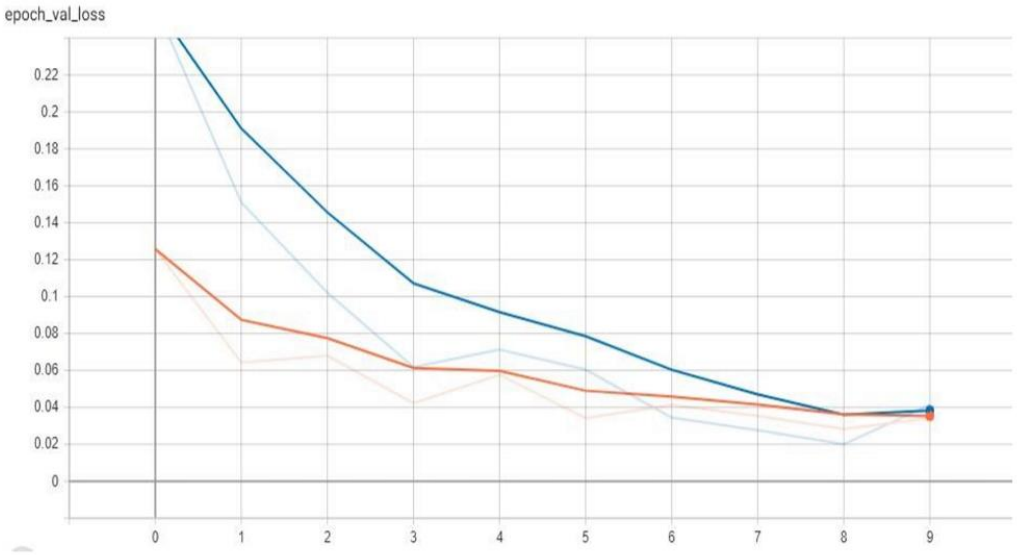


Fig. Validation Loss between LeNet vs The Proposed Model
(Blue line depicts proposed model and Red line depicts LeNet)

Thus, the obtained model was compared against the pre-existing architectures i.e., VGG and LeNet and the graphs were plotted against the model proposed in this paper. The proposed model performed better than both the models as is evident from above graph.

RESULT

The driver anomaly observing framework created is able of identifying laziness, intoxicated and careless practices of driver in a brief time. The Laziness Detecting Framework created based on eye closure of the driver can separate ordinary eye flicker and tiredness and distinguish the laziness while driving. The suggested device is able to avoid the incidents when driving due to sleepiness. The system works properly even in case of drivers sporting spectacles and even below low light stipulations if the digital camera offers higher output. Information about the head and eyes position is obtained through a range of self-developed photograph processing algorithms. During the monitoring, the system is able to figure out if the eyes are opened or closed. When the eyes have been closed for too long, a warning sign is issued. processing judges the driver's alertness level on the groundwork of continuous eye closures.

The following are the central key points that were found during the experimentation stage:

- 32x32x64x128x2(adam)(3conc 2 thick)
- 32x32x64x64x2(adam)(3conv 2 thick)
- 32x32x64x2(adam)(2conv 2 thick)
- 64x64x64x2(adam)(2conv 2dense)

These blends of convolutional and thick were attempted and the principal layer was viewed as the most suitable

- Adagrad (Orange)
- Adam(Blue)
- Nadam (Red)
- Adamax(Light Blue)

These optimisers were tried and Adamax performed better.

The model accomplishes highest precision (without over-fitting) with three secret layers.

CONCLUSION

In this paper, we have assembled a drowsy driver alert system only can execute in different ways. We utilized OpenCV to distinguish faces and eyes utilizing a haar cascade classifier and later that we used a CNN show to anticipate the status. The proposed model was tried under various conditions was lighting and was viewed as genuinely effective. This paper proposes a creative methodology for the location of drowsiness of a driver in another manner by judging the movements of his/her eyes. The past proposed models distinguished something very similar with the assistance of view, recurrence of yawning and a few other actual qualities. The proposed model is genuinely effective as it will ring an alert once the eyes of the driver stay shut more than the predetermined measure of time. Prior to showing up at the last proposed model, experimentation technique was utilized to decide the most ideal optimiser work. Deep learning techniques are notable for picture handling and this was investigated throughout this model.

REFERENCES

[1] Manu, B. N. 2016. Facial features monitoring for real time drowsiness detection. 2016 12th International Conference on Innovations in Information Technology (IIT).

[2] Alshaqaqi, B., Baquhaizel, A. S., Amine Ouis, M. E., Boumehed, M., Ouamri, A., & Keche, M. 2013. Driver drowsiness detection system. 2013 28th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA).

[3] Baek, J. W., Han, B.-G., Kim, K.-J., Chung, Y.-S., & Lee, S.-I. 2018. Real-Time Drowsiness Detection Algorithm for Driver State Monitoring Systems. 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN).

[4] You, F., Li, X., Gong, Y., Wang, H., & Li, H. 2019. A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration. IEEE Access, vol. 7, pp. 179396-179408.

[5] Zhang, S., & Wang, X. 2013. Human detection and object tracking based on Histograms of Oriented Gradients. 2013 Ninth International Conference on Natural Computation (ICNC).

[6] Eraldo, B., Quispe, G., Chavez-Arias, H., Raymundo-Ibanez, C., & Dominguez, F. 2019. Design of a control and monitoring system to reduce traffic accidents due to drowsiness through image processing. 2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX).

[7] Dasgupta, A., Rahman, D., & Routray, A. 2018. A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers. IEEE Transactions on Intelligent Transportation Systems, 1–10.

[8] Singh, M., Kaur, G.: Drowsy detection on eye blink duration using algorithm. International Journal of Emerging Technology and Advanced Engineering 2(4) (2012)

[9] Smith, P., Shah, M., Lobo, N.V.: Monitoring head/eye motion for driver alertness with one camera. Proceedings. 15th International Conference on Pattern Recognition (2000)

[10] Vural, E., Cetin, M., Ercil, A., Littlewort, G., Barlett, M.: Drowsy Driver detection using facial movement analysis" Proc. Springer-Verlag, Berlin; Heidelberg (2007)

[11] Vural, E., Bartlett, M.S., Littlewort, G., Cetin, M., Ercil, E., Movellan, J.: Discrimination of moderate and acute drowsiness based on spontaneous facial expressions. IEEE International Conference on Pattern Recognition (2010)

[12] Weirwille, W.W.: Overview of Research on Driver Drowsiness Definition and Driver Drowsi- ness Detection. 14th International Technical Conference on Enhanced Safety ofVehicles pp. 23–26 (1994)