Drowsiness Detection System

Thesis submitted in partial fulfilment

of the requirements of the degree of

Bachelor of Technology

in

Data Science

by

Abhisekh Pradhan(2019-B-12112000C),

Sanaga Shashank Reddy(2019-B-01062001A),

Kaustubh Dangche(2019-B-01042001A),
Vishal Ramesh Rathod(2019-B-19062001),
Shubham Jamadar(2019-B-26121999)

Under the Supervision of

Rahul Borate

Dec 2022

School of Engineering

Ajeenkya D Y Patil University, Pune

AJEENKYA D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY
School of Engineering

Dec 19, 2022

CERTIFICATE

This is to certify that the dissertation entitled **"Drowsiness Detection System"** is a bonafide work of "**Abhisekh Pradhan(2019-B-12112000C),Sanaga Shashank Reddy(2019-B-01062001A),KaustubhDangche(2019-B-01042001A),VishalRameshRathod(2019-B-19062001),Shubham Jamadar(2019-B-26121999)**" submitted to the School of Engineering, Ajeenkya D Y PatilUniversity, Pune in partial fulfillment of the requirement for the award of the degreeof **"Bachelor of Technology in Data Science"**.

Prof. Rahul Borate
Supervisor

Dr. Biswajeet champaty
Head school of Engineering

Internal-Examiner

External Examiner

AJEENKYA D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY
School of Engineering

Dec 19, 2022

Supervisor's Certificate

This is to certify that the dissertation entitled **"Drowsiness Detection System"** submitted by **Abhisekh Pradhan(2019-B-12112000C), Shashank Sanga Reddy(2019-B-01062001A), Kaustubh Dangche(2019-B-01042001A), Vishal Ramesh Rathod(2019-B-19062001), Shubham Jamadar(2019-B-26121999),** is a record of original work carried out by him/her under my supervision and guidance in partial fulfillment of the requirements of the degree of **Bachelor of Technology in Data Science** at **School of Engineering**, **Ajeenkya DY Patil University, Pune, Maharashtra-412105**. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Prof. Rahul Borate
Supervisor

AJEENKYA D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY
School of Engineering

Declaration of Originality

I, Abhisekh Pradhan(2019-B-12112000C), Sanaga Shashank Reddy(2019-B-01062001A), Kaustubh Dangche(2019-B-01042001A), Vishal Ramesh Rathod(2019-B-19062001), Shubham Jamadar(2019-B-26121999), hereby declare that this dissertation entitled "***Drowsiness Detection System"*** presents my original work carried out as a bachelor student of School of Engineering, Ajeenkya D Y Patil University, Pune, Maharashtra. To the best of my knowledge, this dissertation contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of Ajeenkya D Y Patil University, Pune or any other institution. Any contribution made to this research by others, with whom I have worked at Ajeenkya D Y Patil University, Pune or elsewhere, is explicitly acknowledged in the dissertation.Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I am fully aware that in case of any non-compliance detected in future, the Academic Council of Ajeenkya D Y Patil University, Pune may withdraw the degree awarded to me on the basis of the present dissertation.

Date:

Place: Lohegaon, Pune

_____

Acknowledgement

I remain immensely obliged to **Prof**. **Rahul Borate**, for providing me with the idea of this topic, and for his invaluable support in garnering resources for me either by way of information or computers also his guidance and supervision which made this Internship/Project happen.

I would like to say that it has indeed been a fulfilling experience for working out this Internship/Project.

Abstract

Road accidents are usually caused by drivercarelessness. The major carelessness exhibited by the driver are drunken behavior and negligence. The driver drowsiness detection system in automotive vehicle focuses on abnormal behavior exhibited by the driver using a microcontroller, the Raspberry pi single board computer. In the proposed system a non-intrusive driver drowsiness monitoring system has been developed using computer vision techniques. Irrespective of driver wearing spectacles and darkness level inside the vehicle, the systemis able to detect the drowsiness. The system will detect drowsiness within the time duration of about two to three seconds. The driver is alerted through alarms in real time.

Keywords— OpenCV, Real Time Captured Data .

Contents

CHAPTER 4 : CONCLUSION

List of Figures

List of Abbreviation

ALS	Amyotrophic lateral sclerosis
BCI	Brain-computer interface
DOF	Degree of freedom
DTCL	Dynamic threshold comparison logic
EEG	Electroencephalogram
EMG	Electromyogram
EOG	Electrooculogram

CHAPTER – 1 INTRODUCTION

1.1 ABOUT THE PROJECT

In this paper we are introducing a technique towards car security and security as well as people groups security in this we propose a Tiredness Recognition System.This paper consolidates Picture Processing,Computer Vision,pattern recognisation,Matlab Based programming. These days Driver weariness is a main consideration in an enormous number of vehicle mishaps. Late measurements gauge that every year 1,200 passings and 76,000 wounds can be ascribed to weariness related crashes. The improvement of advancements for distinguishing and staying away from sleepiness in the driver's seat is a significant test in the field of mishap evasion frameworks. In view of the peril that tiredness presents out and about, strategies should be created for checking its effects. The point of this undertaking is to foster a model sleepiness location framework. The emphasis is on planning a framework that will precisely screen the open or shut condition of the drivers eyes progressively. By observing the eyes, it is accepted that the side effects of driver weakness can be distinguished sufficiently early to keep away from an auto collision.

Location of tired includes an example of pictures of a face, and the perception of eye developments and squint rate. The examination of face pictures is a famous exploration region with applications like face acknowledgment, virtual devices, and human recognizable proof security frameworks. This task is utilized the limitation of the eyes, which includes taking a gander at the picture of the face, and deciding the place of the eyes by creating matlab program. When the place of the eyes is found, the framework is intended to decide if the eyes are opened or shut, and distinguish sleepiness. The reason for this study is to identify sleepiness in drivers to forestall mishaps and to further develop wellbeing on the roadways.

A technique for distinguishing sleepiness in drivers is created by utilizing a camera that point directlytowards the drivers face and catch for the continuous video. When the video is caught, checking the face district and eyes to recognize sleepy. The framework ready to observing eyes and decides if the eyes are in a vacant position or shut state. In such a situation when sleepiness is recognized, an admonition signal is given to caution the driver. It can decide a period time period conclusion as theproportion of a period span that the eye is in the shut position. In the event that the drivers eyes are shut in total in excess of a standard worth, the framework makes the determination that the driver is nodding off, and afterward it will enact a caution sound to caution the driver.

1.1.1 DEFINING DROWSINESS

The expression "sluggish" is inseparable from languid, which essentially implies a tendency to nod off. The phases of rest can be sorted as conscious, non-quick eye development rest (NREM), and fast eye development rest (REM). The subsequent stage, NREM, can be partitioned into the accompanying three phases:

- Stage I: progress from alert to snoozing (tired)
- Stage II: light rest
- Stages III: profound rest

To examine driver sleepiness, scientists have for the most part concentrated on Stage I, which is the sluggishness stage. The accidents that happen because of driver sluggishness have various attributes:

- Happen late around evening time (0:00 am-7:00 am) or during mid-evening (2:00 pm-4:00 pm)
- Include a solitary vehicle running off the street
- Happen on rapid streets
- Driver is frequently alone
- Driver is much of the time a youthful male, 16 to 25 years of age
- No pallet imprints or sign of slowing down

Corresponding to these qualities, the Southwest Britain and the Midlands Police information bases utilize the accompanying models to recognize mishaps that are brought about by tiredness:

- Blood liquor level beneath the legitimate driving cutoff
- Vehicle ran off the street or onto the rear of another vehicle
- No indication of brakes being applied
- Vehicle has no mechanical deformity
- Great atmospheric conditions and clear perceivability
- End of "speeding" or "driving excessively near the vehicle in front" as likely causes
- The cop at the scene suspects drowsiness as the essential driver

Insights determined utilizing these measures can't account completely for mishaps brought about by sleepiness due to the intricacy in question; in this manner, mishaps that can be ascribed to driver

sluggishness might be more pulverizing than the measurements uncover. Thus, to stay away from these sorts of mishaps, it is important to infer successful measures to recognize driver sleepiness and alarm the driver.

1.2 EXISTING SYSTEM

The current arrangement of driver sluggishness identification framework has following burdens. Mostly, utilizing of two cameras in the framework one for checking the head development and the other one for looks. The other burden is maturing of sensors and this large number of sensors are connected to the driver's body which might influence the driver. So to conquer this multitude of drawbacks we planned a framework in which a live camera is utilized for observing the driver sluggishness condition and caution the driver which lessens the street mishaps. By utilizing a non meddling machine vision based ideas, sleepiness of the driver identified framework is created. Many existing frameworks require a camera which is introduced before driver. It focuses straight towards the essence of the driver and screens the driver's eyes to recognize the tiredness. For enormous vehicle, for example, weighty trucks and transports this plan isn't appropriate. Transport has a huge front glass window to have a wide view for safe driving. In the event that we put a camera on the window of front glass, the camera obstructs the front facing perspective on driver so it isn't reasonable. Assuming that the camera is put on the edge which is just about the window, then the camera can't keep the front perspective on the essence of the driver accurately. The open CV locator distinguishes just 40% of face of driver in typical driving situation in video recording of 10 minutes. In the diagonal view, the Open CV eye finder (CV-ED) habitually neglects to follow the sets of eyes. On the off chance that the eyes are shut for five progressive approaches the framework presumes that the driver is declining sleeping and issues an admonition signal. Thus existing framework isn't material for enormous vehicles. To vanquish the issue of existing framework, new location framework is created in this venture work.

1.3 SCOPE OF WORK

The tiredness discovery proposed here is a base nosy methodology for checking driver sluggishness, in view of PC vision procedures, introduced on a genuine vehicle, fit for managing genuine activity conditions. Results acquired with the framework are comparable or shockingly better than other business ones being more adaptable and open source. The business frameworks frequently require a non-trifling adjustment system, to change the recognition. This strategy is precise up to 98%. This strategy for tiredness location takes less computational extremely less time. Consequently, it is very benefits to involve this method in the constant applications. Coming to future extension this framework can be additionally stretched out to have security like just specific individuals can get to the vehicle. In the event of robbery, the vehicle doesn't begin and a mms of the thief could be shipped off the proprietor of the vehicle.

The proposed method in this paper is an extraordinary and novel procedure. Utilizing division of picture strategy this tiredness recognition is finished. This technique can successfully distinguish sluggishness contrasted with all current strategy for sleepiness recognition. Face identification part is finished by utilizing face identifier capability accessible in the MATLAB 2012a programming followed by resizing the acquired picture to 512X512 picture, from which we separate the eye some portion of the individual picture. This eye part is gotten by noticing various pictures of driver, those values are taken, and agreeing the eye a piece of the picture is gotten. Consequently we can get just the eye part from the absolute picture.

1.4 OPERATING ENVIRONMENT – HARDWARE & SYSTEM

- Python

- OpenCV: OpenCV is an incredible instrument for picture handling and performing numerous PC vision undertakings. An open-source library can be utilized to perform assignments like face identification, object following, and a lot more undertakings.

- TensorFlow: Tensorflow is a free and open-source library, created by the Google Mind group for AI and man-made consciousness. Tensorflow has a specific spotlight on the preparation and derivation of profound brain organizations.

- Keras: Keras is an open-source programming library and it gives a Python connection point to fake brain organizations. Keras is more easy to understand on the grounds that it is an inbuilt python library.

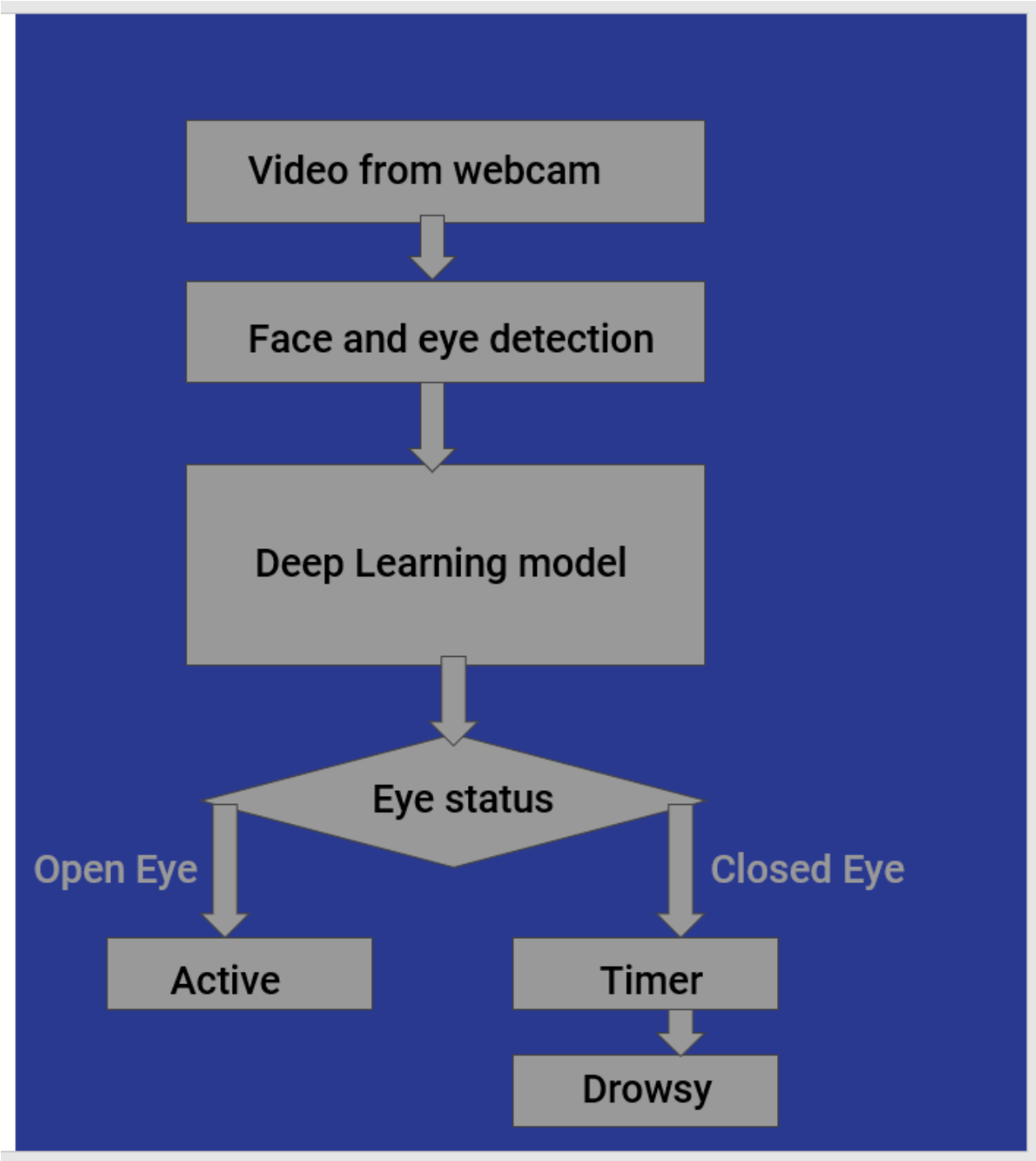
1.5 PROPOSED SYSTEM

The proposed framework here is intended to limit the event of innumerable incidents because of the sleepy driver. These days, exhaustion of driver causes street mishaps occasionally across the world. Thus, these exercises ought to be expected to consequently deal with an execution of shrewd ready framework or carefulness in a vehicle which is a target of this framework. To dissect different social or visual-based perspectives of the driver, face development and eye squint are estimated to concentrate on the condition of the driver. Here, eye flicker is chiefly engaged to identify tiredness of the driver. The limit worth of an EAR lies above 0.25 with no impact of weariness. At the point when a driver naturally closes down, then the limit worth of EAR falls underneath the given reach. An edge worth of sluggish eye flicker test addresses the quantity of video casings of the driver's shut eyes. In the event that the back to back counting outlines increment over the scope of the limit esteem, then the sleepiness of the driver is distinguished. Here, a Pi camera is utilized toconsistently record the all out development of an eye through which the limit worth of an EAR is determined. A counter is likewise remembered for it for counting event of casings. Assume that it surpassed over a scope of 30. All things considered, a voice is enacted by a speaker and a mail is naturally shipped off an approved individual of the vehicle which is by and large handled at the hour of sleepiness recognition. The portrayed modules work appropriately through Raspberry Pi3which ismodified in Python programming language.

CHAPTER 2 – METHODOLGY

2.1 METHODOLOGY

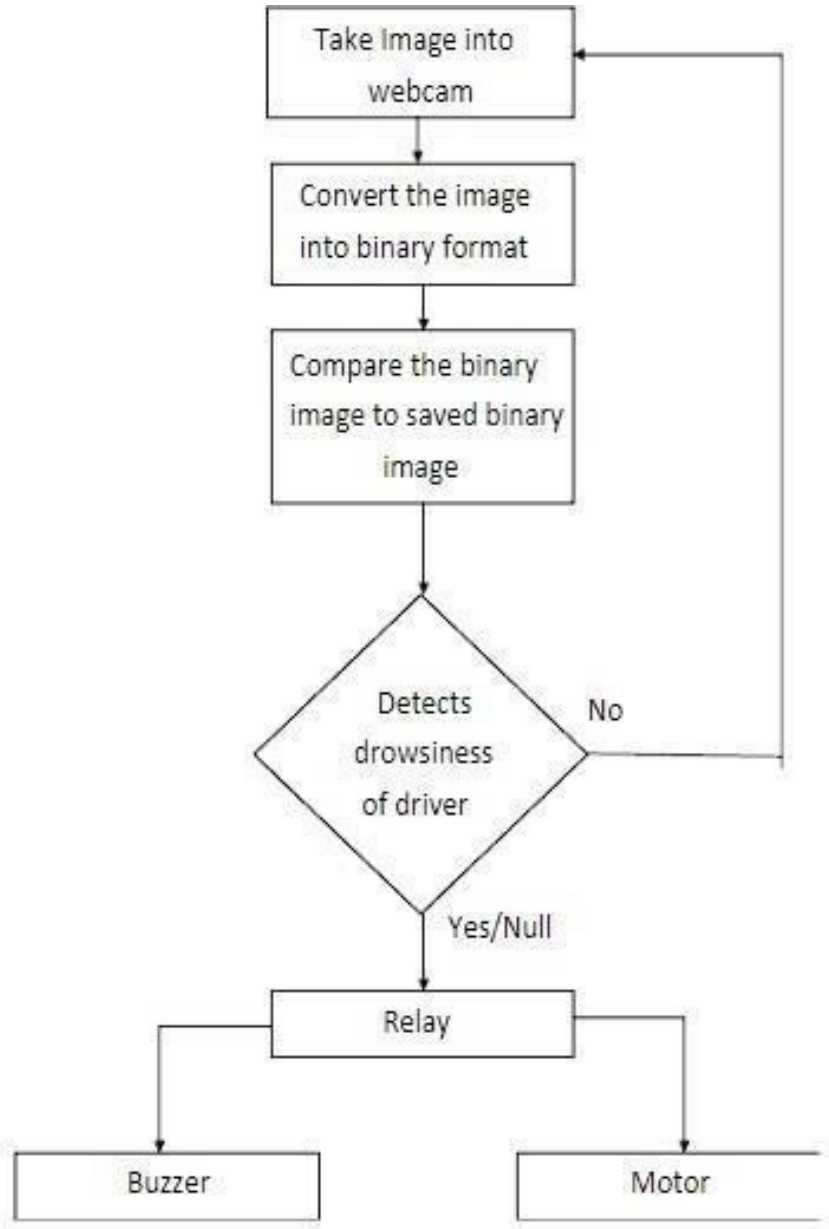
The most common way of losing sharpness in the driver's seat because of exhaustion can be portrayed by a steady movement of facial elements: Shifts connecting with the course of the look of the driver, Changes in the place of the eyelids or the size of the eyes, Quick changes in pace of squinting and direction and position of the head. The new framework is sluggishness location framework for auto. Driver sluggishness not entirely settled from a few side effects that manifest in tired driver's face. Through investigation of the eye expresses, the framework will actually want to tell a sluggish driver from a typical driver. A video transfer will be constantly gotten from the driver's countenances and feed into a microcontroller for handling. Classifiers will then be utilized to group the condition of the driver's eye. In the event that a sluggish driver is distinguished a caution will be raised, until the framework sees the driver is ready. Advancement of the framework was through dexterous strategy where the scrum and the outrageous programming technique were joined. The framework was broken into little modules, these modules were grown freely and tried incorporation was finished. During unit, testing refactoring was taken on to advance the units for their expected reason.



1.1 METHODOLOGY OF DDS

2.1.2 FLOW CHART

By using algorithm user can find the whether the drowsy detection is done or not. In algorithm system first take the image where it is stored, then Binarization of image take place. Find out position of eye, whether Eye open or closed. If Open Eye Image is detected then signal gets no drowsy found, Alarm is OFF. If Closed Eye Image is detected then signal gets drowsy found, Alarm is ON.



1.2 FLOW CHART DIAGRAM

2.2 METHOD TO DETECT DROWSINESS

2.2.1 TEMPLATE MATCHING BASED EYE DETECTION IN FACIAL IMAGE

Eye identification is a pre-imperative stage for some applications like human-PC interfaces, iris acknowledgment, driver tiredness location, security, and science frameworks. In this paper, format based eye identification is portrayed. The format is corresponded with various locales of the face picture. The district of face which gives greatest relationship with layout alludes to eye locale. The technique is straightforward and simple to execute. The viability of the technique is shown in both the cases like open eye as well as shut eye through different reenactment results A novel and basic eye recognition conspire is proposed in this paper. An eye layout is utilized to identify eye locale from face picture. The layout is coordinated with eye locale utilizing cross connection strategy. The strategy requires no complex numerical estimation and earlier information about the eye. It is a basic technique and can undoubtedly be executed by equipment.

2.2.2 DETECTION METHOD FOR DRIVER DROWSINESS APPLICABLE FOR INDIVIDUAL FEATURES

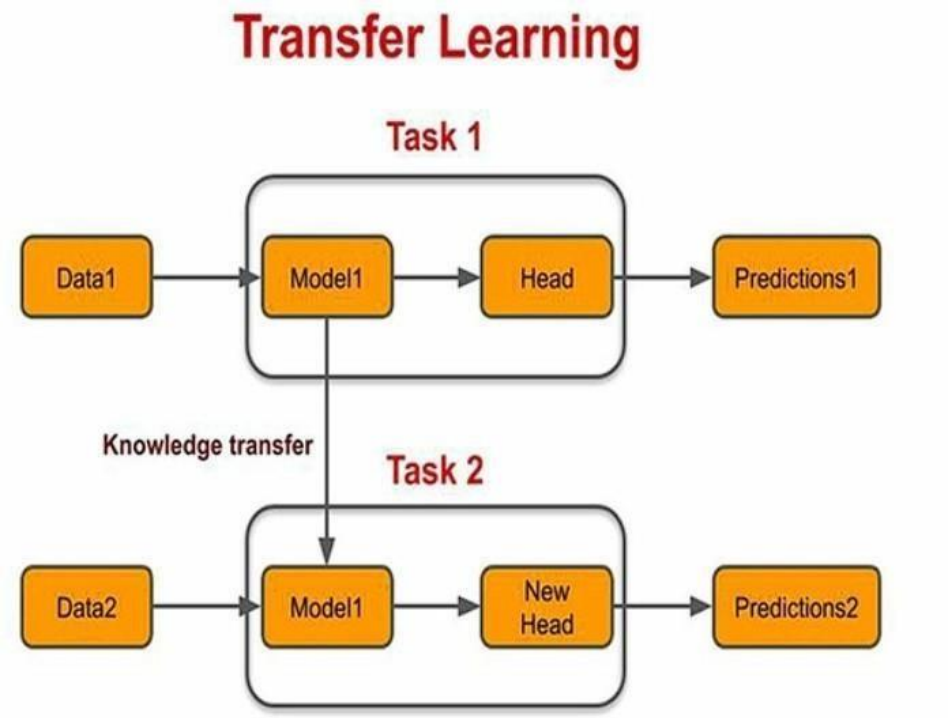
In this technique for example the driver status screen framework, the technique or the timing for offering data to a driver is changed by the level of the cognizance or the consideration of a driver, and the media or its strategy to offer data is changed by consent or desperation level of the data. The motivation behind this study is to understand a framework that wins drive's certainty by the ways referenced previously. The driver Stan's screen distinguishes tiredness from the adjustment of the length of eye conclusion during flickering and in consideration from the shift in the look course. This strategy portrays the discovery of debasement of cognizance.

2.2.3 ALGORITHM FOR COMMERCIAL VEHICLE DRIVER DROWSINESS DETECTION SYSTEM

This strategy depicts a trial examination of financially authorized drivers who were exposed to tiredness conditions in a truck-driving test system and assesses the presentation of a brain network based calculation, which screens just the drivers‟ directing info. Connections are tracked down between the adjustment of directing and the condition of sleepiness. The outcomes show directing signs contrasts can be utilized actually for location. This is a managed preparing where the known inputoutput designs are introduced to the organization and the ANN learns (stores) the data The info designs are the models, for example 15-second added of disparaged controlling point, and the result is known condition of the driver, for example the ideal result vector, D(n).D(n) is addressed by a grouping vector worth of [1,0] for conscious and [0,1] for rest. Consequently, for preparing, for each information model X relates to a known result D(n). The introduction of info yield designs is arbitrary, chose from the 600 models. Preparing an ANN requires choosing the right and ideal engineering for the different preparation boundaries. The ANN preparing is played out various times. An assessment of the ANN model shows great execution under the accident expectation

metric. The framework gave somewhere around one discovery for 97% of the multitude of noticed crashes experienced by any of the subjects. Directing way of behaving is portrayed by a period with no guiding remedy. Subsequently, the ANN calculation can't identify these occasions since it was prepared for.

2.3 TRANSFER LEARNING



1.3 TRANSFER LEARNING DIAGRAM

In this task, we will utilize move figuring out how to fabricate the model. Move Learning is an AI strategy where we utilize a pre-prepared model for another model with the connected issue explanation.

For instance, a model that is utilized for the recognization of vehicles can be utilized for the recognization of trucks.

Here for the most part it centers around the information that is acquired while tackling one issue and it applies to an alternate yet related issue.

2.3.1 DATASET

For this undertaking, we will utilize the MRL Eye dataset. MRL Eye dataset is a huge scope dataset that contains natural eye images.Download the dataset utilizing this connection. http://mrl.cs.vsb.cz/eyedataset

This dataset contains 84898 eye pictures of 37 distinct people of which 33 are men and 4 are ladies. It contains a wide range of pictures like open, shut, with glasses, without glasses, and in various lighting conditions.

In the dataset, we commented on the accompanying properties (the properties are shown in the accompanying request):

•subject ID; in the dataset, we gathered the information of 37 distinct people (33 men and 4 ladies)

•picture ID; the dataset comprises of 84,898 pictures

•orientation [0 - man, 1 - woman]; the dataset contains the data about orientation for each picture (man, lady)

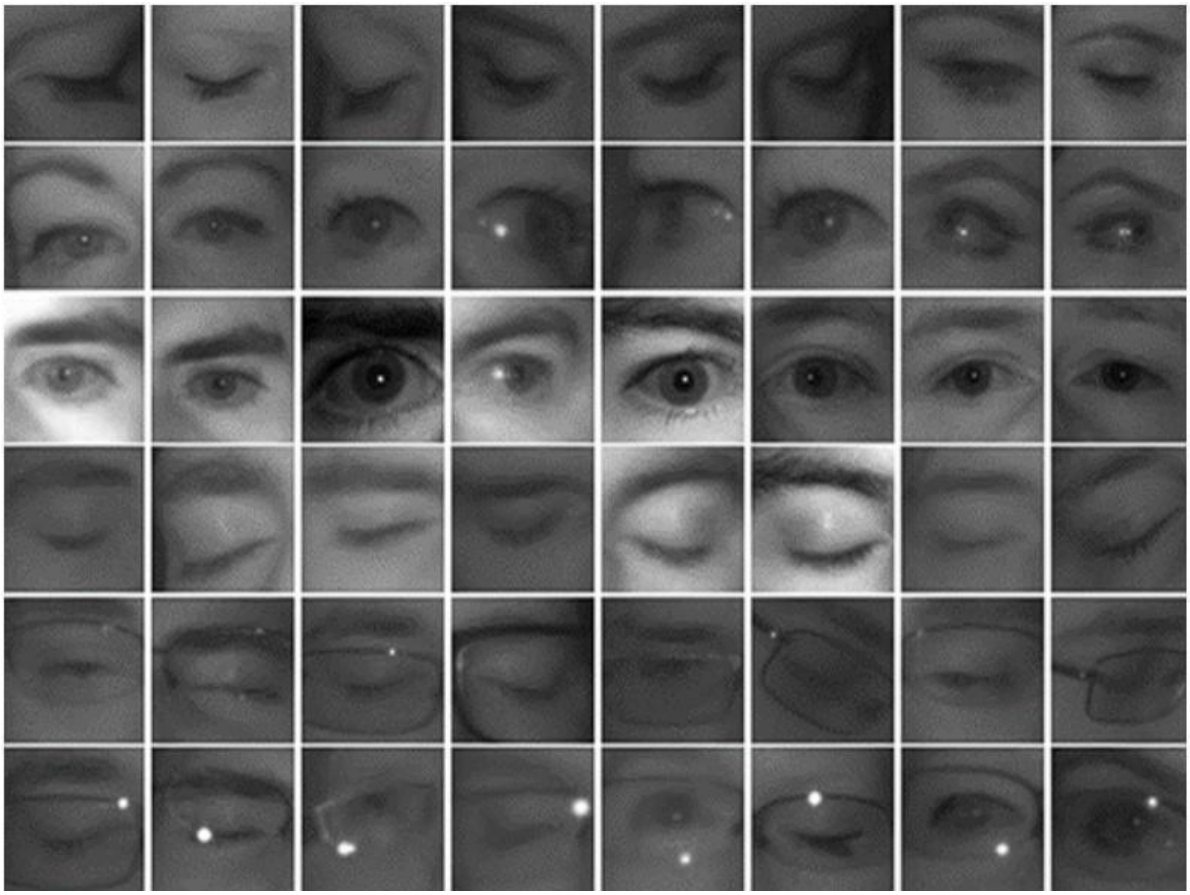
•glasses [0 - no, 1 - yes]; the data assuming the eye picture contains glasses is additionally accommodated each picture (with and without the glasses)

•eye state [0 - shut, 1 - open]; this property contains the data around two eye states (open, close)

•reflections [0 - none, 1 - little, 2 - big]; we commented on three reflection states in light of the size of reflections (none, little, and huge reflections)

•lighting conditions [0 - awful, 1 - good]; each picture has two states (terrible, great) in view of how much light during catching the recordings

•sensor ID [01 - RealSense, 02 - IDS, 03 - Aptina]; right now, the dataset contains the pictures caught by three distinct sensors (Intel RealSense RS 300 sensor with 640 x 480 goal, IDS Imaging sensor with 1280 x 1024 goal, and Aptina sensor with 752 x 480 goal)



1.4 EXAMPLE OF DATASET

Subsequent to downloading separate all pictures into shut and open eyes into two separate envelopes.

The discovery of eyes and their parts, look assessment, and eye-squinting recurrence are significant undertakings in PC vision. In last years, we have been settling these undertakings in the space of driver's way of behaving, which causes the securing of a great deal of testing information that was procured in genuine circumstances. Subsequently, we present the MRL Eye Dataset, the enormous scope dataset of natural eye pictures. This dataset contains infrared pictures in low and high goal, all caught in different lightning conditions and by various gadgets. The dataset is appropriate for testing a few elements or teachable classifiers. To work on the correlation of calculations, the pictures are partitioned into a few classifications, which likewise makes them reasonable for preparing and testing classifiers

CHAPTER 3 – RESULTS AND DISCUSSION

3.1 THE DROWSINESS DETECTION ALGORITHM

The general flow of our drowsiness detection algorithm is fairly straightforward.

First, we'll setup a camera that monitors a stream for faces:

Step 1 - Look for faces in the input video stream.

If a face is found, we apply facial landmark detection and extract the eye regions:

Step 2 - Apply facial landmark localization to extract the eye regions from the face.

Now that we have the eye regions, we can compute the eye aspect ratio to determine if the eyes are closed:

Step 3 - Compute the eye aspect ratio to determine if the eyes are closed.

If the eye aspect ratio indicates that the eyes have been closed for a sufficiently long enough amount of time, we'll sound an alarm to wake up the driver:

Step 4 - Sound an alarm if the eyes have been closed for a sufficiently long enough time.

3.2 Steps for Performing Driver Drowsiness Detection

Stage 1 - Accept Picture as Contribution from a Camera With a webcam, we will accept pictures as info. So to get to the webcam, we made a limitless circle that will catch each edge. We utilize the technique given by OpenCV, cv2.VideoCapture(0) to get to the camera and set the catch object (cap). cap.read() will peruse each casing and we store the picture in an edge variable.

Stage 2 - Distinguish Face in the Picture and Make a District of Interest (return on initial capital investment) To recognize the face in the picture, we really want to initially change over the picture into grayscale as the OpenCV calculation for object location takes dim pictures in the information. We don't require variety data to recognize the items. We will utilize haar overflow classifier to recognize faces. This line is utilized to set our classifier face = cv2.CascadeClassifier(' way to our haar overflow xml document'). Then we play out the recognition utilizing faces = face.detectMultiScale(gray). It returns a variety of location with x,y directions, and level, the width of the limit box of the article. Presently we can emphasize over the countenances and draw limit boxes for each face.

Stage 3 - Identify the eyes from return for money invested and feed it to the classifier. A similar technique to identify faces is utilized to recognize eyes. To begin with, we set the outpouring classifier for eyes in leye and reye separately then recognize the eyes utilizing left_eye = leye.detectMultiScale(gray). Presently we really want to remove just the eyes information from the full picture. This can be accomplished by separating the limit box of the eye and afterward we can take out the eye picture from the edge with this code. l_eye just holds back the picture information of the eye. This will be taken care of into our CNN classifier which will anticipate in the event that eyes are open or shut. Additionally, we will separate the right eye into r_eye.

Stage 4 - Classifier will Sort whether Eyes are Open or Shut We are involving CNN classifier for anticipating the eye status. To take care of our picture into the model, we really want to play out specific activities on the grounds that the model necessities the right aspects to begin with. To start with, we convert the variety picture into grayscale utilizing r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY). Then, 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 union r_eye = r_eye/255 (All values 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 foresee each eye with our model lpred = model.predict_classes(l_eye). If the worth of lpred[0] = 1, it expresses that eyes are open, in the event that worth of lpred[0] = 0, it expresses that eyes are shut.

Stage 5 - Work out Score to Check whether Individual is Tired The score is fundamentally a worth we will use to decide how long the individual has shut his eyes. So assuming the two eyes are shut, we will continue to expand score and when eyes are open, we decline the score. We are drawing the outcome on the screen utilizing cv2.putText() capability which will show constant

status of the individual. An edge is characterized for instance on the off chance that score becomes more noteworthy than 15 that implies the individual's eyes are shut for a significant stretch of time. This is the point at which we signal the caution utilizing sound.play().

In this Python project, we have constructed a sluggish driver ready framework that you can carry out in various ways. We utilized OpenCV to identify faces and eyes utilizing a haar overflow classifier and afterward we utilized a CNN model to foresee the status.

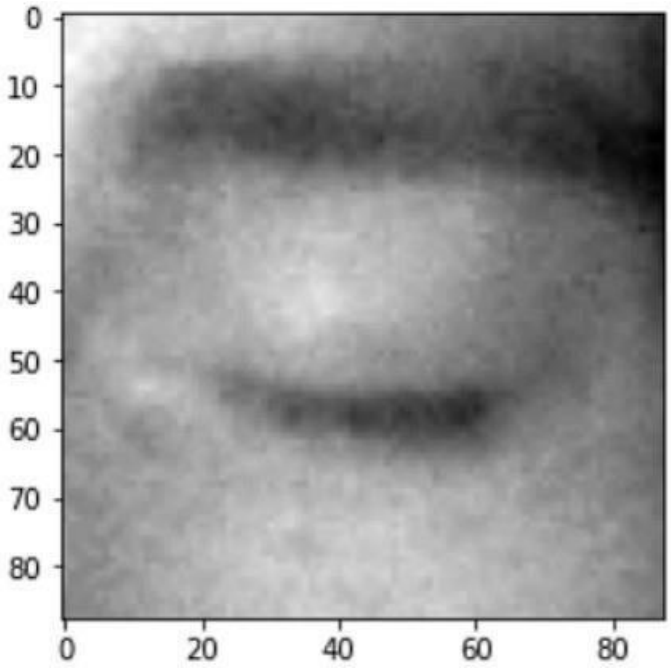
3.3 IMPLEMENTATION

First of all required important libraries

```
import tensorflow as tf
import cv2
import os
import matplotlib.pyplot as plt
import numpy as np
```

Reading all the images in the dataset

```
Datadirectory = "mrlEyeDataset2/"
Classes = ["Closed_Eyes", "Open_Eyes"]
for category in Classes:
    path = os.path.join(Datadirectory, category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img), cv2.IMREAD_GRAYSCALE)
        backtorgb = cv2.cvtColor(img_array, cv2.COLOR_GRAY2RGB)
        plt.imshow(img_array, cmap="gray")
        plt.show()
        break
    break
```



Resizing the Image

Resize all the images into 224 x 224 for better feature extraction.

```
img_size = 224
new_array = cv2.resize(backtorgb, (img_size, img_size))
plt.imshow(new_array, cmap="gray")
plt.show()
```

Creating training data.

```
training_Data = []
def create_training_Data():
   for category in Classes:
       path = os.path.join(Datadirectory, category)
       class_num = Classes.index(category) # 0 1,
       for img in os.listdir(path):
           try:
               img_array = cv2.imread(os.path.join(path, img), cv2.IMREAD_GRAYSCALE)
               backtorgb = cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
               new_array = cv2.resize(backtorgb, (img_size, img_size))
               training_Data.append([new_array,class_num])
           except Exception as e:
               pass
create_training_Data()
```

Random shuffling to avoid overfitting.

```
import random
random.shuffle(training_Data)
```

Creating arrays to store features and labels

```
X = []
y = []
for features,label in training_Data:
  X.append(features)
  y.append(label)
X = np.array(X).reshape(-1, img_size, img_size, 3)
```

Import libraries that are required to build the model. Import the model and change the last fully connected(fc) layers and build the new model. Then compile the model using binary cross-entropy loss function and adam optimizer and then train the model for 5 epochs and then save the model so that no need to train again and again.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
```

```
img_array = cv2.imread('mrlEyeDataset/Closed_Eyes/s0001_00080_0_0_0_0_0_01.png', cv2.IMREAD_GRAY
backtorgb = cv2.cvtColor(img_array, cv2.COLOR_GRAY2RGB)
new_array = cv2.resize(backtorgb, (img_size, img_size))
X_input = np.array(new_array).reshape(1, img_size, img_size, 3)
X_input = X_input/255.0 #normalizing data
prediction = new_model.predict(X_input)
```

```
model = tf.keras.applications.mobilenet.MobileNet()
base_input = model.layers[0].input ##input
base_output = model.layers[-4].output
Flat_layers = layers.Flatten()(base_output)
final_output = layers.Dense(1)(Flat_layers)
final_output = layers.Activation('sigmoid')(final_output)
new_model = keras.Model(inputs = base_input, outputs = final_output)
new_model.compile(loss="binary_crossentropy", optimizer= "adam", metrics=["accuracy"])
new_model.fit(X,Y, epochs = 5, validation_split = 0.2) ##training
new_model.save('my_model.h5')
```

Detection and Alerting

Now let us see the implementation for detection and alerting. Here we set the timer for 2 seconds and 5 frames per second. If the eyes were closed for 2 seconds continuously that means if the model captures continuous 10 closed eye frames then it will alert the driver by beeping the alarm sound.

```
import winsound
frequency = 2500  # Set frequency to 2500
duration = 1500  # Set duration to 1500 ms == 1.5 sec
import numpy as np
import cv2
path = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml
cap = cv2.VideoCapture(1)
#check if webcam is opened correctly
if not cap.isOpened():
    cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FPS, 5)
counter = 0
while True:
```

```
ret,frame = cap.read()
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
eyes = eye_cascade.detectMultiScale(gray, 1.1, 4)
for x,y,w,h in eyes:
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
    cv2.rectangle(frame, (x,y), (x+w,y+h), (0, 255, 0), 2)
    eyess = eye_cascade.detectMultiScale(roi_gray)
    if len(eyess) == 0:
        print("Eyes are not detected")
    else:
        for (ex, ey, ew, eh) in eyess:
            eyes_roi = roi_color[ey: ey+eh, ex: ex+ew]
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
if(faceCascade.empty()==False):
    print("detected")
faces = faceCascade.detectMultiScale(gray, 1.1, 4)
# Draw a rectangle around eyes
for (x,y,w,h) in faces:
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
font = cv2.FONT_HERSHEY_SIMPLEX
```

```
font = cv2.FONT_HERSHEY_SIMPLEX
final_image = cv2.resize(eyes_roi, (224,224))
final_image = np.expand_dims(final_image, axis=0)
final_image = final_image/255.0
Predictions = new_model.predict(final_image)
if (Predictions>=0.3):
    status = "Open Eyes"
    cv2.putText(frame,
            status,
            (150,150),
            font, 3,
            (0, 255, 0),
            2,
            cv2.LINE_4)
    x1,y1,w1,h1 = 0,0,175,75
    cv2.rectangle(frame, (x1, y1), (x1 + w1, y1 + h1), (0,0,0), -1)
    #Add text
    cv2.putText(frame, 'Active', (x1 + int(w1/10),y1 + int(h1/2)), cv2.FONT_HERSHEY_SIMPLEX
elif Predictions<0.3:
    counter = counter + 1
    status = "Closed Eyes"
    cv2.putText(frame,
```

```
        cv2.putText(frame,
                    status,
                    (150,150),
                    font, 3,
                    (0, 0, 255),
                    2,
                    cv2.LINE_4)
        x1,y1,w1,h1 = 0,0,175,75
        cv2.rectangle(frame, (x1,y1), (x1 + w1, y1 + h1), (0,0,255), 2)
        if counter > 10:
            x1,y1,w1,h1 = 0,0,175,75
            #Draw black background rectangle
            cv2.rectangle(frame, (x1, y1), (x1 + w1, y1 + h1), (0,0,0), -1)
            #Add text
            cv2.putText(frame, "Sleep Alert !!!", (x1 + int(w1/10), y1 + int(h1/2)), cv2.FONT_
            winsound.Beep(frequency, duration)
            counter = 0
    cv2.imshow("Drowsiness Detection", frame)
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break
ap.release()
v2.destroyAllWindows()
```

CHAPTER 4 – CONCLUSION

4.1 CONCLUSION

The tiredness discovery proposed here is a base meddling methodology for observing driver sleepiness, in view of PC vision procedures, introduced on a genuine vehicle, fit for managing genuine activity conditions. Results acquired with the framework are comparable or stunningly better than other business ones being more adaptable and open source. The business frameworks frequently require a non-trifling adjustment methodology, to change the discovery. This technique is exact up to 98% .This strategy for sleepiness location takes less computational exceptionally less time. Thus, it is very benefits to involve this procedure in the continuous applications. Coming to future extension this framework can be additionally reached out to have security like just specific individuals can get to the vehicle. In the event of robbery, the vehicle doesn't begin and a mms of the criminal could be shipped off the proprietor of the vehicle.

At the point when you run the code it will open the webcam and will catch the video and provides yield in light of your eyelid a sense of finality. This sleepiness discovery framework helps the drivers a ton and forestalls numerous street mishaps that are caused because of sluggishness. Up until this point we have seen,

-> How to utilize move learning?

-> How to construct the model, train the model?

-> How to change the layers as per our concern explanation?

-> At long last execution of the sleepiness discovery framework.

Trust you all thought that it is helpful.

4.2 LIMITATIONS

1. Reliance on encompassing light: - With unfortunate lighting conditions despite the fact that face is effectively recognized, in some cases the framework can't distinguish the eyes. So it gives an incorrect outcome which should be dealt with. Continuously situation infrared backdrop illuminations ought to be utilized to stay away from unfortunate lighting conditions.

2. Ideal reach required: - when the distance among face and webcam isn't at ideal reach then specific issues are emerging. At the point when face is excessively near webcam(less than 30 cm), the framework can't identify the face from the picture. At the point when face is away from the web cam (more than 70cm) then, at that point, the backdrop illumination is lacking to appropriately enlighten the face. So eyes are not identified with high exactness which shows blunder in location of sluggishness. This issue isn't truly considered as progressively situation the distance between

drivers face and webcam doesn't surpass 50cm. so the issue won't ever emerge. Taking into account the above troubles, the ideal distance range for sluggishness location is set to 40-70 cm

4. Direction of face: - when the face is shifted partially it tends to be recognized, yet past this framework neglects to distinguish the face. So when the face isn't identified, eyes are likewise not distinguished. This issue is settled by utilizing following capabilities which track any development and pivot of the articles in a picture.

5. Unfortunate location with exhibitions: - When the driver wears glasses the framework neglects to recognize eyes which are the main downside of our framework. This issue has not yet been settled and is really difficult for practically all eye identification frameworks planned up until this point.

6. Issue with numerous appearances: - In the event that more than one face is identified by the webcam, our framework gives a wrong outcome. This issue isn't significant as we need to distinguish the sleepiness of a solitary driver.

REFERENCES

[1].Ameratunga.S , Bailey.J, Connor.J, Civil.I, Dunn.R , Jackson.R , Norton.R, and, Robinson.E, ―Driver sleepiness and risk of serious injury to car occupants: Population control study.‖ British Medical Journal, vol. 324, 2002, pp. 1125–1129.

[2]. Bronte.S, Bergasa.L, Delgado.B, Garc´ıa.I, Hern´andez.N , and Sevillano.M, ―Vision-based drowsiness detector for a realistic driving simulator,‖ in IEEE Intelligent Transportations Systems Conference (ITSC), 2010.

[3]. Distante.A, D'Orazio.T, Guaragnella.C, and Leo.M, ―A visual approach for driver inattention detection,‖ Pattern Recogn., vol. 40, no. 8, 2007, pp. 2341–2355.

[4].Dingus.T.A, Klauer.S.G, Neale.V.L, and Sudweeks.J.D,‖ The impact of driver inattention on near-crash/crash risk: An analysis using the 100 car naturalistic driving study data‖, National Highway Traffic Safety Adminstration, DC, DOT HS, 2006, vol.810.

[5]. Faber.J, ―Detection of different levels of vigilance by eeg pseudo spectra.‖ Neural Network World, vol. 14, no. 3-4, pp. 285–290, 2004.

[6]. Igarashi.K ,Itou.K, Itakura.F, Miyajima.C, Ozawa.T, Takeda.K and Wakita.T, ―Driver identification using driving behavior signals,‖ IEICE - Trans. Inf. Syst., vol. E89- D, no. 3, 2006, pp. 1188–1194.

[7].Nakano.T, Suzuki.M, Yamamoto.N, Yamamoto.O, and Yamamoto.S, ―Measurement of driver's consciousness by image processing a method for presuming driver's drowsiness by eyeblinks coping with individual differences.‖ Systems, Man and Cybernetics, vol. 4, 2006, p. 2891– 2896.

[8].Tippayanate. N, The impact of sleep deprivation on driving performance. Khon Kaen Hospital Medical Journal, 30(3): 320-325, 2006.