DROWSINESS DETECTION SYSTEM

A Project Report
Submitted in partial fulfilment of the Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

THOUTI HYNDAVI (1602-18-737-074)

PEDDI LALITH MOHAN (1602-18-737-078)

BEJUGAM MAHALAXMI (1602-18-737-080)

Under the guidance of

Mr. R. Dharma Reddy

Professor

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University) Ibrahimbagh,

Hyderabad-31

2020-2021

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Information Technology

DECLARATION BY THE CANDIDATES

We, **THOUTI HYNDAVI, PEDDI LALITH MOHAN, BEJUGAM MAHALAXMI** bearing hall ticket number, **1602-18-737-074, 1602-18-737-078, 1602-18-737-080,** hereby declare that the project report entitled **"DROWSINESS DETECTION SYSTEM"** under the guidance of **Mr.R.Dharma Reddy**, Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement of MINI PROJECT of VI semester of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

THOUTI HYNDAVI (1602-18-737-074)
PEDDI LALITH MOHAN (1602-18-737-078)
BEJUGAM MAHALAXMI (1602-18-737-080)

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**DROWSINESS DETECTION SYSTEM"** being submitted by **THOUTI HYNDAVI, PEDDI LALITH MOHAN, BEJUGAM MAHALAXMI** bearing **1602-18-737-074, 1602-18-737-078, 1602-18-737-080,** in partial fulfilment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Mr.R.Dharma Reddy	**Dr. K. Ram Mohan Rao**
Professor	**HOD, IT**
Internal Guide	

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We would like to take the opportunity to express ourhumble gratitude to **Mr.R.Dharma Reddy** (**Professor)** under whom we executed this project. We would also use this opportunity to thank our Head Of Department **Dr.K.Ram Mohan Rao** . We would also like to thank all faculty members and staff of the Department ofInformation Technology for their generous help in various ways for the completion of this project.

Finally, We would like to express our heartfelt thanks to our senior **ALLA NIKHITHA REDDY(1602-17-737-087**).We are grateful to his guidance, and constructivesuggestions that helped us in the preparation of this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

ABSTRACT

Traffic accidents are extremely common in big cities. One of the main causes of traffic accidents is Drowsy driving. When drowsy, drivers cannot react to dangerous situations which is the cause of major accidents. Driving for a long time without any rest also leads to accidents. There is statistical evidence that the drowsiness of drivers is a primary cause of road accidents all over the world. To reduce such accidents early detection of drowsy driving is needed.

There is a relation between drivers' drowsiness and facial expressions. Monitoring driver's facial expressions is a method for detecting driver drowsiness. The face contains information that can be used to interpret levels of drowsiness. Facial features can be extracted from the face to infer the level of drowsiness which include eye blinks, head movements, and yawning.

Drowsiness also has a negative impact on people in working and classroom environments. Drowsiness while working with heavy machinery may result in serious injuries as in the case of drowsy driving.

Due to the relevance of this problem, it is important to develop a solution for drowsiness detection, especially in the early stages to prevent accidents and to alert the drivers. The objective is to build a detection system using video analysis that identifies key attributes of drowsiness and triggers an alert when someone is drowsy.

Contents

Chapter 1

Introduction

1.1 PURPOSE

1.1.1 HUMAN PSYCHOLOGY WITH CURRENT TECHNOLOGY

Humans have always invented machines and devised techniques to ease and protect their lives, for mundane activities like traveling to work, or for more interesting purposes like aircraft travel. With the advancement in technology, modes of transportation kept on advancing and our dependency on it started increasing exponentially. It has greatly affected our lives as we know it. Now, we can travel to places at a pace that even our grandparents wouldn’t have thought possible. In modern times, almost everyone in this world uses some sort of transportation every day. Some people are rich enough to have their own vehicles while others use public transportation. However, there are some rules and codes of conduct for those who drive irrespective of their social status. One of them is staying alert and active while driving.

Neglecting our duties towards safer travel has enabled hundreds of thousands of tragedies to get associated with this wonderful invention every year. It may seem like a trivial thing to most folks but following rules and regulations on the road is of utmost importance. While on road, an automobile wields the most power and in irresponsible hands, it can be destructive and sometimes, that carelessness can harm lives even of the people on the road. One kind of carelessness is not admitting when we are too tired to drive. In order to monitor and prevent a destructive outcome from such negligence, many researchers have written research papers on driver drowsiness detection systems. But at times, some of the points and observations made by the system are not accurate enough. Hence, to provide data and another perspective on the problem at hand, in order to improve their implementations and to further optimize the solution, this project has been done.

1.1.2 FACTS & STATISTICS

Our current statistics reveal that just in 2015 in India alone, 148,707 people died due to car related accidents. Of these, at least 21 percent were caused due to fatigue causing drivers to make mistakes. This can be a relatively smaller number still, as among the multiple causes that can lead to an accident, the involvement of fatigue as a cause is generally grossly underestimated. Fatigue combined with bad infrastructure in developing countries like India is a recipe for disaster. Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs, which have clear key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue when needed. The former is hard and much more expensive to achieve, and the latter is not possible without the former as driving for long hours is very lucrative. When there is an increased need for a job, the wages associated with it increases leading to more and more people adopting it. Such is the case for driving transport vehicles at night. Money motivates drivers to make unwise decisions like driving all night even with fatigue. This is mainly because the drivers are not themselves aware of the huge risk associated with driving when fatigued. Some countries have imposed restrictions on the number of hours a driver can drive at a stretch, but it is still not enough to solve this problem as its implementation is very difficult and costly.

1.2 DOCUMENT CONVENTIONS

Main Heading Font size: 24 (bold fonts) Sub-

headings Font size: 16 (bold fonts)

Sub-headings Content Font size: 14 (normal fonts)

1.3 INTENDED AUDIENCE

The intended audience for this document are the development team, the project evaluation jury, and other tech-savvy enthusiasts who wish to further work on the project.

1.4 PRODUCT SCOPE

There are many products out there that provide the measure of fatigue level in the drivers which are implemented in many vehicles. The driver drowsiness detection system provides the similar functionality but with better results and additional benefits. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure.

1.5 PROBLEM DEFINITION

Fatigue is a safety problem that has not yet been deeply tackled by any country in the world mainly because of its nature. Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs, which have clear key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue when needed. The former is hard and much more expensive to achieve, and the latter is not possible without the former as driving for long hours is very lucrative.

Chapter 2

Literature Survey

2.1 SYSTEM REVIEW

This survey is done to comprehend the need and prerequisite of the general population, and to do as such, we went through different sites and applications and looked for the fundamental data. Based on these data, we made an audit that helped us get new thoughts and make different arrangements for our task. We reached the decision that there is a need of such application and felt that there is a decent extent of progress in this field too.

2.2 TECHNOLOGY USED

a. PYTHON - Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

b. JUPYTER Lab - Project Jupyter is a nonprofit organization created to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

c. IMAGE PROCESSING - In computer science, digital image processing is the use of computer algorithms to perform image processing on digital images.

d. MACHINE LEARNING - Machine learning is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly told.

Chapter 3

Software Requirements Specification

3.1 Python

- Numpy
- Scipy
- Playsound
- Dlib
- Imutils
- opencv, etc.

3.2 Operating System

- Windows or Ubuntu

Hardware Requirements Specification

I. Laptop with basic hardware.
II. Webcam

Chapter 4

Requirement Analysis

4.1 Python: Python is the basis of the program that we wrote. It utilizes many of the python libraries.

4.2 Libraries:

- Numpy: Pre-requisite for Dlib
- Scipy: Used for calculating Euclidean distance between the eyelids.
- Playsound: Used for sounding the alarm
- Dlib: This program is used to find the frontal human face and estimate its pose using 68 face landmarks.
- Imutils: Convenient functions written for Opencv.
- Opencv: Used to get the video stream from the webcam, etc.

4.3 OS: Program is tested on Windows 10 build 1903 and PopOS 19.04

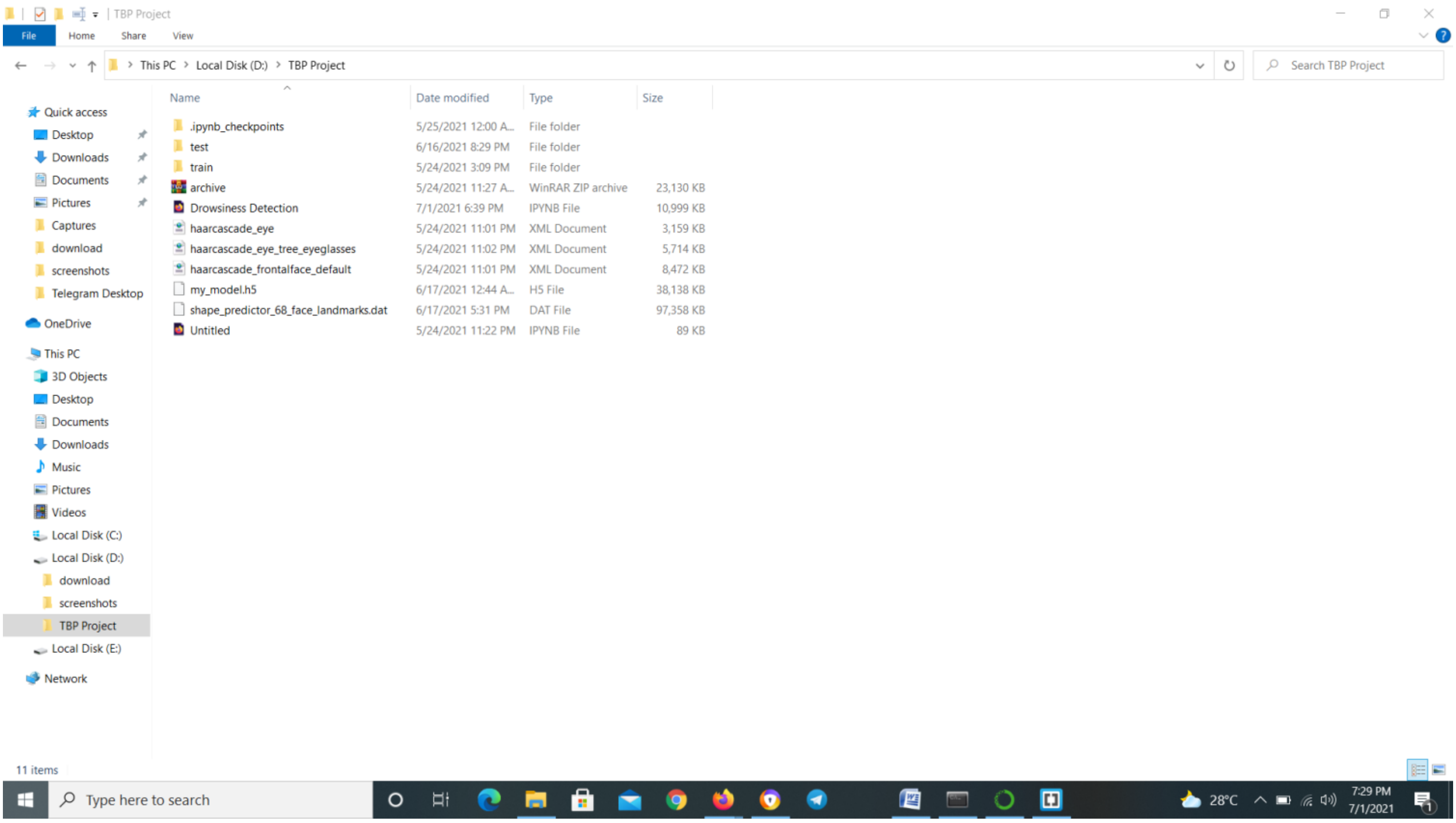
4.3 Laptop: Used to run our code.

4.4 Webcam: Used to get the video feed.

Chapter 5

Implementation

5.1 SYSTEM ARCHITECTURE



5.2 IMPLEMENTATION AND CODE

```
# In[1]:


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


# In[2]:
```

```
img_array=cv2.imread("Train/Open_Eyes/s0012_08716_0_0_1_1_0_02.png",cv2.IMREAD_
GRAYSCALE)


# In[3]:


plt.imshow(img_array,cmap="gray")


# In[3]:


Datadirectory="Train/"
Classes=["Closed_Eyes","Open_Eyes"]
img_size=224


# In[4]:


training_Data=[]
def create_training_Data():
for category in Classes:
path = os.path.join(Datadirectory,category)
class_num=Classes.index(category)
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



# In[5]:


create_training_Data()


# In[6]:
```

```
print(len(training_Data))


# In[6]:


import random
random.shuffle(training_Data)


# In[7]:


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)
Y=np.array(y)



# In[8]:


X.shape


# In[8]:


X=X/255.0;


# In[9]:


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


# In[10]:


model=tf.keras.applications.mobilenet.MobileNet()
```

In[11]:

```
model.summary()
```

In[12]:

```
base_input=model.layers[0].input
```

In[13]:

```
base_output=model.layers[-4].output
```

In[14]:

```
Flat_layer=layers.Flatten()(base_output)
final_output=layers.Dense(1)(Flat_layer)
final_output=layers.Activation('sigmoid')(final_output)
```

In[15]:

```
new_model=keras.Model(inputs=base_input,outputs=final_output)
```

In[ ]:

```
for i,layer in enumerate(new_model.layers):
print(i,layer.name)
```

In[16]:

```
new_model.summary()
```

In[17]:

```
new_model.compile(loss="binary_crossentropy",optimizer="adam",metrics=["accuracy"])


# In[18]:


new_model.fit(X,Y,epochs=1,validation_split=0.1)


# In[ ]:


new_model.save('my_model.h5')


# In[1]:


import tensorflow as tf
new_model=tf.keras.models.load_model('my_model.h5')


# In[2]:



import cv2
import matplotlib.pyplot as plt
import numpy as np
import os
Data_dir="Test/"
path=os.path.join(Data_dir)
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,(224,224))
plt.imshow(new_array)
plt.show()
X_input=np.array(new_array).reshape(1,224,224,3)
X_input=X_input/225.0
prediction=new_model.predict(X_input)
if(prediction==1):
print("Open")
else:
print("close")
```

```
except Exception as e:

pass


# In[2]:


import numpy as np
def lip_distance(shape):
top_lip = shape[50:53]
top_lip = np.concatenate((top_lip, shape[61:64]))

low_lip = shape[56:59]
low_lip = np.concatenate((low_lip, shape[65:68]))

top_mean = np.mean(top_lip, axis=0)
low_mean = np.mean(low_lip, axis=0)

distance = abs(top_mean[1] - low_mean[1])
return distance


# In[14]:


import winsound
frequency=2500
duration=1000
import cv2
from imutils import face_utils
import numpy as np
import dlib
YAWN_THRESH = 30
path="haarcascade_frontalface_default.xml"
faceCascade=cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.
xml')
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
cap=cv2.VideoCapture(1)
if not cap.isOpened():
cap=cv2.VideoCapture(0)
if not cap.isOpened():
raise IOError("Cannot open webcam")

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)
rects = faceCascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30),
flags=cv2.CASCADE_SCALE_IMAGE)
for (x, y, w, h) in rects:
rect = dlib.rectangle(int(x), int(y), int(x + w),int(y + h))
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)
distance = lip_distance(shape)
lip = shape[48:60]
cv2.drawContours(frame, [lip], -1, (0, 255, 0), 1)
for x,y,w,h in eyes:
roi_gray=gray[y:y+h,x:x+h]
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]
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==1):
status="Open Eyes"
x1,y1,w1,h1=0,0,175,75
cv2.rectangle(frame,(x1, x1), (x1+w1, y1+h1),(0,0,0), -1)
cv2.putText(frame, 'Active' , (x1+int(w1/10),y1 + int(h1/2)),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,255,0), 2)
else:
counter = counter+1
status = "Closed Eyes"
cv2.rectangle(frame,(x, y), (x+w, y+h),(0,0,255), 2)
if(counter>5):
x1,y1,w1,h1=0,0,175,75
cv2.rectangle(frame,(x1, x1), (x1+w1, y1+h1),(0,0,0), -1)
cv2.putText(frame, 'Sleep Alert!!' , (x1+int(w1/10),y1 + int(h1/2)),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)
winsound.Beep(frequency, duration)
counter=0
if (distance > YAWN_THRESH):
print("distance=",distance)
cv2.rectangle(frame,(x1, x1), (x1+w1, y1+h1),(0,0,0), -1)
cv2.putText(frame, 'Yawn Alert!!' , (x1+int(w1/10),y1 + int(h1/2)),
```

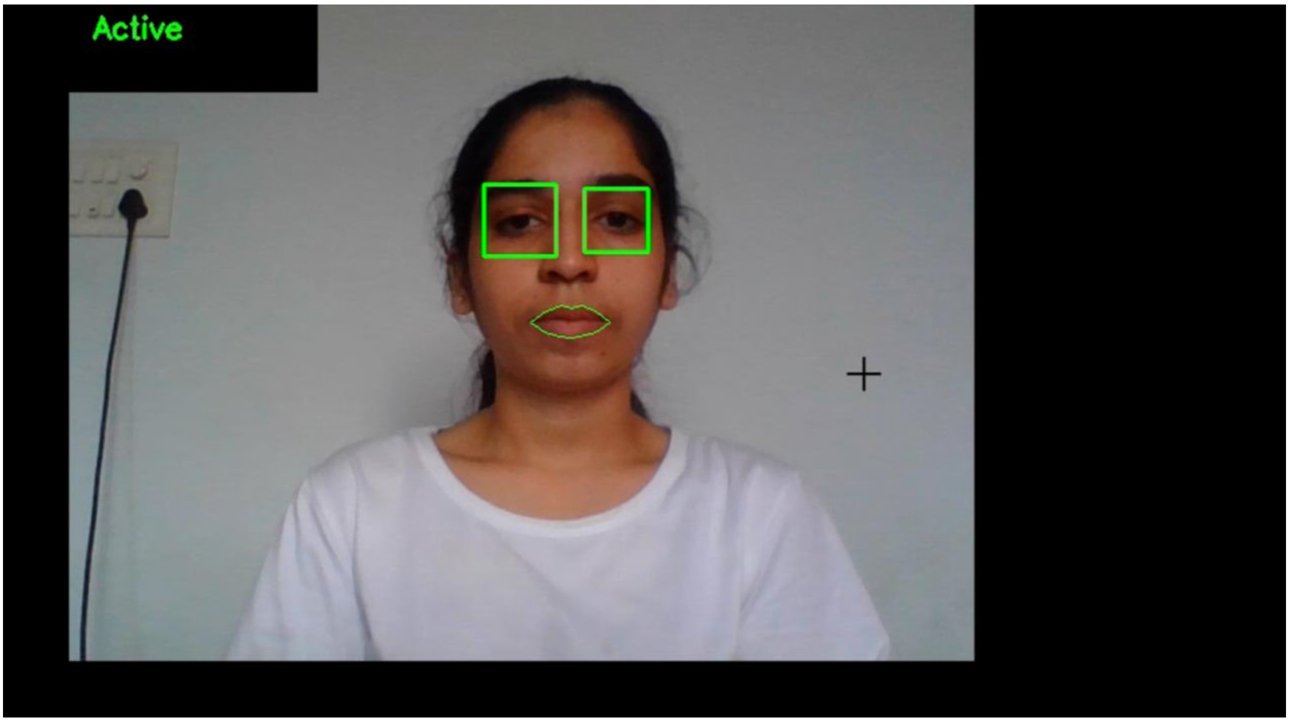
```
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)
winsound.Beep(frequency, duration)


cv2.imshow('Drowsiness detection system',frame)
if cv2.waitKey(2) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
```

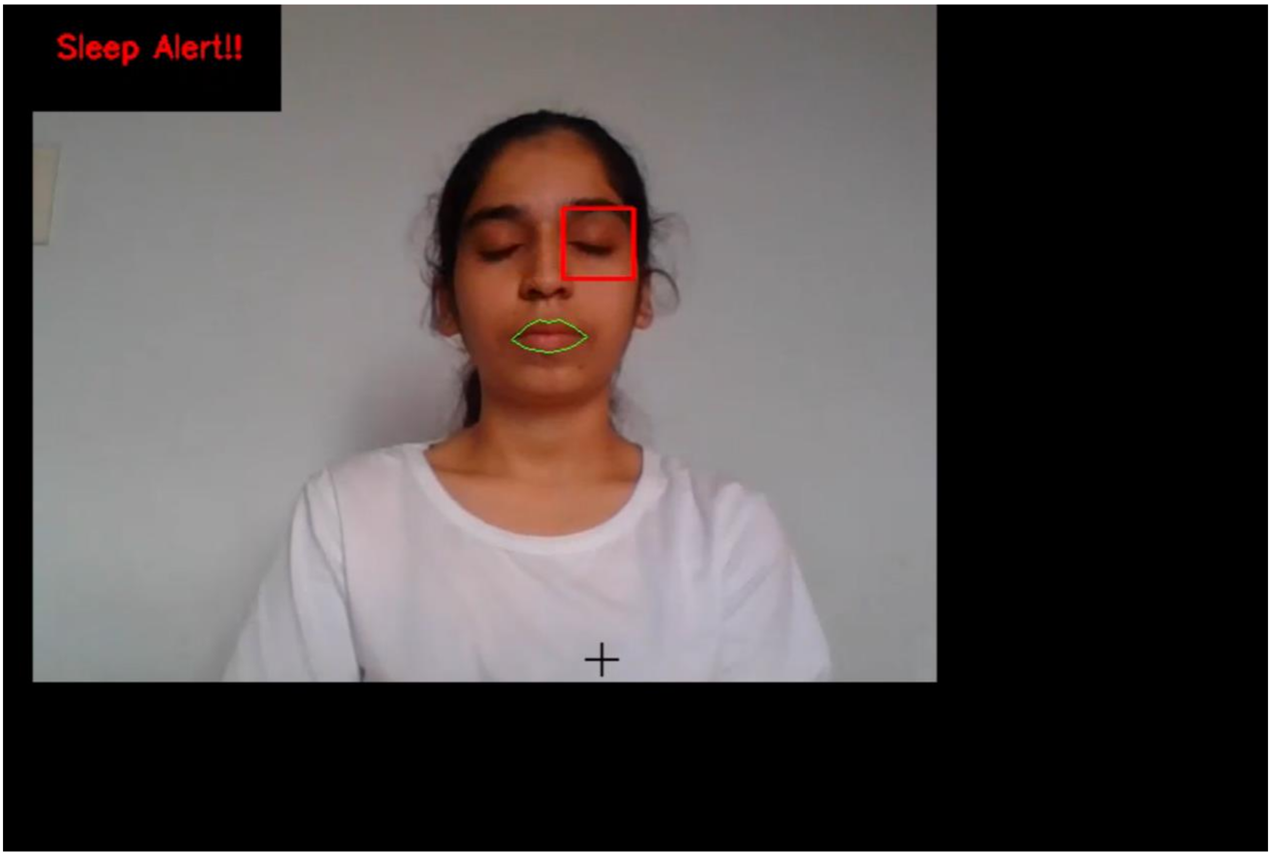
Chapter 6

RESULTS

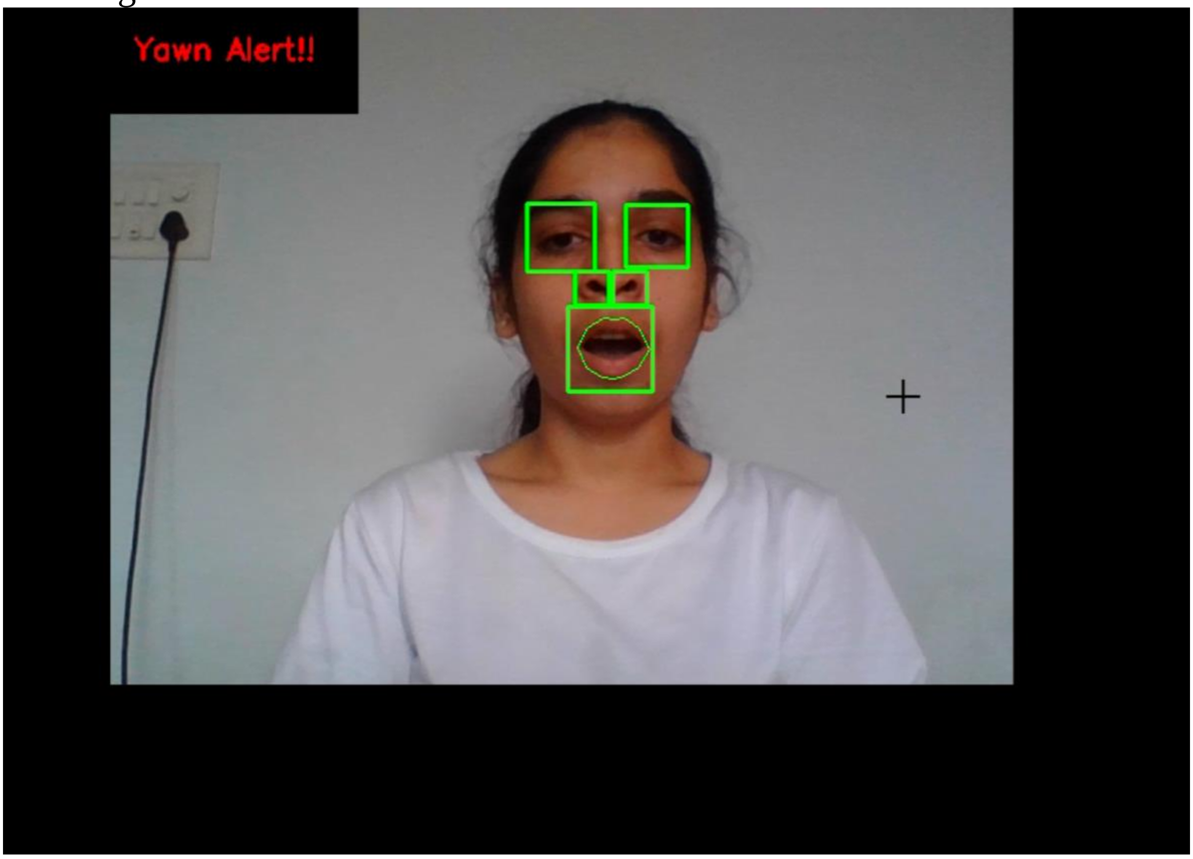
Active state



Sleep Alert



Yawning state



Chapter 7

TESTING

Validation was performed using various closed and open state images.Acceptance of images was tested using png and jpeg pictures.

Chapter 8

Conclusion and Future Scope

10.1 Conclusion

It completely meets the objectives and requirements of the system. The framework has achieved an unfaltering state where all the bugs have been disposed of. The framework cognizant clients who are familiar with the framework and comprehend it's focal points and the fact that it takes care of the issue of stressing out for individuals having fatigue-related issues to inform them about the drowsiness level while driving.

10.2 Future Scope

The model can be improved incrementally by using other parameters like blink rate, yawning, state of the car, etc. If all these parameters are used it can improve the accuracy by a lot.

We plan to further work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers.

Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleep and stop the video accordingly. It can also be used in application that prevents user from sleeping.

Chapter 9

References

https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a
https://www.kaggle.com/prasadvpatil/mrl-dataset
http://dlib.net/face_landmark_detection_ex.cpp.html
https://ieeexplore.ieee.org/document/8808931/figures