SYNOPSIS

The project "**DROWSINESS DETECTION**" deals with the various levels of project development and will account for time used in analysis, design, programming, testing and verification . It is well known fact that software companies undertake huge projects more than one at a time. Hence, there is a profound need for the organizations to manage all the projects efficiently and ensure that projects cycle goes on smoothly and they are completed on time.

The interface of the project work entitled " DROWSINESS DETECTION" is developed using PYTHON.

TABLE OF CONTENTS

CONTENTS   **PAGE NO**

1.INTRODUCTION

The project "**drowsiness detection**" system is done to develop a system to prevent accidents from happening because of driver fatigue and sleepiness. This project will detect that if person's eyes are closed for a few seconds. Then, this system will alert the driver when drowsiness is detected.

The main aim of this project is to reduce the accidents caused due to driver fatique and sleepiness. Our current statistics reveal that just in 2015 in India alone 148,707 people died due to car related accidents. Out 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.

With the advent of modern technology and real time scanning systems using cameras we can prevent major mishaps on the road by alerting car driver who is feeling drowsy through a drowsiness detection system.

By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. Detection of fatigue involves the observation of eye movements and blink patterns in a sequence of images of a face.

SYSTEM SPECIFICATION

Hardware Specification:

Processor: Intel Core i3

Ram: 4 GB

Hard disk: 1 TB

Software Specification:

Operating System: Windows 10 Home (64 bit)

Software:

1.1 Python 3:

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system.

Its implementation began in December 1989.Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Python supports modules and packages, which encourages program modularity and code reuse.

Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python is a very productive language. Due to the simplicity of Python, developers can focus on solving the problem. They don't need to spend too much time in understanding the syntax or behaviour of the programming language. You write less code and get more things done.

1.2 Libraries:

- **Numpy:**

   It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

The NumPy arrays takes significantly less amount of memory as compared to python lists. It also provides a mechanism of specifying the data types of the contents, which allows further optimisation of the code.

An n-dimension array is generally used for creating a matrix or tensors, again mainly for the mathematical calculation purpose.

NumPy is not only about efficient storing the data, it also makes it extremely easy to perform mathematical operations on it.

• **Scipy:**

SciPy was created by NumPy's creator Travis Olliphant. It is a scientific computation library that uses NumPy underneath.

It provides more utility functions for optimization, stats and signal processing.

Most new Data Science features are available in Scipy rather than Numpy.

• **Playsound:**

Pygame is a cross-platform set of Python modules which is used to create video games.

It consists of computer graphics and sound libraries designed to be used with the Python programming language.

It is mainly used for sound libraries.

- **Dlib:**

  It is a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face.

  It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments.

  Dlib along with OpenCV can handle bad and inconsistent lighting and various facial positions such as tilted or rotated faces.

  Dlib is ahead of the Haar cascade classifier over implementation, speed, and accuracy.

Dlib for face detection uses a combination of HOG (Histogram of Oriented Gradient) & Support Vector Machine (SVM) which is trained on positive and negative images meaning some images have faces and ones that don't.

- **Imutils:**

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python.

- **OpenCV:**

OpenCV (OpenSource Computer Vision Library) is an open source computer vision and machine learning software library.

OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products

The library has more than 2500 optimized algorithms, which an be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image

of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery.

The main functionality of OpenCV is Image/video I/O, processing, display, Object/feature detection, Geometry-based monocular or stereo computer vision, Computational photography, Machine learning & clustering, CUDA acceleration (gpu).

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

The existing system of drowsiness detection has very slow rate of detection of blinking eyes, poor face recognisation algorithm, takes more time to make alarm sound.

Sometimes, it may even fail to detect the person in lowlight and continuously beeping the alarm.

DRAWBACKS OF EXISTING SYSTEM:

1. Old sensors which were very slow and has short life span.

2. The algorithms used were too slow and has very high time complexity.

3. Poor camera quality due to which errors occur.

4. Sometimes, due to poor face recognition the alarm sound automatically beeps.

5. In, most of the country's driver drowsiness detection were not implemented. Due, to which development in this sector is very slow.

6. Sometimes, due to error alarm starts beeping and we cannot even switch off or restart device.

3.2 PROPOSED SYSTEM:

 To overcome these difficulties, we have developed a new algorithm which is more accurate and faster. This system works more faster and alerts the person on correct time.

ADVANTAGES OF PROPOSED SYSTEM:

1. Nowadays, the algorithm was very fast and more accurate.

2. New sensors were cheap and have high reliability.

3. The quality of camera was developed so that face recognition becomes more accurate even in low light conditions.

4. In this algorithm, errors were reduced due to which accuracy is increased.

5. The processing speed of algorithm is increased so that the distance between the eyelids were calculated more faster and hence it also increases the accuracy.

6. Now, Alarm sounds starts to beep faster and accurately. Thus, the latency issue has been reduced.

4.SYSTEM DESIGN

4.1 MODULE DESCRPTION:

DATA COLLECTION:

- To detect the faces, HELEN dataset is used to create the model.

- Helen dataset has a total of 7780 images of human faces.

- These types of images were ran into the model to improve its efficiency to get better results.

ANALYSIS MODULE:

- Analysis is done by using various algorithms such as Linear Regression, Logistic Regression, Decision Tree, Naive Bayes, GBM, XGBoost, LightGBM, CatBoost etc.

- Based on these types of the algorithm, report is generated using dlib and scikit library functions.

- The landmarks on the face were analysed by dlib.

PREDICTION MODULE:

- Prediction model is done after running it into analysis through different models.

- After passing our video feed to the dlib frame by frame, we are able to detect left eye and right eye features of the face.

- Now, we drew contours around it using OpenCV.

- Using Scipy's Euclidean function, we calculated sum of both eyes' aspect ratio which is the sum of 2 distinct vertical distances between the eyelids divided by its horizontal distance.
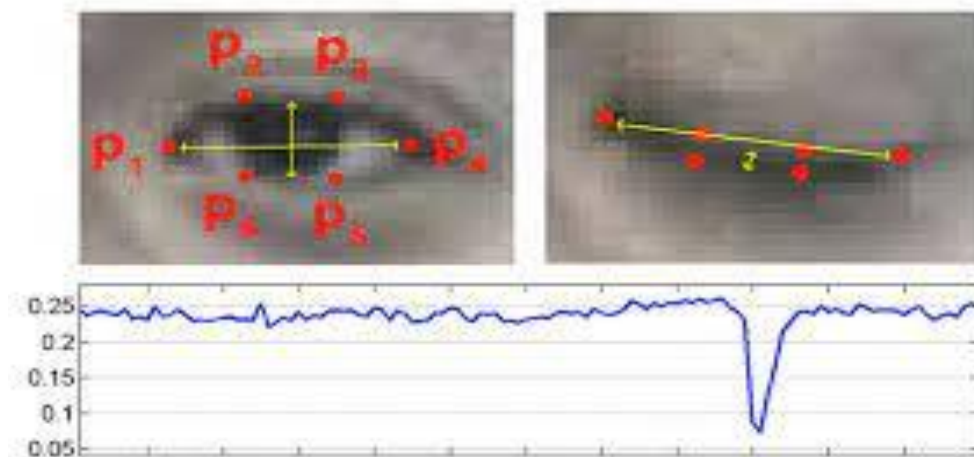
- Now we check if the aspect ratio value is less than 0.25 (0.25 was chosen as a base case after some tests). If it is less an alarm is sounded and user is warned.
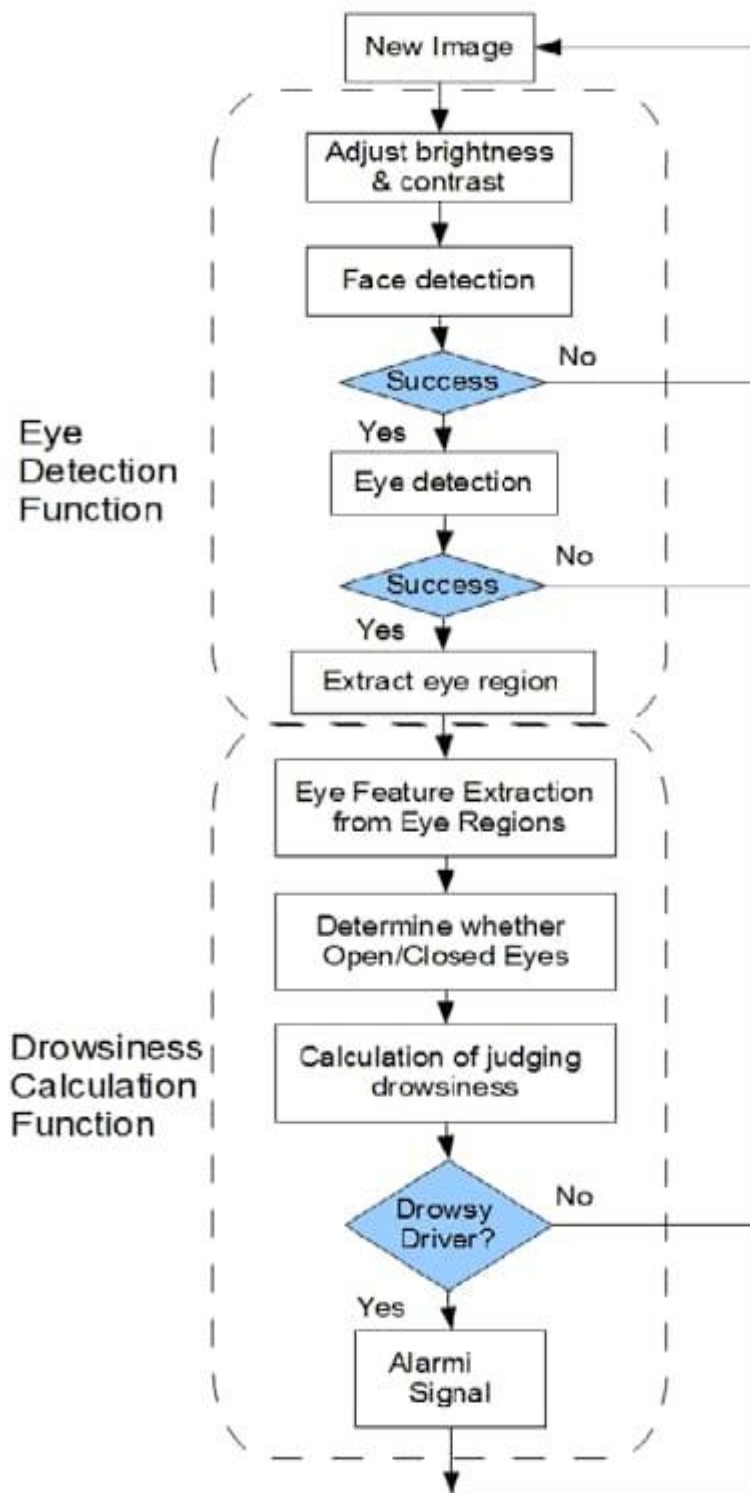
4.2 DATA FLOW DIAGRAM

Data flow diagrams are used to graphically represent the flow of data in a information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.
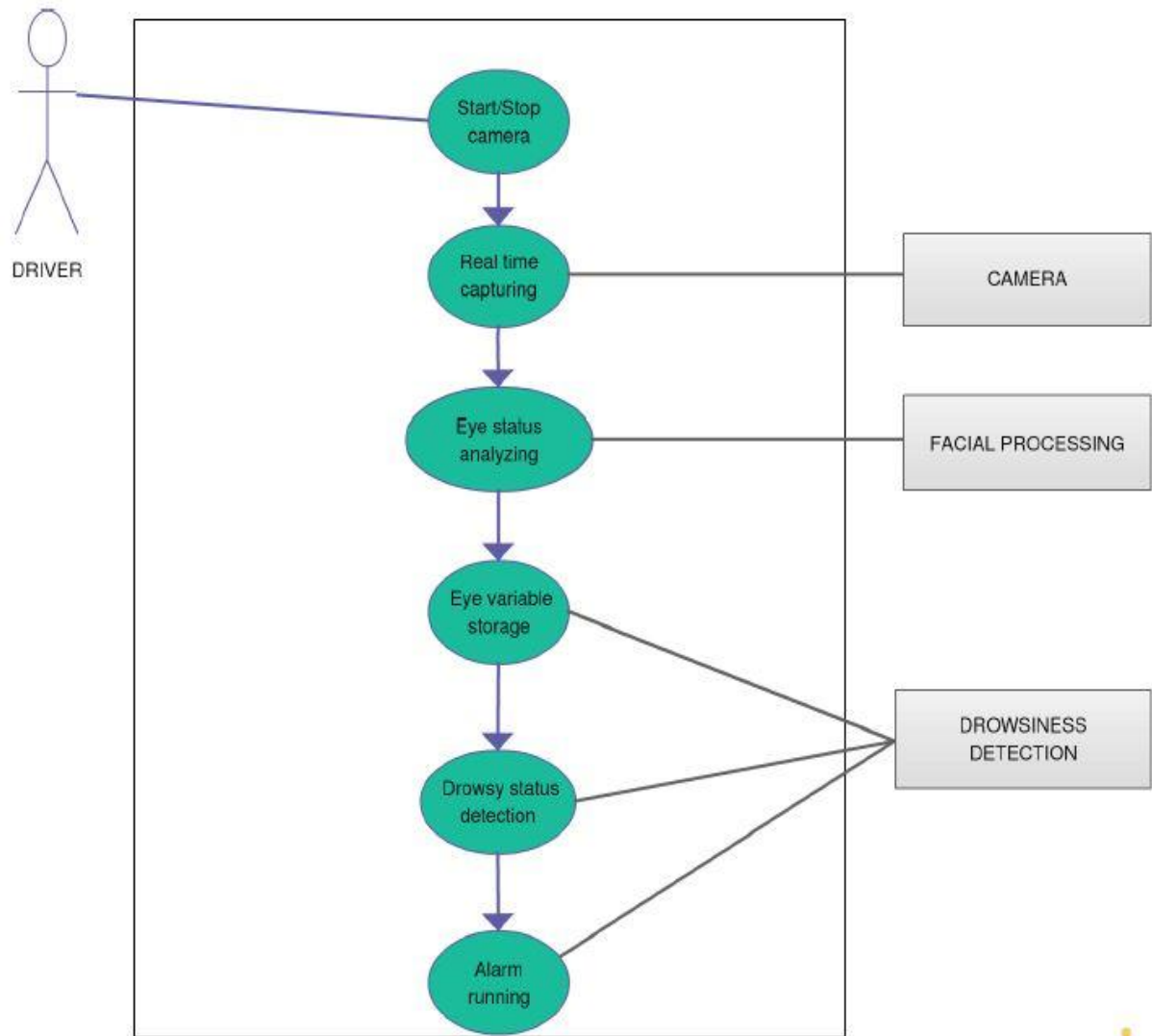
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer.

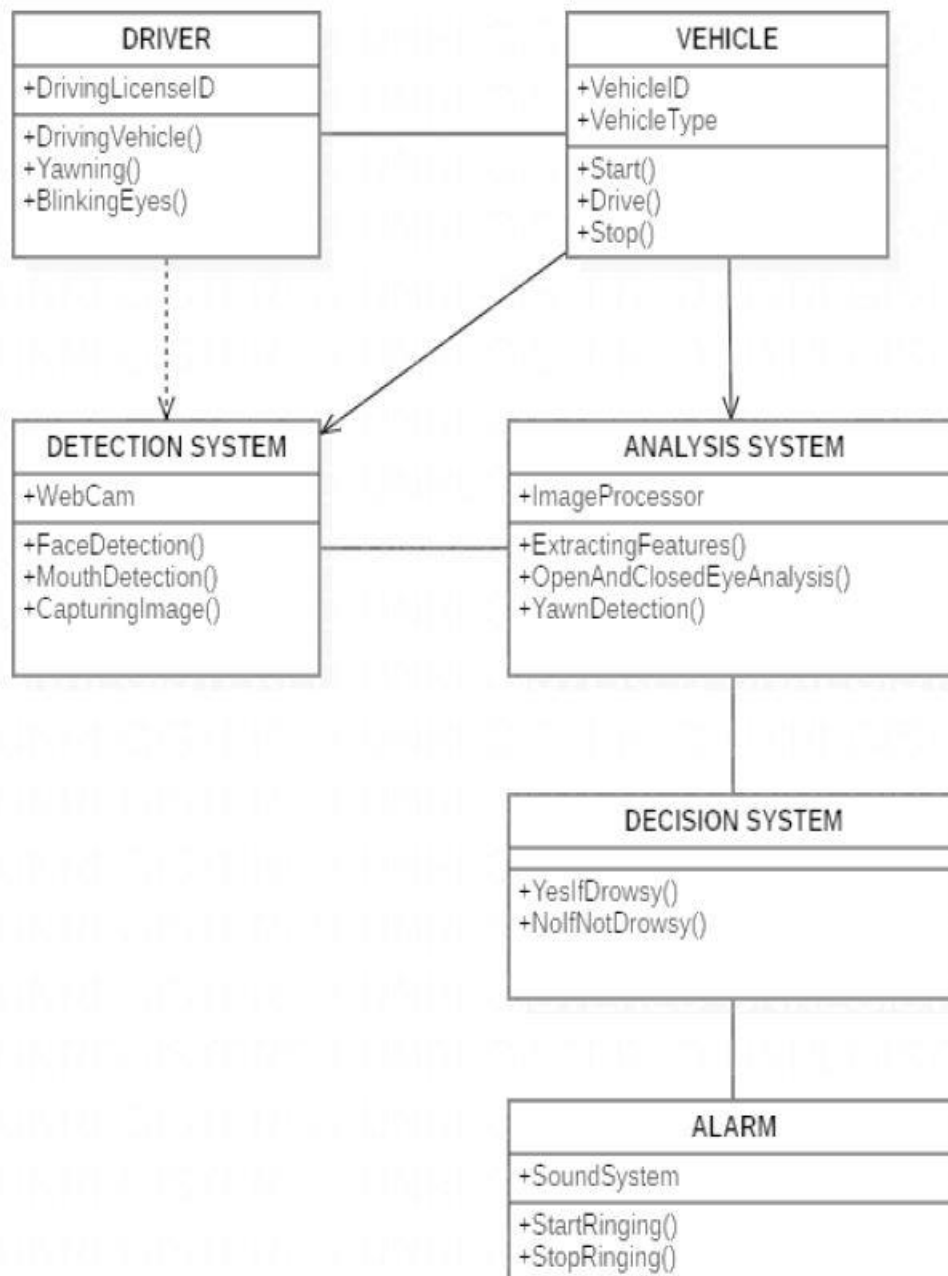DFD has often been used due to the following reasons:

- Logical information flow of the system.

- Determination of physical system construction requirements.

- Simplicity of notation.

- Establishment of manual and automated systems requirements.

- To show the scope and boundaries of a system.

- To show that the whole system has been considered.

- May be used as a communications tool between a systems analyst and any person who plays a part in the system.

- To act as the starting point for redesigning a system.

New Image

Eye Detection Function

Adjust brightness & contrast

Face detection

Success — No

Yes

Eye detection

Success — No

Yes

Extract eye region

Drowsiness Calculation Function

Eye Feature Extraction from Eye Regions

Determine whether Open/Closed Eyes

Calculation of judging drowsiness

Drowsy Driver? — No

Yes

Alarmi Signal

4.3 ENTITY RELATIONSHIP DIAGRAM

DRIVER

Start/Stop camera

Real time capturing — CAMERA

Eye status analyzing — FACIAL PROCESSING

Eye variable storage

Drowsy status detection — DROWSINESS DETECTION

Alarm running

CLASS DIAGRAM:

4.4 DATABASE DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly.

Table design is the process of the producing a detailed data model of a database.

This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database fully attributed data model contains detailed attributes for each.

The term database design can be used to describe many different parts of the design of an overall database system.

Principally, and most correctly, it can be thought of as the logical design of the vase data structure used to store the data.

In the relation model these are the table and views. In an object database the entities and relationship map directly to object classes and named relationships.
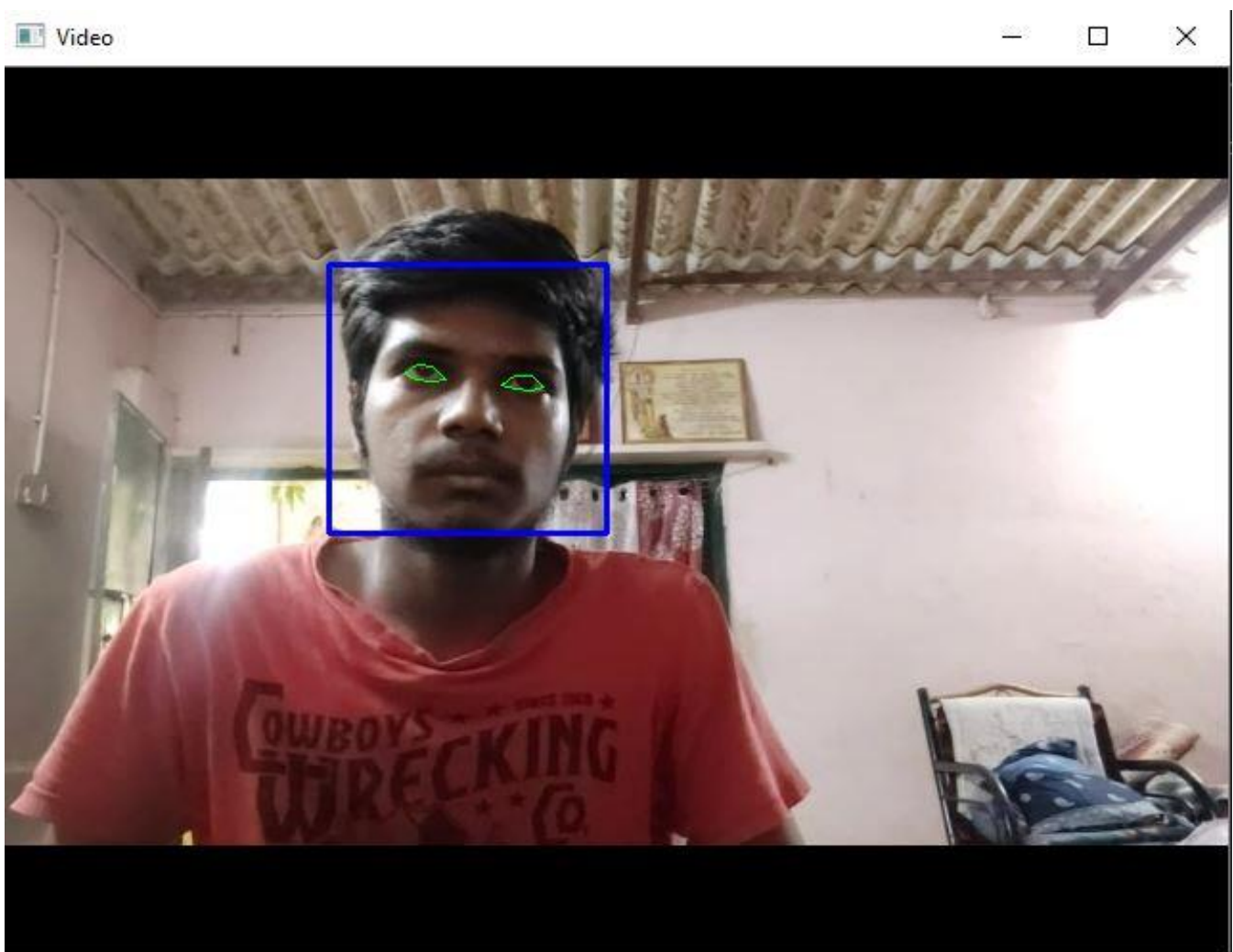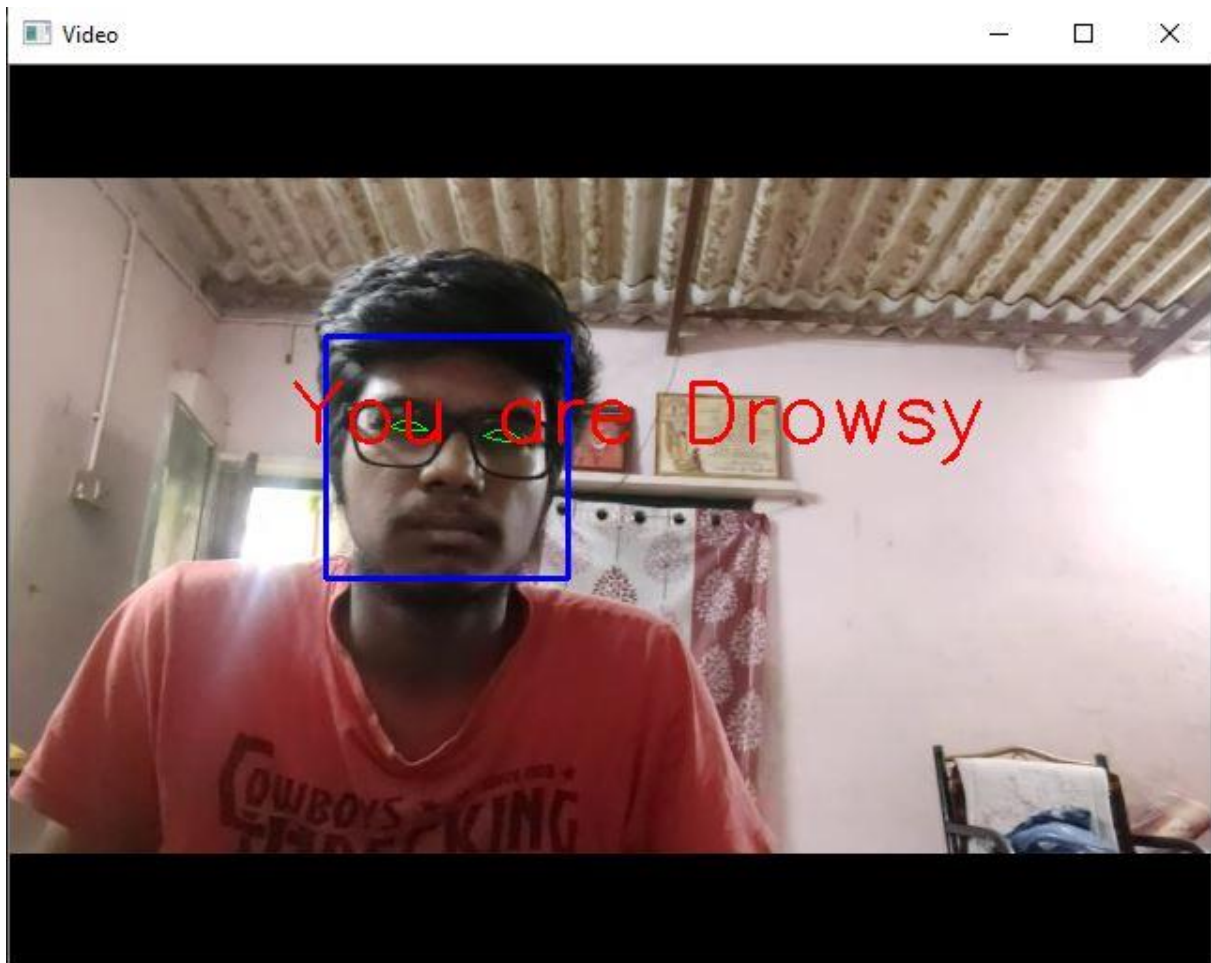
Face marked image by Dlib
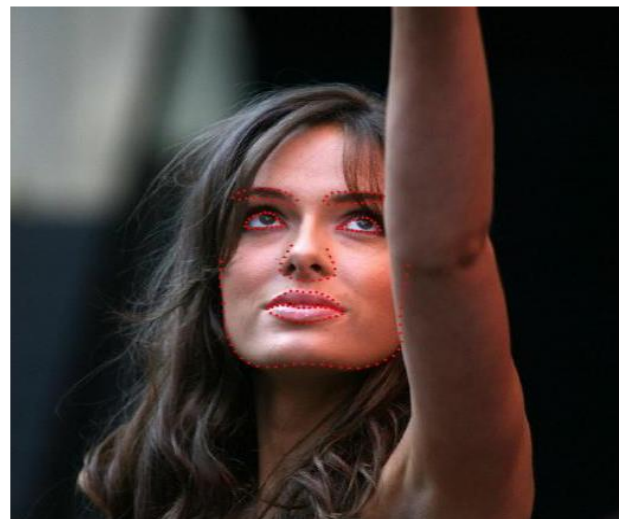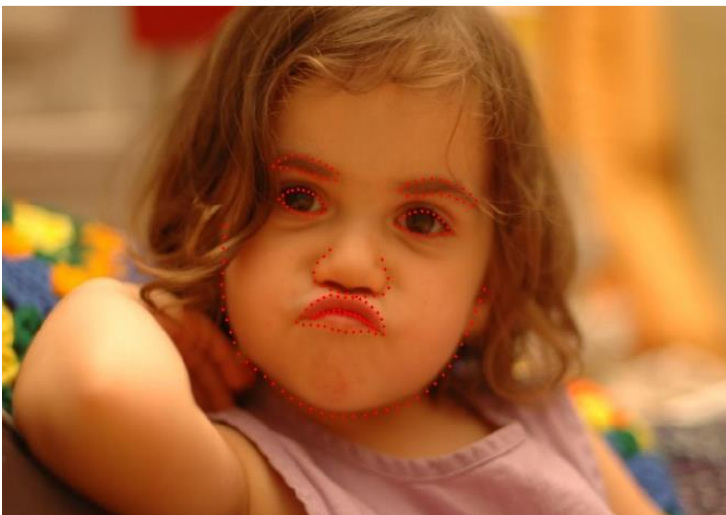
HELEN Dataset Sample Images

4.5 INPUT AND OUTPUT DESIGN

INPUT DESIGN:

OUTPUT DESIGN:

SYSTEM TESTING AND IMPLEMENTATION

5.1 SYSTEM TESTING:

It is the final test to verify that the product to be delivered meets the specifications mentioned in the requirement document.

It is the stage of implementation, which ensures that system works accurately and effectively before the live operation Commences.

It is a confirmation that all are correct and gives opportunity to show the users that the system must be tested with test data and show that the system will operate successfully and produces expected results under expected conditions.

The purpose of system testing to identify and correct errors in the candidate system. As important as this phase is, one is frequently compromised.

Testing is vital to the parts of the system.so that the goal will be successfully achieved.

Inadequate testing or non-testing leads to errors that may not appear until months later.

This creates many problems:

- The time lag between the cause and appearance of the problem.
- The effect of system errors on files and records within the system.

- It may lead to software failure.

- Privacy protection may fall when the databases don't undergo continually checking for breaches.

- It can cause huge crashes in the system that cause a lot of money to repair.

THERE ARE TWO TYOES OF SYSTEM TESTING:

- Black Box Testing.
- White Box Testing.

BLACK BOX TESTING:

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing**.**

These are the steps to be involved:

- Initially, the requirements and specifications of the system are examined.

- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.

- Tester determines expected outputs for all those inputs.

- Software tester constructs test cases with the selected inputs.

- The test cases are executed.

- Software tester compares the actual outputs with the expected outputs.

- Defects if any are fixed and re-tested.

WHITE BOX TESTING:

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing**.**

White box testing involves Internal security holes, Broken or poorly structured paths in the coding processes, The flow of specific inputs through the code, Expected output, The functionality of conditional loops, ,Testing of each statement, object, and function on an individual basis.

UNIT TESTING

In Unit testing, we have to test the programs making up the system. The software units in a system are called modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on the modules, independently of one another, to locate errors. This enables to detect errors in coding and logic that are contained within the module. In the lines of this strategy all the individual functions and modules were put to the test independently. This method was applied in combination with white and black box testing techniques to find the errors in each module.

INTEGRATED TESTING

Again this software testing strategy has two different approaches namely the top-down approach in which the integration is carried out from the top level module to the bottom and the bottom-up approach in which the integration is carried out from the low level modules to the top. The module was tested using the bottom up approach by introducing stubs for the top level functions. This test was used to identify the errors in the interfaces, the errors in passing the parameters between the functions and to correct them.

VALIDATION TESTING

Software validation is achieved through a series of black box test that demonstrate conformity with requirements. Both plan and procedure are designed to ensure that all functional required area achieved. Using validation testing we have tested this project. The user should not leave any input area blank and it is not allowed to enter improper data. Many validations are needed for each and every file in the form like textbox validations.

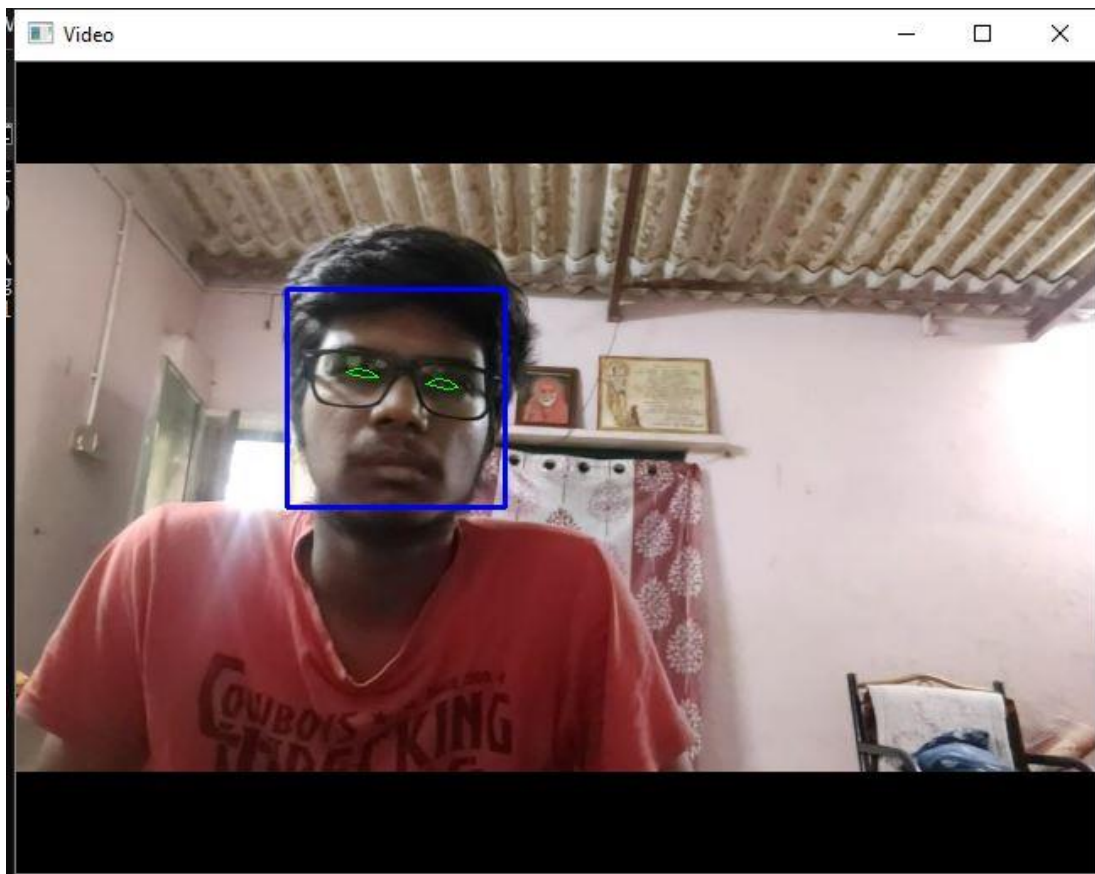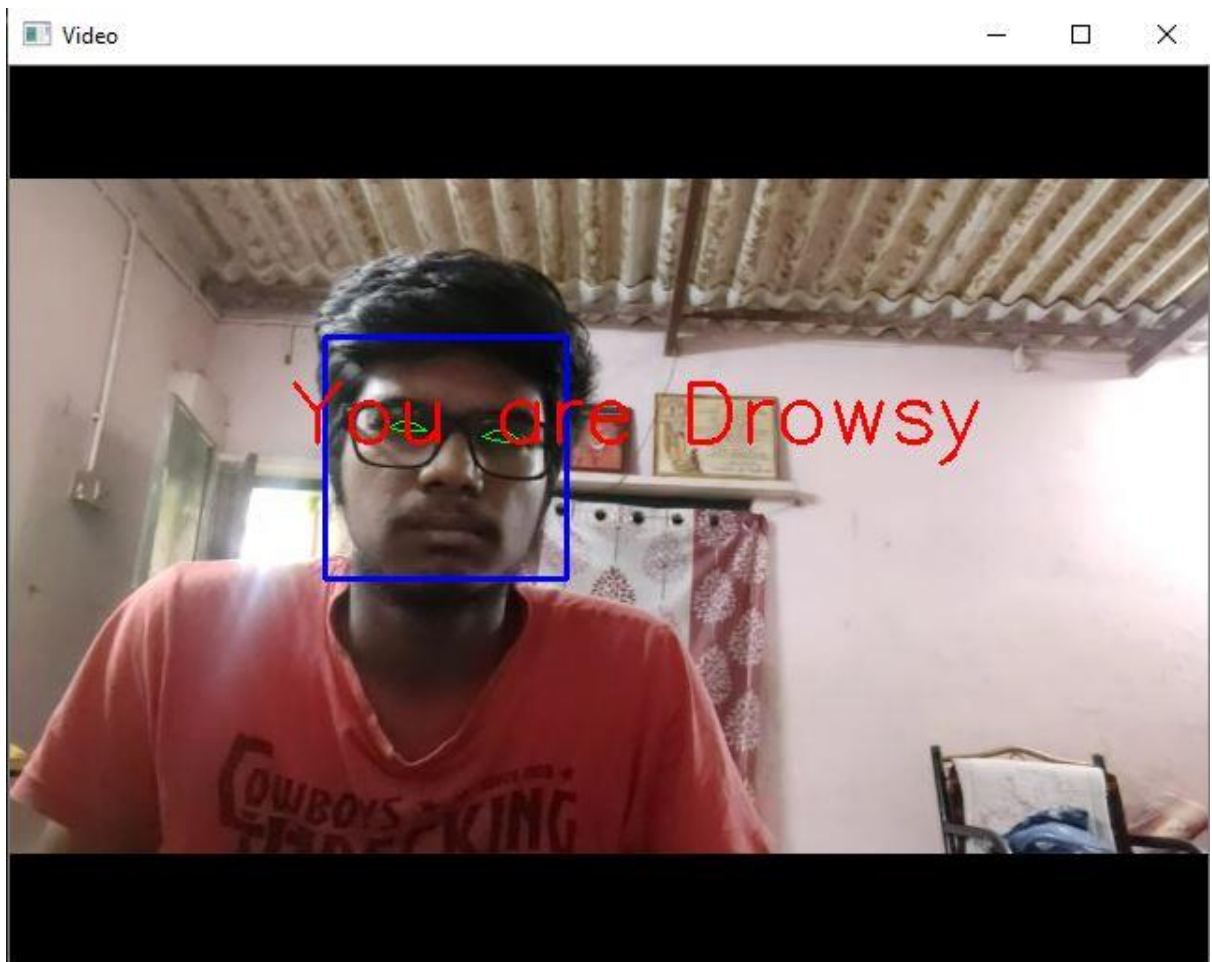| Test No	Test Condition	System Behaviour	Excepted Result
1	Straight face without glasses	Non-Drowsy	Non-Drowsy
2	Straight face without glasses	Drowsy	Drowsy
3	Straight face with glasses	Non-Drowsy	Non-Drowsy
4	Straight face with glasses	Drowsy	Drowsy

Test No: 1
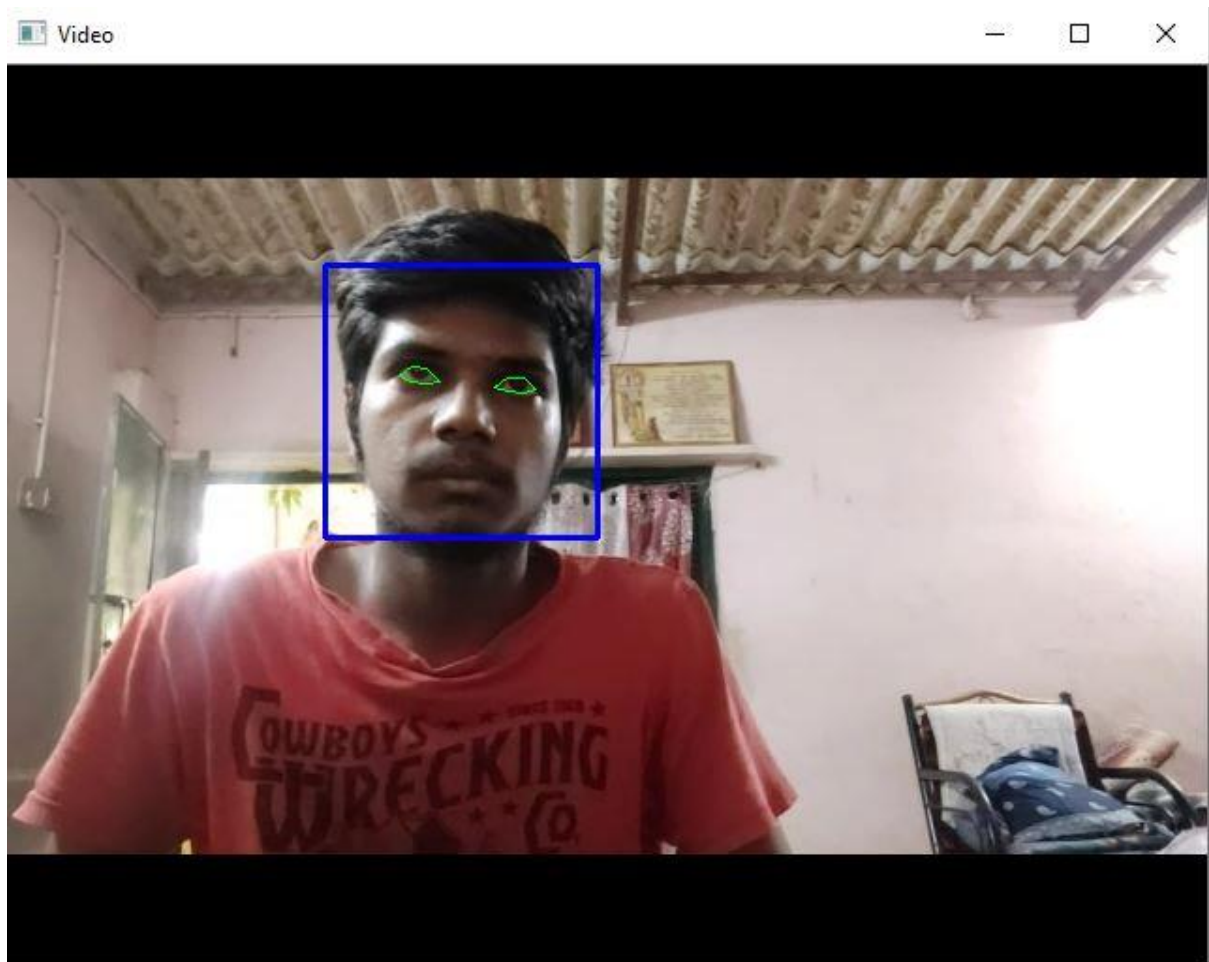
Test No: 2

Test No: 3

Test No: 4

5.2 SYSTEM IMPLEMENTATION

System implementation is the important stage of product where the theoretical design is turned into the practical system. And, it is also a set of procedures performed to complete the design (as necessary) contained in the approved systems design document and to test, install, and begin to use the new or revised Information System. It depicts systems implementation as the fifth major step in the development of an Information System.

The systems implementation goals are as follows:

- Complete as necessary the design contained in the approved systems design document. For example, the detailed contents of new or revised documents, computer screens, and database must be laid out and created.

- Write, test, and document the programs and procedures required by the approved systems design document.

- Ensure, by completing the preparation of user manuals and other documentation and by training personnel, that the organization's personnel can operate the new system.

- Determine, by thoroughly testing the system with users, that the system satisfies the users requirements.

- Ensure a correct conversion by planning, controlling, and conducting an orderly installation of the new system.

-

7. SCOPE FOR FUTURE ENHANCEMENT

In the real time driver fatigue detection system, it is required to slow down a vehicle automatically when fatigue level crosses a certain limit. Instead of threshold drowsiness level it is suggested to design a continuous scale driver fatigue detection system. It monitors the level of drowsiness continuously and when this level exceeds a certain value a signal is generated which controls the hydraulic braking system of the vehicle.

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.

This type of detector can be used wherever we need. For example, in mines many accidents are happening due to drowsiness which cause may losses to the company.

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.

CONCLUSION

The project titled "**Drowsiness Detection**" is developed to improve the efficiency in detecting drowsiness of the driver due to which accidents may happen.

The Drowsiness Detection System developed based on eye closure of the driver can differentiate normal eye blink and drowsiness and detect the drowsiness while driving.

The proposed system can prevent the accidents due to the sleepiness while driving. The system works well even in case of drivers wearing spectacles and even under low light conditions if the camera delivers better output.

Information about the head and eyes position is obtained through various self-developed image processing algorithms.

During the monitoring, the system is able to decide if the eyes are opened or closed.

When the eyes have been closed for too long, a warning signal is issued. processing judges the driver's alertness level on the basis of continuous eye closures.

8. BIBLIOGRAPHY

- COMPUTATIONALLY EFFICIENT FACE DETECTION; B. SCHLKOPF-A. BLAKE, S. ROMDHANI, AND P. TORR.

- INTRODUCTION TO PROGRAMMING USING PYTHON; Y. DANIEL ILANG.

- USE OF THE HOUGH TRANSFORMATION TO DETECT LINES AND CURVES IN PICTURE; R. DUDA AND P. E. HART.

- OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI.
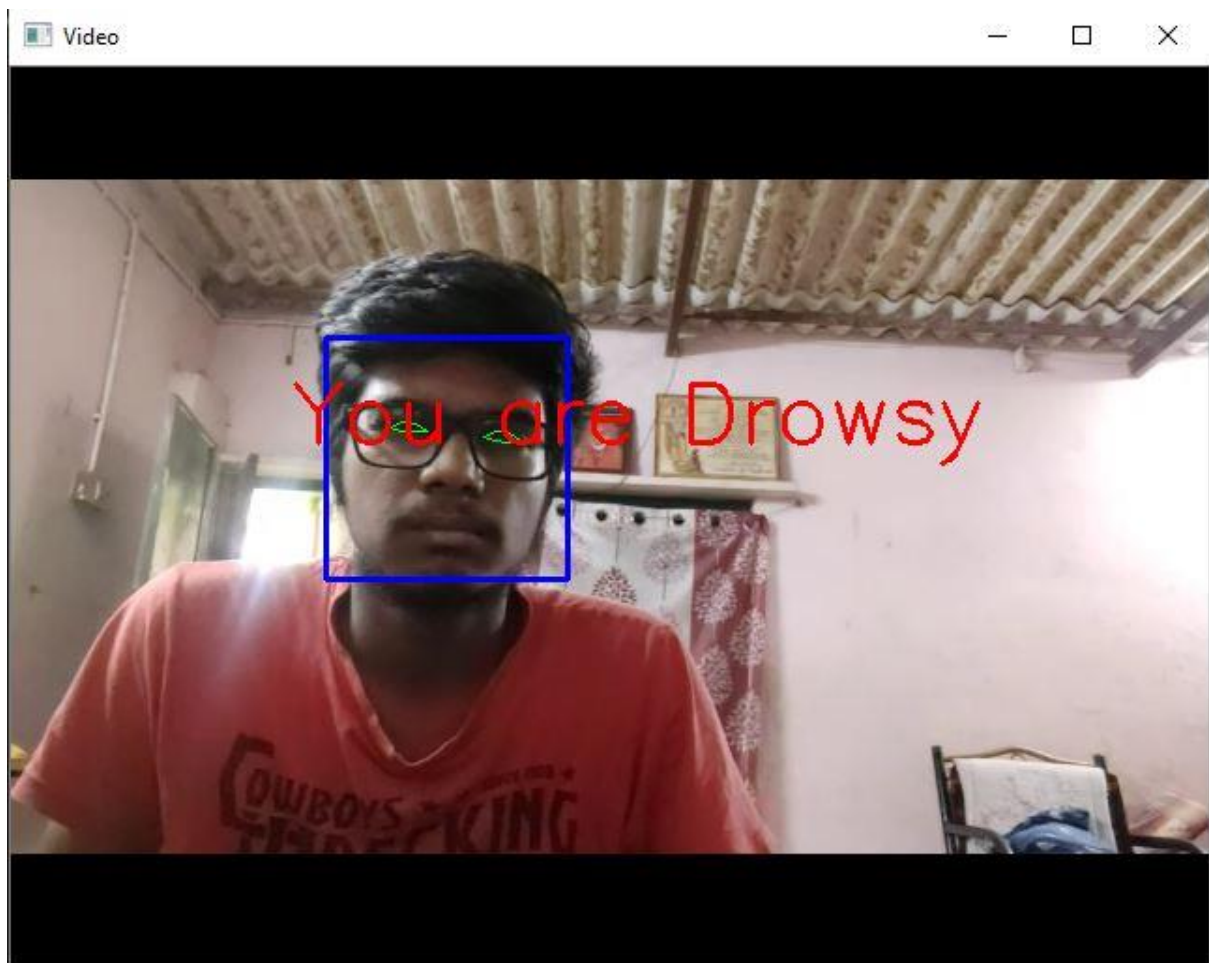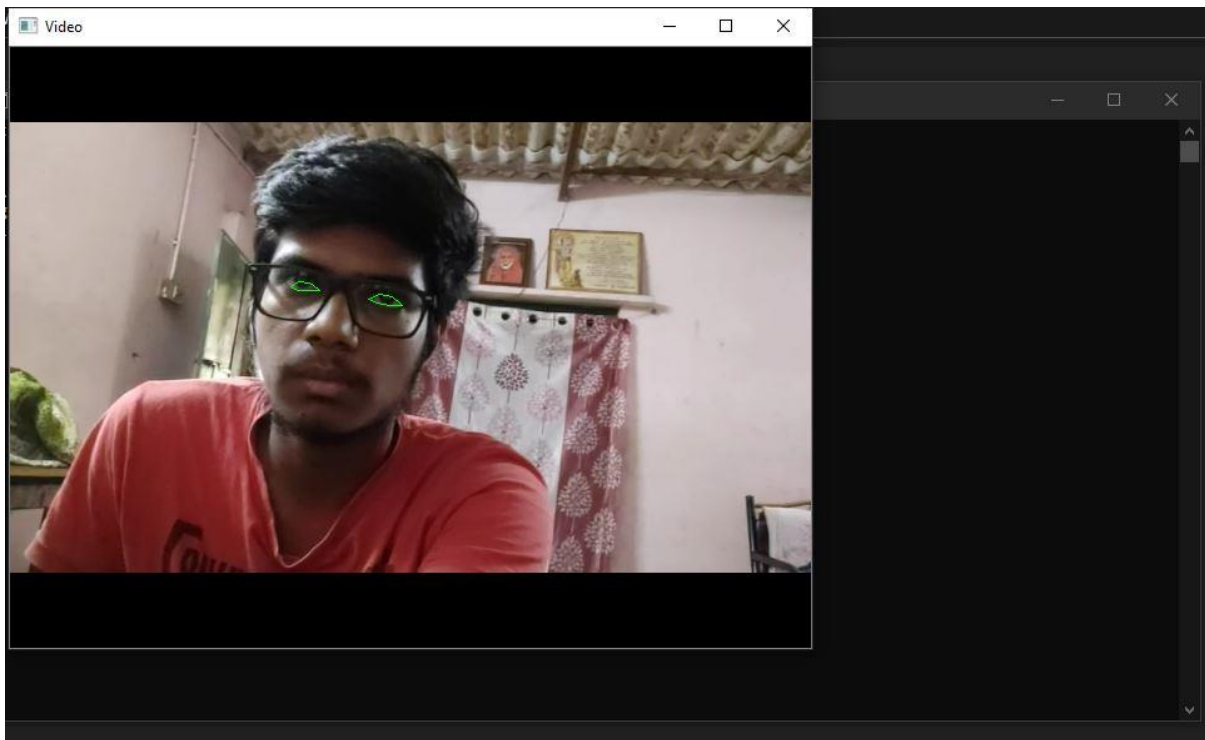
WEBSITES REFERRED:

- https://www.scribd.com/book/15491045/Learning-OpenCV-Computer-Vision-with-the-OpenCV-Library

- https://www.scribd.com/document/46566105/opencv

- www.tutorialspoint.com

- https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

9. APPENDIX

9.2 SCREENSHOTS