DRIVER DROWSINESS DETECTION SYSTEM

A PROJECT REPORT

submitted By

ALEN SAM CHERIAN

TVE20MCA-2007

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the degree

of

Master of Computer Applications

Department of Computer Applications

College of Engineering

Trivandrum-695016

FEBRUARY 2022

Declaration

I undersigned hereby declare that the project report DRIVER DROWSINESS DETECTION SYSTEM, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University Kerala is a bonafide work done by me under supervision of Assistant Prof. Priya S A. This submission represents my ideas in my words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum **ALEN SAM CHERIAN**

Date : 06/03/2022

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **DRIVER DROWSINESS DETECTION SYSTEM** submitted by **ALEN SAM CHERIAN** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Assistant Prof. Priya S A
Project Guide

Prof. Deepa S S
Head of the Department

Acknowledgement

If words are considered as symbols of approval and tokens of acknowledgement, then let words play the heralding role of expressing our gratitude.

First of all we would like to thank **God** almighty for bestowing with wis- dom,courage and perseverance which had helped to complete this project AI Music Composer. This project has been a reality as a result of the help given by a large number of personalities.

I would like to thank **Dr Jiji C V**, Principal, College of Engineering Trivandrum, who helped me during the entire process of work.

I am extremely grateful to **Prof. Deepa S S**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

We express our sincere thanks to **Assistant Prof. Priya S A**, Department of Computer Applications, College of Engineering Trivandrum for his valuable guidance, support and advices that aided in the successful completion of our project.

I profusely thank other Asst. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this seminar. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

ALEN SAM CHERIAN

ABSTRACT

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving. The objective of this intermediate Python project is to build a drowsiness detection system that will detect that a person's eyes are closed for a few seconds. This system will alert the driver when drowsiness is detected. In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into a process which will whether the person's eyes are 'Open' or 'Closed'. The model we used is built with Keras using Convolutional Neural Networks (CNN). A convolutional neural network is a special type of deep neural network which performs extremely well for image classification purposes. A CNN basically consists of an input layer, an output layer and a hidden layer which can have multiple numbers of layers. A convolution operation is performed on these layers using a filter that performs 2D matrix multiplication on the layer and filter

Contents

List of Figures

Chapter 1

Introduction

The attention level of driver degrades because of less sleep, long continuous driving or any other medical condition like brain disorders etc. Several surveys on road accidents says that around 30 percent of accidents are caused by fatigue of the driver. When driver drives for more than normal period for human then excessive fatigue is caused and also results in tiredness which drives the driver to sleepy condition or loss of consciousness. With the ever-growing traffic conditions, this problem will further deteriorate. For this reason, it is necessary to develop driver alertness system for accident prevention due to Driver Drowsiness is a complex phenomenon which states that there is a decrease in alerts and conscious levels of the driver. Though there is no direct measure to detect the drowsiness but several indirect methods can be used for this purpose. This system is a real time system which captures image continuously and measures the state of the eye according to the specified algorithm and gives warning if required. The system provides easiness to users and have wide area of future enhancement. The focus in this project is detection using a low-cost camera. This would mean that the program does not only work with expensive technology such as infrared cameras or other such cameras.

Chapter 2

Requirement Analysis

2.1 Preliminary Investigation

Preliminary investigation is a problem-solving activity that requires intensive communications between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which decisions about the strategies to be followed for effective system study and analysis can be taken at preliminary investigation an initial picture about the system working is got from the information got from the study, the data collection methods were identified. Right from the investigation about the system many existing drawbacks of the system could be identified, which helped a lot in the later stages of more rigorous study and analysis of the manual system. The most critical phase of managing system projects is planning. To launch a system investigation, we need a master plan detailing the steps to be taken, the people to be questioned and the outcome expected.

2.2 Existing System

By using a non-intrusive machine vision-based concepts, drowsiness of the driver detected system is developed Here we are used SVM (support vector machine) to classify the components in the input video. While cropping the region of interest components in the video is not accurate. Sometimes it will show regions wrong. To sense the eyes first we have to create boundary boxes for that and a classification algorithm. The algorithm of SVM will not support in the existing system. The system is less trained and the dataset resources are least accurate compared with the modern hardware support. Hence existing system is not applicable for

practical use. In order to conquer the problem of existing system, new detection system is developed in this project work.

2.3 Proposed System

The proposed system covers all the major drawbacks of existing system. The system provides high-definition accuracy by means of image processing technique implemented with Python Deep learning algorithms. The system is completely software based and requires only webcam interface unlike any other sensors. The system is an always On-line type once it's been activated. The system is able to provide cent percentage accuracy inside vehicle environments. There are several different algorithms and methods for eye tracking, and monitoring. Most of them in some way relate to features of the eye (typically reflections from the eye) within a video image of the driver. The original aim of this project was to use the retinal reflection as a means to finding the eyes on the face, and then using the absence of this reflection as a way of detecting when the eyes are closed. Applying this algorithm on consecutive video frames may aid in the calculation of eye closure period. Eye closure period for drowsy drivers are longer than normal blinking. It is also very little longer time could result in severe crash. So we will warn the driver as soon as closed eye is detected.

2.4 Feasibility Analysis

Feasibility analysis is a system proposal according to its workability, impact on the organization, ability to meet client and user needs and efficient use of resources. The key considerations that are involved in the feasibility analysis are:

- Economic feasibility
- Technical feasibility
- Behavioural feasibility

2.4.1 Economic Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. Most commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system, otherwise further

alterations will have to be made, if it is to have a chance of being approved. The proposed system is cost effective because of its experimental and user-friendly interface. The administrator can directly view and change the records.

2.4.2 Technical Feasibility

Technical feasibility examines the work for the project be done with correct equipment, existing software technology and available personnel. The important advantage of the system is that it is platform independent.

2.4.3 Behavioural Feasibility

The proposed project would be beneficial to all organizations that, it satisfies the objectives when developed and installed. All the behavioural aspects are considered carefully, thus the project is behaviourally feasible and it can also be implemented easily and is very user friendly. The project is inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication, authorization and security of sensitive data of organization.

2.5 Advantages of Proposed System

· High Accuracy

· Less cost

· Easy installation

· It is non-invasive in nature

· Completely image depended

· Decreasing road accidents

Chapter 3

System Configuration

3.1 Hardware Specification

The selection of hardware is very important in the and proper working of an existence software. When selecting hardware, the size and capacity requirements are also important. Below is some of hardware that is required by the system:

· Processor: Dual core processor of 2.0 GHz or more

· RAM: 4 GB DDR4

· Hard Disk Space: 2 GB free hard disk space

· Input Device: Mouse, Standard Keyboard and Webcam 2.0

· Output Device: Monitor with 1280 x 720 resolution

3.2 Software Specification

We require much different software to make the application which is in making to work efficiently. It is very important to select the appropriate software so that the software works properly. Below are the software that are required to make the new system.

· Operating System: Windows 10 / 8.1

· Front-End: Python-OpenCV

3.2.1 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. 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. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.2.2 Open CV

OpenCV (Open-Source 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. Being a BSD licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of the-art computer vision and machine learning algorithms. These algorithms can 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 and establish markers to

overlay it with augmented reality, etc.

OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many start-ups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

3.2.3 Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python. The name Tkinter comes from Tk interface. Tkinter was written by Steen Lumholt and Guido van Rossum, then later revised by Fredrik Lundh.Tkinter is free software released under a Python license.

3.2.4 Dlib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. 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's open-source licensing allows you to use

it in any application, free of charge. Good unit test coverage. The ratio of unit test lines of code to library lines of code is about 1 to 4. The library is tested regularly on MS Windows, Linux, and Mac OS X systems. However, it should work on any POSIX system and has been used on Solaris, HPUX, and the BSDs. No other packages are required to use the library. Only APIs that are provided by an out of the box OS are needed.

There is no installation or configure step needed before you can use the library. See the How to compile page for details.All operating system specific code is isolated inside the OS abstraction layers which are kept as small as possible. The rest of the library is either layered on top of the OS abstraction layers or is pure ISO standard C++. For Image processing Routines for reading and writing common image formats. Automatic color space conversion between various pixel types Common image operations such as edge finding and morphological operations Implementations of the SURF, HOG, and FHOG feature extraction algorithms. Tools for detecting objects in images including frontal face detection and object pose estimation. High quality face recognition.

Chapter 4

System Design

4.1 Introduction

System design is the process of defining the architecture, components, modules, interface and data for a system to satisfy specified requirements. It a solution to an approach compared to system analysis which is it translates these "what is" orientation. System requirements into way of making them operational. The design phase focuses on detailed implementation of the system recommended in the feasibility study. Planning of system or to replace or complement an existing system. But before this, planning should be done. It must be thoroughly understood about the old system and determine how computers can make its operations more effective. The importance of system design is the quality. Design is the place where quality is fostered in the software development. Design representation of software provides us with that can be assessed for quality. System design is a transaction from a user-oriented documents to a programmer or database personal. It is a creative activity in both art and technology. It involves the following procedures, they are

· Database Design

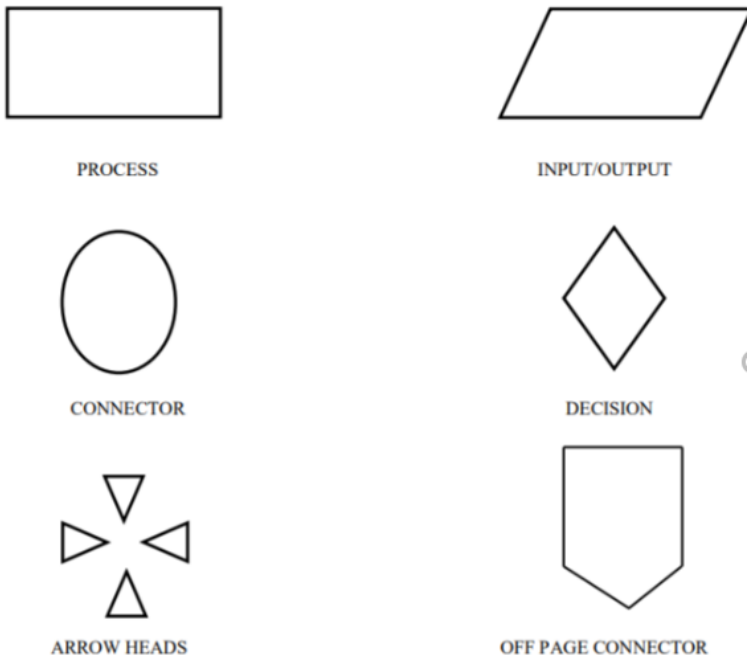
· Input Design

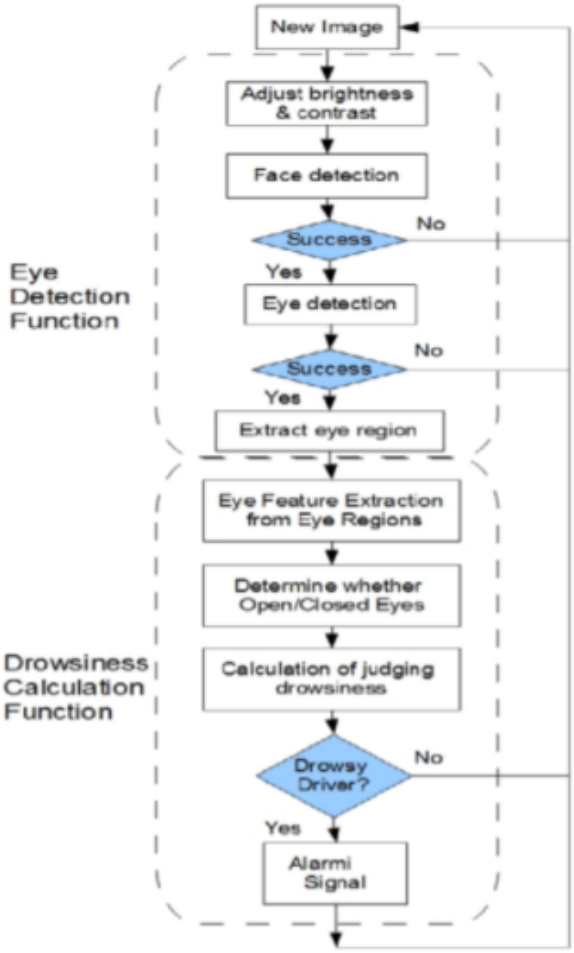
· Output Design

4.2 System Flowchart

The flowchart is a graphic technique specifically developed for using dataflow. The flowchart is a pictorial representation that uses predefined symbols to describe dataflow of a system about its logic. Flowcharts were first used in the early 20th century to describe engineering and manufacturing systems. With the rise of computer programming, the system flowchart has become a valuable tool for depicting the flow of control through a computer system and where decisions are made that affect the flow. Computer programming requires careful planning and logical thinking. Programmers need to thoroughly understand a task before beginning to code. System flowcharts were heavily used in the early days of programming to help system designers visualize all the decisions that needed to be addressed. Other tools have since been introduced that may be more appropriate for describing complex systems. One of these tools is pseudocode, which uses a combination of programming language syntax and English-like natural language to describe how a task will be completed. Many system designers find pseudocode easier to produce and modify than a complicated flowchart. However, flowcharts are still used for many businesses application.

BASIC SYMBOLS





4.3 Data Flow Diagram

The data flow diagram (DFD) is one of the most important tools used by system analysis. A DFD is also known as "Bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design phase. So, it is the starting point of the design phase that functionally decomposes the requirement specifications down to the lowest level of detail. Data flow diagrams are made up of a number of symbols, which represent system components. Most data flow modelling methods use four kinds of symbols. These symbols used to represent four kinds of the system components. Processes, data stores, data flows and external entities. Circles in DFD represent processes. Data flow is represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities. Constructing a DFD Several rules of thumb are used in drawing a DFD. Process should be named and numbered for easy reference. Each name should be representative of the process. The direction of flow is from top to bottom and left to right. When a process is exploded

into lower-level details, they are numbered. The names of data stores, sources and destinations are written Process and data flow in capital letters. Names have the first letter of each word capitalized. To construct a, DFD we use,

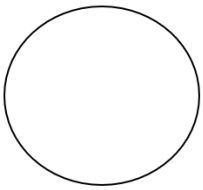
- Arrow
- Circles
- Pen Ended Box
- Squares

An arrow identifies the data flow in motion. It is pipeline through which information is flown like the rectangle in the flow chart. A circle stands for process that converts data into information. An open-ended box represents a data store, data at rest or a temporary repository of data. A square defines a source or destination of system data.

Five rules for constructing a DFD

- Arrows should not cross each other
- Squares, circles and files must be names
- Decomposed data flow squares and circles can have same names.
- Choose meaningful names for data flow
- Draw all data flows around the outside of the diagram :

An arrow identifies the Data Flow in motion.
It is a pipeline through which information is flown like
the rectangle in the flowchart.

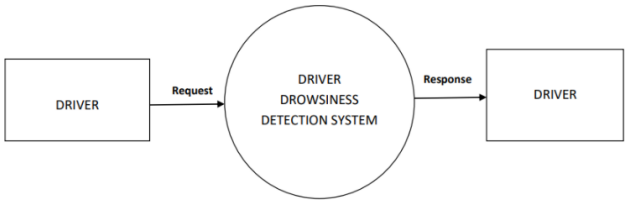


A circle stands for that converts data into information

An open-ended box represents a data store, data at test or
a temporary repository of data.

A square defines a source or destination of system

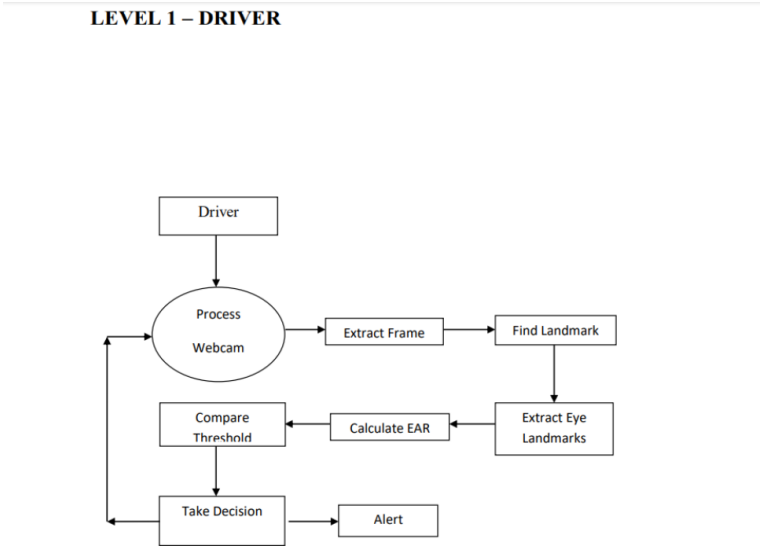
CONTEXT LEVEL



4.4 Unit testing

4.5 Input Design

Input design is one of the most expensive phases of the operation of computerized system and often the major problem of a usually. A larger number of problems with a system can be traced back to fault input design and methods. Needless to say, therefore that output data is the block of a system and has to be analysed and designed consideration. It is the process of converting the user-oriented description of into a computer-based business information system inputs of input design are to create to a programmer-oriented specification. The objective errors. An input layout that is easy to follow and prevent operator. It covers all phases of input from creation of initial data into actual entry of the data to the system for processing. The input design is the link that ties the system into world of its users. The user interface design is very important for any application. The interface design defines how the software communication within itself, to system that interpreted with it and with human who use it. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input



design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, also considered for the development of the project. At right time are

Requirements of Form Design:

– Identification and wording.

– Maximum readability and use

– Physical factors

– Order of data items.

– Easy of data entry

– Size and arrangement.

– Use of instructions

4.6 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the user and to the other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Thus, output design generally refers to the result and information that are generated by the system. For many ends users' output is the main reason for developing the system and the basis on which they are evaluate the usefulness of application.

The objective of a system finds its shape in terms of the output. The analysis of the objective of the system leads to determination of outputs. Outputs of a system can take various forms. The most common are reports, Screens, Printed form, Animations etc. The outputs also vary in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be are all considered. The output forms a system in the justification for its existence. If the outputs are inadequate in anyway, the system itself is inadequate. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner. The outputs are in the form of reports When designing output, the system analyst must accomplish things like, to determine what information to be present, to decide whether to display or print the information and select the output medium to distribute the output to intended recipients. The output is the most important and direct source of information to the user. So, it should be provided in a most efficient formatted way. An efficient and intelligent output of the system improves the relationship between the user and the system and help in decision making.

Chapter 5

Implementation and Testing

5.1 Implementation Approaches

The implementation is one phase of software development. Implementation is that stage in the project where theoretical design is turned into working system. Implementation involves placing the complete and tested software system into actual work environment. Implementation is concerned with translating design specification with source code. The primary goal of implementation is to write the source code to its specification that can be achieved by making the source code clear and straight forward as possible. Implementation means the process of converting a new or revised system design into operational one. The implementation is the final stage and it's an important phase. It involves the individual programming, system testing, user training and the operational running of developed proposed system that constitute the application subsystems. One major task of preparing for implementation is education of users, which should really have been taken place much earlier in the project when they were being involved in the investigation and design work. During this implementation phase system actually takes physical shape. Depending on the size of the organization and its requirements the implementation is divided into three parts:

1. Stage Implementation

Here system is implemented in stages. The whole system is not implemented at once. Once the user starts working with system and is familiar with it, then a stage is introduced and implemented. Also, the system is usually updated regularly until a final system is sealed.

2. Direct Implementation

The proposed new system is implemented directly and the user starts working on the new System. The shortcoming, if any, faced are then rectified later.

3. Parallel Implementation

Was implemented on approach of prototype model whose functionality was in- creased day by day, as the client was given full liberty in choosing his needs and gets to the maximum benefit out of the system developed. Implementation is that process plan where the theoretical design is put into real test. All the theoretical and practical works are now implemented as a working system. This is most crucial stage in the life cycle of a project. The project may be accepted or rejected depending on how it gathers confidence among the users. The implementation stage involves the following tasks.

5.1.1 Implementation methods

Implementation of software refers to final installation of package in the real environment, to the satisfaction of the intended users and the successful operation of the system. Implementation is the stage of the project where the theoretical design is turned into a working system. Implementation includes all those activities that takes place to convert from the old system to new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements.

5.1.2 Implementation plans

The Implementation Plan describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort, and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase the final version is provided in the Integration and Test Phase and is used for guidance during the implementation phase.

5.1.3 Testing Methods

Testing is the process of examining the software to compare the actual behavior with that of the excepted behavior. The major goal of software testing is to demonstrate that faults are

not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present. System testing is the first Stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operations commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

Software testing is an integral part of to ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing. There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time- consuming process.

Software reliability is defined as the probability that the software will not undergo failure for a specified time under specified condition. Failure is the inability of a system or a component to perform a required function according to its specification. Different levels of testing were employed for software to make an error free, fault free and reliable. Basically, in software testing four type of testing methods are adopted.

Levels of testing

- Unit Testing
- Integration Testing
- Validations
- System Testing

1. Unit testing

In this each module is tested individually before integrating it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing as here each module is tested to check whether it is producing the desired output and to see if any error occurs. Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by- step instructional

document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may execute as an integration test case that involves many software components, and thus preclude the achievement of most if not all of the goals established for unit testing. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as "module testing".

The modules of this project are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules of the project independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the project.

2.Integration testing

Integration testing (sometimes called integration and testing, abbreviated I and T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.

3. System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to

detect defects both within the "inter-assemblages" and also within the system as a whole. System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System Testing is the first level where the System is tested as a whole. The System is tested to verify whether it meets the functional and technical requirements. The application/System is tested in an environment that closely resembles the production environment where the application will be finally deployed. The perquisites for System Testing are:

- All the components should have been successfully Unit Tested.
- All the components should have been successfully integrated.

User acceptance testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system. User acceptance of a system is a key factor to success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the prospective system user at the time of developing and making changes whenever required. This is done with regard to the following points.

- Input screen design.
- Output screen design.
- Format of the report and other.

5.2 Coding Details and Code

from tkinter import *

import sys

import winsound

Importing OpenCV Library for basic image processing functions

import cv2

Numpy for array related functions

import numpy as np

Dlib for deep learning based Modules and face landmark detection

import dlib

face$_utilsforbasicoperationsofconversion$

from imutils import face$_utils$

def activate():

print("HELLO")

cap = cv2.VideoCapture(0)

Initializing the face detector and landmark detector

detector = dlib.get$_frontal_face_detector$()

predictor = dlib.shape$_predictor$("$shape_predictor_68_face_landmarks.dat$")

status marking for current state

sleep = 0

drowsy = 0

active = 0

status=""

color=(0,0,0)

def compute(ptA,ptB):

dist = np.linalg.norm(ptA - ptB)

return dist

def blinked(a,b,c,d,e,f):

up = compute(b,d) + compute(c,e)

down = compute(a,f)

ratio = up/(2.0*down)

Checking if it is blinked

if(ratio¿0.25):

return 2

elif(ratio¿0.21 and ratio¡=0.25):

return 1

else:

return 0

while True:

,$frame = cap.read()$

gray = cv2.cvtColor(frame, cv2.COLOR$_BGR2GRAY$)

faces = detector(gray)

detected face in faces array

for face in faces:

x1 = face.left()

y1 = face.top()

x2 = face.right()

y2 = face.bottom()

face$_frame$ = $frame.copy()$

cv2.rectangle(face$_frame$, $(x1, y1), (x2, y2), (0, 255, 0), 2)$

landmarks = predictor(gray, face)

landmarks = face$_tils.shape_on_p(landmarks)$

The numbers are actually the landmarks which will show eye

left$_blink$ = $blinked(landmarks[36], landmarks[37],$

landmarks[38], landmarks[41], landmarks[40], landmarks[39])

right$_blink$ = $blinked(landmarks[42], landmarks[43],$

landmarks[44], landmarks[47], landmarks[46], landmarks[45])

Now judge what to do for the eye blinks

if(left$_blink == 0 or right_blink == 0)$:

sleep+=1

drowsy=0

active=0

if(sleep¿6):

status="SLEEPING !!!"

color = (255,0,0)

winsound.PlaySound("sou.wav", winsound.SND$_FILENAME$)

elif(left$_blink == 1 or right_blink == 1)$:

sleep=0

active=0

drowsy+=1

if(drowsy¿6):

status="Drowsy !"

color = (0,0,255)

else:

drowsy=0

sleep=0

active+=1

if(active¿6):

status="Active :)"

color = (0,255,0)

cv2.putText(frame, status, (100,100), cv2.FONT$_H ERSHEY_S IMPLEX$, 1.2, *color*, 3)

for n in range(0, 68):

(x,y) = landmarks[n]

cv2.circle(face$_f rame$, $(x, y), 1, (255, 255, 255), -1$)

cv2.imshow("Frame", frame)

cv2.imshow("Result of detector", face$_f rame$)

if cv2.waitKey(1) 0xFF == ord('q'):

break

def quit():

sys.exit("DONE")

a=Tk()

a.title("DRIVER DROWSINESS DETECTION SYSTEM")

a.geometry("550x630+300+100")

a.configure(background="f4ab79")

Label(a,text="DROWSINESS DETECTION SYSTEM",font="Tw-Cen-MT-CondensedExtraBold18

Button(a,text="ACTIVATE",height=3,font="Bahnschrift 14 bold",width=20,border=0,command=a

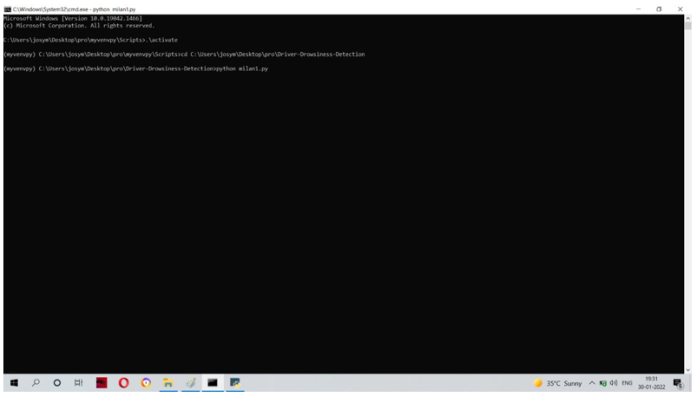
Button(a,text="QUIT",height=2,font="Bahnschrift 14 bold",command=quit,width=10,border=0,ba

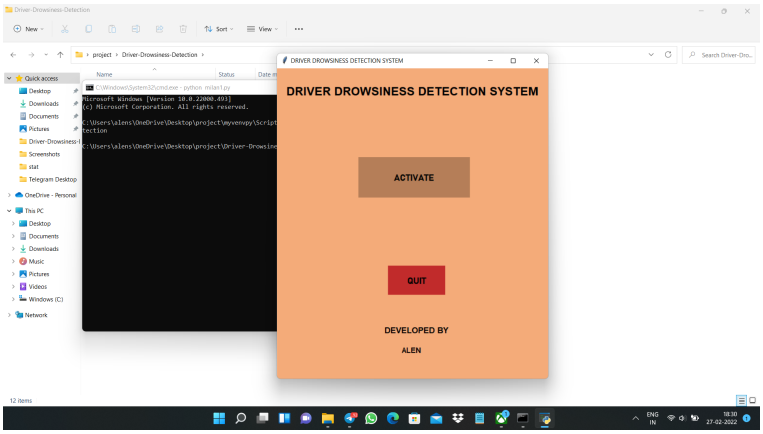
Label(a,text="BY",font="Tw-Cen-MT-Condensed-Extra-Bold 13 bold",bg="f4ab79").place(x=215,y

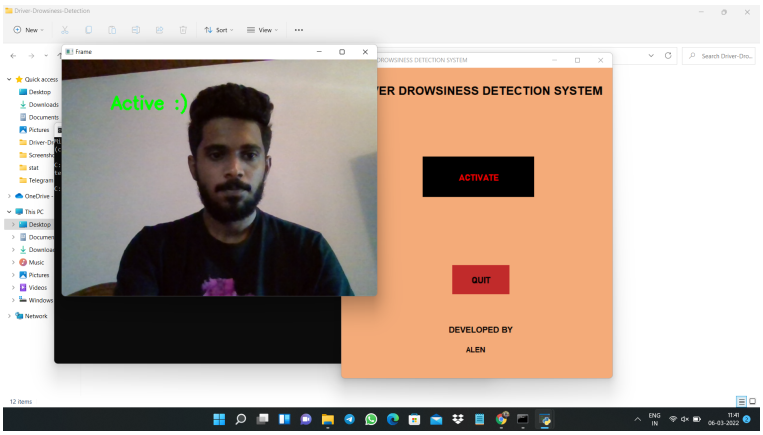
Label(a,text="ALEN ",font="Tw-Cen-MT-Condensed-Extra-Bold 11 bold",bg="f4ab79").place(x=2

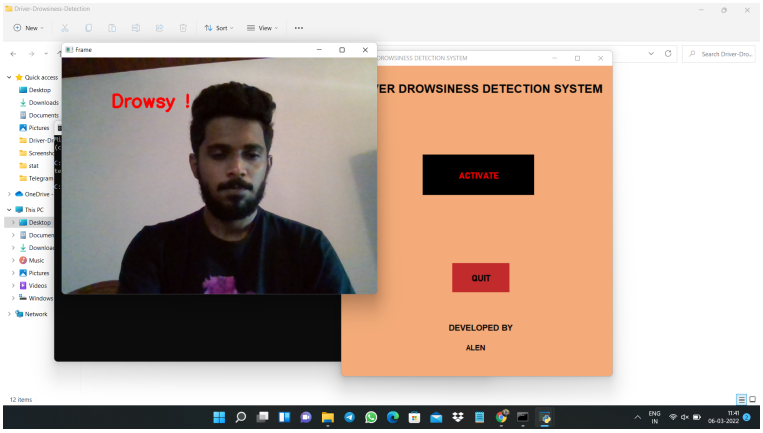
a.mainloop()

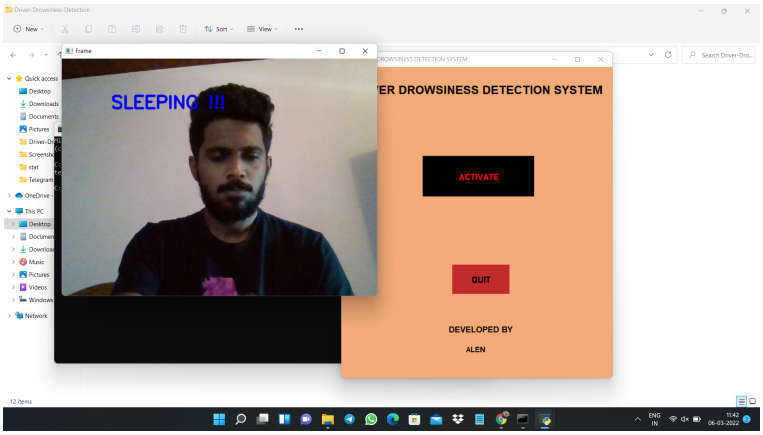
INTERFACE




DRIVER DROWSINESS DETECTION SYSTEM
ACTIVATE
QUIT
DEVELOPED BY
ALEN


Frame
Active :)
ACTIVATE
QUIT
DEVELOPED BY
ALEN


Drowsy !
ER DROWSINESS DETECTION SYSTEM
ACTIVATE
QUIT
DEVELOPED BY
ALEN


SLEEPING !!!
ER DROWSINESS DETECTION SYSTEM
ACTIVATE
QUIT
DEVELOPED BY
ALEN

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

This project proposes a drowsiness and intrusion detection system based on driver behaviour. The role of the system is to detect facial landmark from images that are collected while the person is driving the vehicle by a camera module attached to the vehicle and deliver the obtained data to the trained model to identify the driver's state. Once the collected data is detected to be showing signs of drowsiness the person will be alerted using the speakers in the vehicle so that the person can stop the vehicle to avoid any accidents due to his drowsy state. However, there is still space for the performance improvement. The further work will focus on detecting the distraction and yawning of the driver. Also, accelerometer is to be incorporated to track the speed of the car.

6.2 Future Scope

Our model is designed for detection of drowsy state of eye and give and alert signal or warning may be in the form of audio or any other means. But the response of driver after being warned may not be sufficient enough to stop causing the accident meaning that if the driver is slow in responding towards the warning signal, then accident may occur. Hence to avoid this we can design and fit a motor driven system and synchronize it with the warning signal so that the vehicle will slow down after getting the warning signal automatically.

Chapter 7

References

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

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

[3] JAIN, "FACE DETECTION IN COLOR IMAGES; R. L. HSU, M. ABDEL-MOTTALEB, AND A. K. JAIN.

[4] OPEN/CLOSED EYE ANALYSIS FOR DROWSINESS DETECTION; P.R. TABRIZI AND R. A. ZOROOFI