**Accident Prevention Master**

**DISSERTATION**

*Submitted in partial fulfillment of the*

*Requirements for the award of the degree*

*Of*

**Bachelor of Technology**

in

**Information Technology**

By:

**Sanjot Kaur (07013203117/IT-2/2017)**

**Santa Singh (07113203117/IT-2/2017)**

**Simranjeet Kaur (07613203117/IT-2/2017)**

**Snehpreet Kaur (08113203117/IT-2/2017)**

Under the guidance of:

**Ms. Jasleen Kaur Bhatia**

**Department of Computer Science & Engineering**

**Guru Tegh Bahadur Institute of Technology**

**Guru Gobind Singh Indraprastha University**

**Dwarka, New Delhi**

**Year 2017-2021**

**DECLARATION**

We hereby declare that all the work presented in the dissertation entitled “Accident Prevention Master” in the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in **Information Technology,** Guru Tegh Bahadur Institute of Technology, Guru Gobind Singh Indraprastha University, New Delhi is an authentic record of our own work carried out under the guidance of **Ms. Jasleen Kaur Bhatia.**

Date: 05/06/2021

Sanjot Kaur (07013203117/IT-2/2017)

Santa Singh (07113203117/IT-2/2017)

Simranjeet Kaur (07613203117/IT-2/2017)

Snehpreet Kaur (08113203117/IT-2/2017)

# **CERTIFICATE**

This is to certify that dissertation entitled **“Accident Prevention Master”,** which is submitted by **Sanjot Kaur (07013203117/IT-2/2017), Santa Singh (07113203117/IT-2/2017), Simranjeet Kaur (07613203117/IT-2/2017) and Snehpreet Kaur (08113203117/IT-2/2017)** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in **Information Technology**, Guru Tegh Bahadur Institute of Technology, New Delhi is an authentic record of the candidate’s own work carried out by them under our guidance. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Ms Jasleen Kaur Bhatia Mr. P.S. Bedi

(Project Guide ) Head of IT Department

**Date: 05/06/2021**

**ACKNOWLEDGEMENT**

We hereby take this opportunity to thank all those people whose knowledge and experience helped us to bring this report in its present form. It would have been a tough task for us to complete the report without their help.

We express our sincere thanks and gratitude to our supervisor **Ms. Jasleen Kaur Bhatia** for providing us with support and suggestions. Without her help we could not have presented this work upto the present standard.

Finally We would like to express our deep appreciation to our families and friends who have been a constant source of inspiration. We are eternally grateful to them for always encouraging us wherever and whenever we needed them.

**Date: 05/06/2021**

Sanjot Kaur

sanjot307@gmail.com

07013203117

Santa Singh

scionsn99@gmail.com

07113203117

Simranjeet Kaur

simranjk.rana@gmail.com

07613203117

Snehpreet Kaur

snehpreet1998@gmail.com

08113203117

**ABSTRACT**

Driving while drowsy is a serious and often tragic problem in the United States. A report from the National Highway Traffic Safety Administration estimates that drowsy driving leads to 100,000 police-reported crashes each year, 71,000 injuries and 800 fatalities, and a 125 crore cost. The National Sleep Foundation reported that 37 percent of people admitted to falling asleep behind the wheel. Young adults between the ages of 18–29 are even more likely to drive when drowsy, with a reported 71 percent compared to approximately half of those in the age range 30 to 64.

A solution to this problem is to identify when a driver is falling asleep, and alert the driver and passengers of the situation so that appropriate measures can be taken.

In this work we propose two image-based frameworks for the detection and recognition of drowsiness based on a video feed of the driver’s face. Both frameworks are broadly divided into two parts, namely, feature extraction followed by time series based prediction. The two frameworks differ in the way features are extracted and fed into a CNN Network to make predictions. The model aims to monitor a driver’s condition in real-time. From the video, an image-based noninvasive technique is used to detect the facial features of the driver with time and classify drowsiness (Alert or Drowsy) and sound an alarm if drowsy.

The project Accident Prevention includes two features: yawn detector and drowsiness detector.

**LIST OF FIGURES AND TABLES**

|  |  |  |
| --- | --- | --- |
| **Fig No** | **Figure Name** | **Page no.** |
| 1 | ER Diagram | 17 |
| 2 | Flow Chart | 18 |
| 3 | Use Case diagram | 19 |
| 4 | Class Diagram | 20 |
| 5 | Design Sructure | 28 |
| 6 | Facial Landmarks | 33 |
| 7 | Model Architecture | 34 |
| 8 | Test Cases | 37 |

**CONTENTS**

**CHAPTER PAGE NO.**



**Title Page**  ii

**Certificate** iii

Declaration iv

Acknowledgement v

Abstract vi

List of Figure and Table vii

1. Introduction 1
   1. Need 1
   2. About Project 2
   3. Objective of Project 3
   4. Purpose of Project 4
   5. Scope of Project 4
2. Requirement Analysis 5
   1. Libraries 5
   2. Hardware Requirements 5
   3. Software Requirements 5
   4. Functional Requirements 6
   5. Non – Functional Requirements 7
3. System Design 8

3.1. Designing Process 8

3.2 Entity Relationship Diagram 11

3.3 Flow Chart Diagram 12

3.3 Use Case Diagram 13

3.4 Class Flow Diagram 14

3.5 Design Methodologies 18

3.6 Structured Design 21

1. Technologies Used 22
   1. Python
   2. Command Prompt
   3. Image Processing
   4. Machine Learning
   5. Dataset
   6. Test Cases
2. RESULT 29
3. SUMMARY AND CONCLUSIONS 29
4. REFERENCES 36
5. APPENDIX A (SCREENSHOTS) 39
6. APPENDIX B (CODE) 45

**1.INTRODUCTION**

**1.1 Need:-**

Accident Prevention is very useful software for preventing Road Accidents. It is divided into two parts:

* Yawn Detector
* Drowsiness Detector

The increasing number of traffic accidents due to a diminished vigilance level of drivers has become a serious problem for society. Accidents related to driver hypervigilance are more serious than other types of accidents, since sleepy drivers often do not take evasive action prior to a collision. Monitoring the driver’s level of vi- gilance and alerting the driver when he is not paying adequate attention to the driving has become a matter of concern in order to prevent accidents. The prevention of such accidents is a major focus of effort in the field of active safety research.

Many researchers are working on the development of monitoring systems using specific techniques. The best detection techniques are based on physiological phenomena like brain waves, heart rate, pulse rate and respiration. These techniques are intrusive, causing annoyance. A driver’s state of vigilance can also be characterized by indirect behaviors of the vehicle like lateral position, steering wheel movements.

Although these techniques are not intrusive, they are subjected to several limitations such as the vehicle type, driver experience, geometric characteristics and state of the road. People in fatigue show some visual behavior easily observable from changes in their facial features like eyes, head and face.

**1.2 About Project**

The drowsiness of drivers is one of the main reasons behind road accidents. It is natural for drivers who frequent long routes to doze off when behind the steering wheel. Even stress and lack of sleep can cause drivers to feel drowsy while driving. This project aims to prevent and reduce such accidents by creating drowsiness detection and yawn detection.

Yawn Detection is all about detecting yawning( open one’s mouth wide and inhale deeply due to tiredness or boredom) using OpenCV and Dlib. It can be used in various major applications like Self Driving Cars, Driver’s Fatigue detection, Driver’s Drowsiness detection, Driver’s consciousness detection etc.

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.

**1.3 Objectives of the Project**

With this intermediate-level Python project, we will be making a drowsiness detecting

device. A countless number of people drive on the highway day and night. Taxi drivers,

Due to which it becomes very dangerous to drive when feeling sleepy.

Driving while drowsy is a serious and often tragic problem in the United States. A report from the National Highway Traffic Safety Administration estimates that drowsy driving leads to 100,000 police-reported crashes each year, 71,000 injuries and 800 fatalities, and a 125 crore cost. The National Sleep Foundation reported that 37 percent of people admitted to falling asleep behind the wheel. Young adults between the ages of 18–29 are even more likely to drive when drowsy, with a reported 71 percent compared to approximately half of those in the age range 30 to 64.

The majority of accidents happen due to the drowsiness of the driver. So, to prevent these accidents we will build a system using Python, OpenCV, and Keras which will alert the driver when he feels sleepy

1.4 Purpose of the project

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.

**1.5 Scope of the project**

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 similar functionality but with better results and additional benefits. Also, it alerts the user on reaching a certain saturation point of the drowsiness measure. 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 dependence 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.

**2. REQUIREMENT ANALYSIS AND SPECIFICATIONS**

**2.1. LIBRARIES**

* Numpy
* Scipy
* Playsound
* Dlib
* Imutils
* Opencv

**2.2. HARDWARE REQUIREMENTS**

* Webcam
* A computer system with at least i3 or ryzen 3 processor
* Minimum 2GB of RAM.
* 122 Keys Keyboard
* Color Monitor

**2.3. SOFTWARE REQUIREMENTS**

* Operating System : WINDOWS 7 or above
* Python 3.x or above
* Command Prompt

**2.4. REQUIREMENTS ANALYSIS**

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

Libraries:

* Numpy: Prerequisite 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.
  1. OS: Program is tested on Windows 10 build 1903 and PopOS 19.04
  2. Laptop: Used to run our code.
  3. Webcam: Used to get the video feed.

Yawn Detection is all about detecting yawn( open one’s mouth wide and inhale deeply due to tiredness or boredom) using OpenCV and Dlib. It can be used in various major applications like Self Driving Cars, Driver’s Fatigue detection, Driver’s Drowsiness detection, Driver’s consciousness detection etc. 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 for such an application and felt that there is a decent extent of progress in this field too.

**2.5 NON FUNCTIONAL REQUIREMENTS**

A Non – Functional Requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. functional requirements define what a system is supposed to *do* and non-functional requirements define how a system is supposed to be.Non-functional requirements are often called "quality attributes" of a system.Some of the non-functional requirements are:

### Reliability: When a user wants to call the system over a given period of time, the system should correctly deliver services as expected by the user. The reliability of the system shall be good if it delivers services as specified. Otherwise, reliability is bad and it shall produce unexpected output. So, the program should be changed according to the right situation.

**Availability:** When the system has any request at any given time, the system should be available, it should be up and running and able to deliver useful service at this time. The availability of the system shall be good if it delivers services when it is requested. Otherwise, if requests are not responded at any given time then it implies bad availability.

**Security:** The system should resist accidental or deliberate intrusions, when users operate on the system. If the system should not resist accidental or deliberate intrusions, then important data – such as credit card number, id number, username, etc. – which belongs to the user, shall be stolen by a hacker. Thus, security of the system shall be low and trust of users shall be ruined. So, security of the system is very important for users.

**Maintainability:** When the system is used, new requirements may emerge. When these requirements are emerged, the system should be changeable to accommodate these requirements for maintaining the usefulness of the system. If the system is not maintainable, then the system can not be modified for new requirements. In this situation, a new system should be developed to provide new requirements. The maintainability is important for avoiding cost.

**Performance**: The system should use the minimum part of memory. The processes of the system should use the processor most efficiently. Users should finish operation in the least time interval.

**Reparability**: When the system is used, system failures are inevitable. The disruption caused by failure can be minimized if the system can be repaired quickly. So, the system should be possible to find the problem, access the component that has failed and make changes to fix that component. When the system has a repair, the source code should be modified.

**3. SYSTEM DESIGN**

**3.1. Designing process**

Before we start to think about an Accident Prevention System we should make a design of the Accident Prevention System.For design we will make use of ER(Entity Relationship) diagram.

Before using the concept of ER diagram we must know what an ER diagram is .

An Entity relationship diagram shows the relationship of entity sets stored in a database. An entity in this context is a component of data. In other words ER diagrams illustrate the logical structure of databases.

At first glance an entity relationship looks very much like a flowchart. It is the specialized symbols , and the meanings of those symbols ,that make it unique.

The systems objectives outlined during the feasibility study serve as the basis from which the work of system design is initiated. Much of the activities involved at this stage is of technical nature requiring a certain degree of experience in designing systems, sound knowledge of computer related technology and thorough understanding of computers available in the market and the various facilities provided by the vendors. Nevertheless, a system cannot be designed in isolation without the active involvement of the user. The user has a vital role to play at this stage too. As we know that data collected during feasibility study will be utilized systematically during the system design. It should, however, be kept in mind that detailed study of the existing system is not necessarily over with the completion of the feasibility study. Depending on the plan of feasibility study, the level of detailed study will vary and the system design stage will also vary in the amount of investigation that still needs to be done. This investigation is generally an urgent activity during the system design as the designer needs to study minute’s details in all aspects of the system. Sometimes, but rarely, this investigation may form a separate stage between Feasibility Study and Computer System Design. Designing a new system is a creative process, which calls for logical as well as lateral thinking. The logical approach involves systematic moves towards the end product keeping in mind the capabilities of the personnel and the equipment at each decision making step. Lateral thought implies encompassing ideas beyond the usual functions and equipment. This is to ensure that no efforts are being made to fit previous solutions into new situations.

**An Overview to system analysis**

The system analysis phase is considered to be one of the most important phases in the system development life cycle. It is immensely important that the software developer make thorough study of the existing system. Thorough study of the system is made and needs i.e. features that are critical to system success and users wants (i.e. features that would be good but not essential) are brought out. The study will enable the developer to know the intricacies of the existing system.

Requirement analysis is done in order to understand the problem which the S/W system is to solve e.g., the problem could be automating the existing manual system or developing a completely new automated system or a combination of the two. For large systems having a large number of features and the need to perform many different tasks, understanding the requirement of the system is a major task. The emphasis in requirement analysis is on identifying what is needed from the system, and not how the system achieves its goal.

The main objective behind any business organization is to maximize its profit besides maintaining quality and strategic norms. This can be achieved by improving the efficiency of the system by providing more facilities using automation, by adopting faster data access, proper communication. , whereas the main objective behind automation is not only to maximize profit but also to take care of passenger’s interest by providing them better facilities.

The most important objective behind automation is to minimize Paper Work. Paper Work/Registers are replaced by a Centralized Data Bank, which is well equipped to store / provide information as and when required. Data Bank also helps speed up the communication between various depts. outside agencies, as there is no need of making requests against different departments for specific data and to wait for it for a long period. This also improves efficiency as it saves time and human resources.

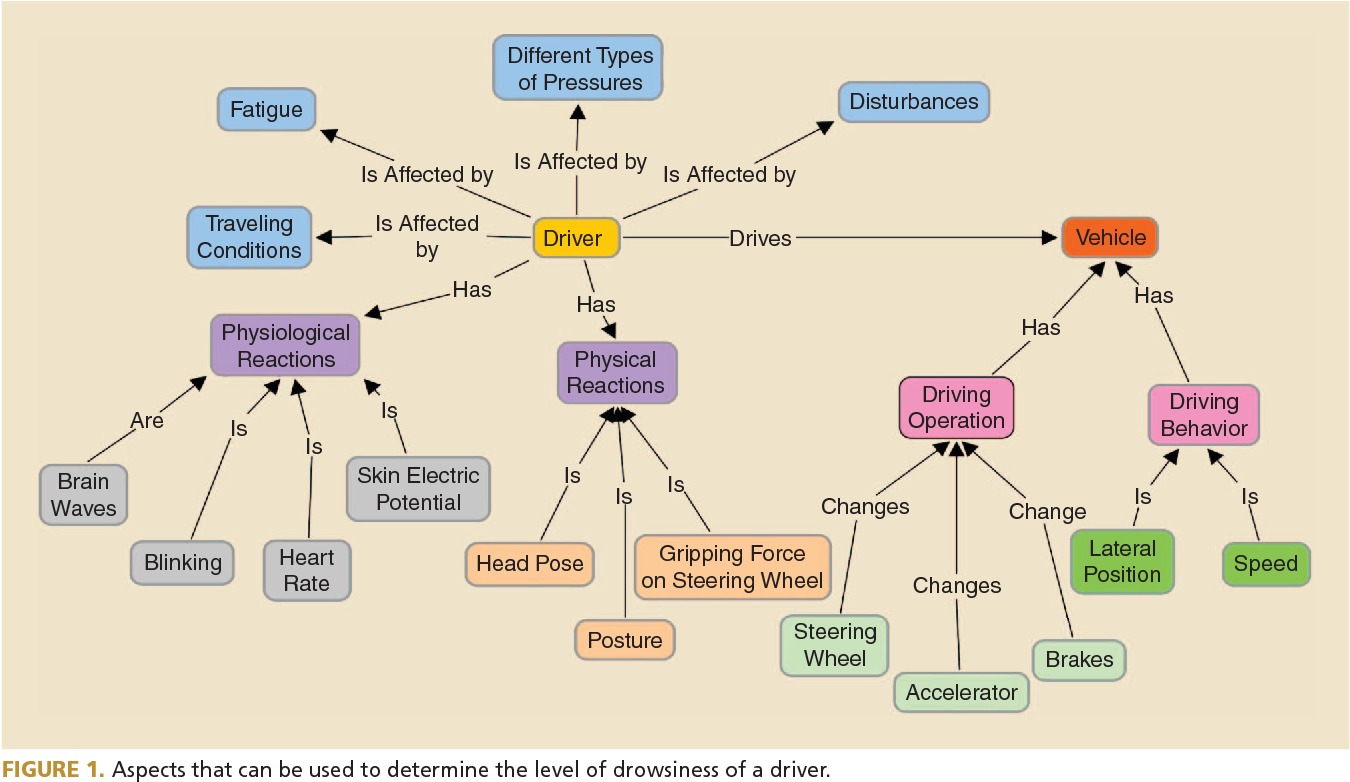
By making the manual system computerized, we can ensure complete utilization of our existing resources. Automation helps in generating the reports / information in a consistent way, which saves time and labor if done manually.

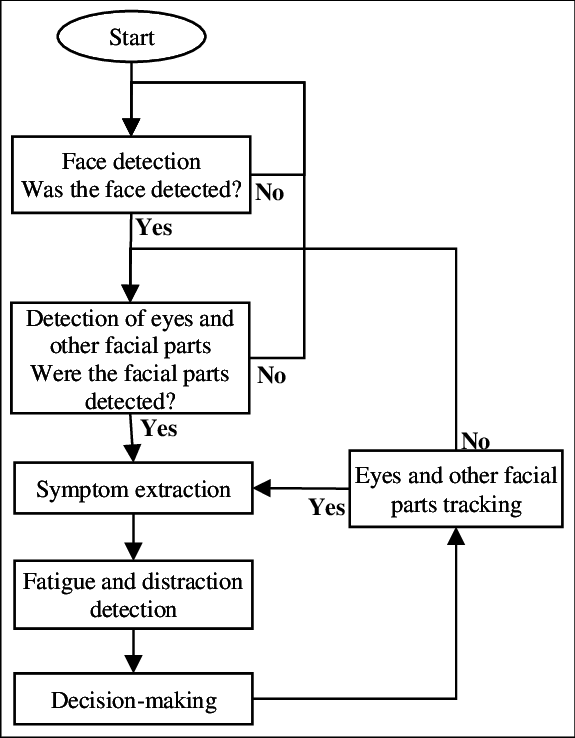
In this project we have used the Rapid Application Development (RAD) model. RAD is an incremental software development process model that emphasizes an extremely short development cycle. The following phases are encompassed:

* **Business modeling:** All the information about the business functioning of the Airways department is collected, how the data and information is flow from one end to another end using the following questions: What information drives the department process? What information is generated? Who generates it? Where does the information go? Who process it?
* **Data modeling:** The information collected in Business modeling phase is refined into a set of data objects that are needed to support the project. The attributes of each object are identified and the relationships between these objects defined.
* **Process modeling:** Processing descriptions and functions like adding, modifying, deleting records, printing reports, providing information, file handling etc. are created.
* **Application generation:** The fourth generation techniques are used to generate application, like reusing the predefined functions or creating reusable components.
* **Testing:** Most of the functions are already tested, as they are predefined functions.

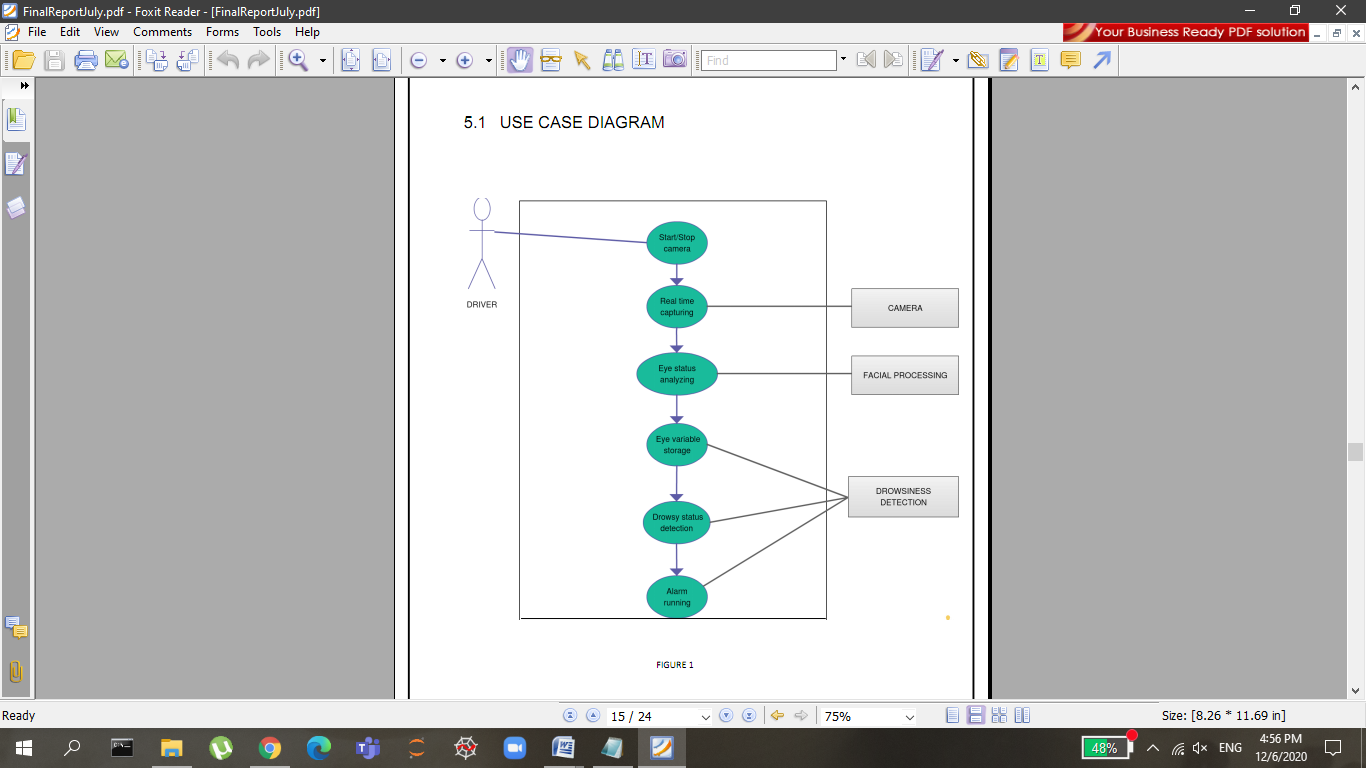
**3.2. Entity Relationship diagram:-**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. If the application is primarily a database application, the entity-relationship approach can be used effectively for modeling some parts of the problem. The main focus in ER modeling is the Data Items in the system and the relationship between them. It aims to create a conceptual scheme for the Data from the user’s perspective. The model thus created is independent of any database model. The ER models are frequently represented as ER diagram. Here we present the ER diagram of the above mentioned project.

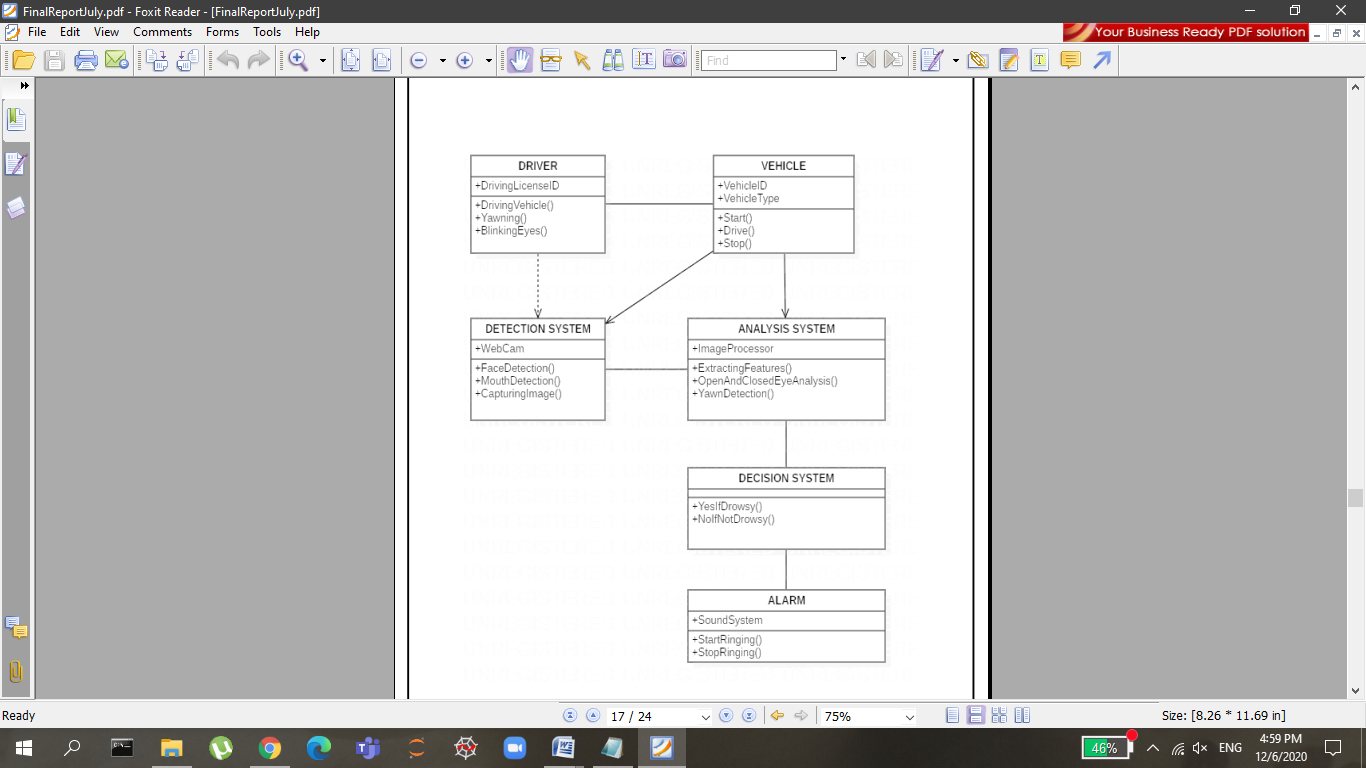
****

**3.3 ACCIDENT PREVENTION FLOW CHART**

**3.4 USE CASE DIAGRAM**

****

**3.5 Class Diagram**

****

**3.4 DATA FLOW DIAGRAM**

Data flow diagrams are commonly used during problem analysis. Data flow diagrams are quite general and not limited to problem analysis for software requirement specification. A DFD shows the flow of data through a system. It views a system as a function that transforms the inputs into desired outputs. Any complex system does not perform this transformation into a single step and a data will typically undergo a series of transformation before it becomes an output. The DFD aims to capture the transformations that take place within a system to the input data so that eventually the output data is produced.

The agent that performs the transformation of data from one state to another is called a process. So, a DFD shows the movement of data through the different transformations or processes in the system. Named circles show the processes and data named arrows entering or leaving the bubbles represent flows.

The rectangle represents a source and sink and is a net originator or consumer of data. A source or sink is typically outside the main system of study.All external files are shown as a labeled straight line.

The need of multiple data flows by a process is represented by a “\*” between the data flows.the symbol represents the AND relationship.for example, if there is a “\*” between the two input data flows A and B for a process,it means that A AND B are needed for the process.

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer.

When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user or limitations of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions thereof since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system-designing phase:

**The primary objective of the design:** Of course, is to deliver the requirements as specified in the feasibility report. In general, the following design objectives should be kept in mind:

* **Practicality**: The system must be stable and can be operated by people with average
* **Efficiency**: This involves accuracy, timeliness and comprehensiveness of the system

output.

* **Cost**: it is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.
* **Flexibility**: The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.
* **Security**: This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse.

System design involves first logical design and then physical construction of the system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, databases and procedures. The physical construction, which follows the logical design, produces actual program software, files and a working system.

The designer normally will work under following constraints:

* **Hardware**: The existing hardware will obviously affect the system design.
* **Software**: The available software (operating system, utilities, language etc.) in the market will constrain the design.
* **Budget**: The budget allocated for the project will affect the scope and depth of design.
* **Time-scale**: The new system may be required by a particular time (e.g. the start of a financial year). This may put a constraint on the designer to find the best design.
* **Interface with other systems**: The new system may require some data from another computerized system or may provide data to another system in which case the files must be compatible in format and the system must operate with a certain processing cycle.

**Processing Techniques:**

The processing options available to the designer are:

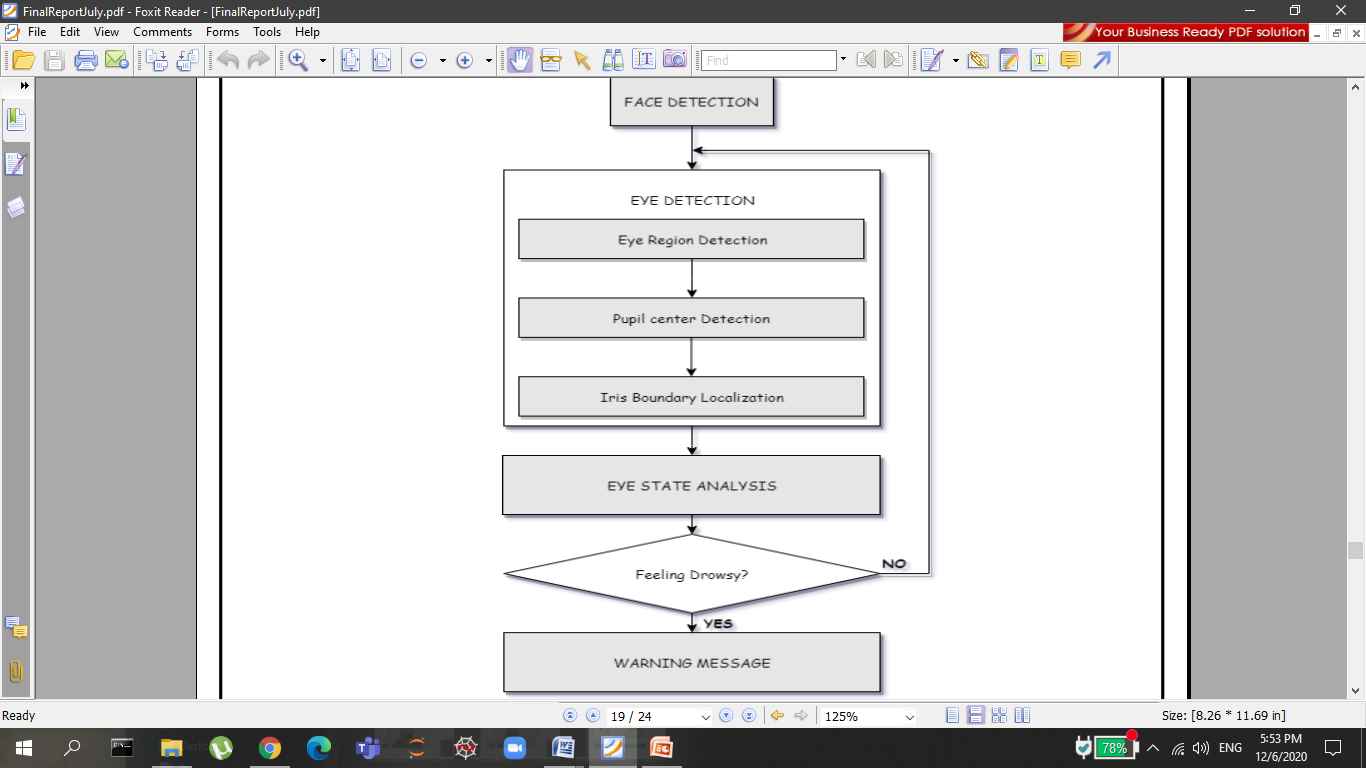
* Batch processing
* Real-time processing
* On-line processing
* A combination of all the above

You are already aware of these techniques. It is quite interesting to note, however, that a combination of these is often found to be ideal in traditional data processing applications. This increases throughput of the system as also brings down the response time of on-line activities. In most of die business applications, 24-hour data is acceptable enough and hence it is possible to update voluminous data after office-hours in batch mode.

**3.5 DESIGN METHODOLOGIES**

The scope of the systems design is guided by the framework for the new system developed during analysis. More clearly defined logical method for developing system that meets user requirements has led to new techniques and methodologies that fundamentally attempt to do the following:

* Improve productivity of analysts and programmers
* Improve documentation and subsequent maintenance and enhancements.
* Cut down drastically on cost overruns and delays
* Improve communication among the user, analyst, designer, and programmer.
* Standardize the approach to analysis and design
* Simplify design by segmentation.



**3.6 STRUCTURED DESIGN**

Structured design is a data flow based methodology. The approach begins with a system specification that identifies inputs and outputs and describes the functional aspects of the system. The specifications then are used as a basis for the graphic representation. The next step is the definition of the modules and their relationships to one another in a form called a structure chart, using a data dictionary and other structured tools.

Logical design proceeds from the top down. General features, such as reports and inputs are identified first. Then each is studied individually and in more detail. Hence, the structured design partitions a program into small, independent modules. They are arranged in a hierarchy that approximates a model of the business area and is organized in a top-down manner. Thus, structured design is an attempt to minimize the complexity and make a problem manageable by subdividing it into smaller segments, which is called Modularization or decomposition. In this way, structuring minimizes intuitive reasoning and promotes maintainable provable systems.

**4.TECHNOLOGIES USED**

**4**  **DEVELOPMENT TOOLS AND TECHNOLOGIES**

**4.1 PYTHON 3:**

Python is a general purpose interpreted, interactive, object-oriented, and high-level

programming language. It was created by Guido Van Rossum during 1985-1990.

Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is named after a TV show called ‘‘Monty Python’s Flying Circus’’.

Python 3.0(also called “Python 3000” or “Py3K”) was released in December 3,

2008. The latest version of python accumulated new and redundant ways to program

the same task, Python 3.6 had an emphasis on removing duplicative constructs and modules, in keeping with “There should be one and preferable only one – obvious way to do it”. Python’s dynamic typing encourage the programmer to write a code that is clear, well structured as well as easy to understand.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

The features of dynamic

typing are:

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software  Types are bound to values but not to variables.
* Function and method lookup is done at runtime.
* Values are inspect-able.
* There is an interactive interpreter, more than one, in fact.
* You can list the methods supported by any given object.

Because code is automatically compiled to byte code and executed, Python is

suitable for use as a scripting language, Web application implementation language

etc. Because of its strong structuring constructs (nested code blocks, functions,

classes, modules and packages) and its consistent use of objects and OOP, Python

enables you to write clear and logical code for small and large projects.

Python has varied advantageous features, and programmers prefer this language to other programming languages because it is easy to learn and code too. However, this language has still not made its place in some computing arenas that includes Enterprise Development Shops

## 4.1.1 Software System Attributes

### Reliability: When user wants to call the system over a given period of time, the system should correctly deliver services as expected by the user. The reliability of the system shall be good if it delivers services as specified. Otherwise, reliability is bad and it shall produce unexpected output. So, program should be changed according to right situation.

**Availability:** When the system has any request at any given time, system should be available, it should be up and running and able to deliver useful service at this time. The availability of the system shall be good if it delivers services when it is requested. Otherwise, if requests are not responded at any given time then it implies bad availability.

**Security:** The system should resist accidental or deliberate intrusions, when users operate on the system. If the system should not resist accidental or deliberate intrusions, then important data – such as credit card number, id number, username, etc. – which belongs to user, shall be stolen by hacker. Thus, security of the system shall be low and trust of users shall be ruined. So, security of the system is very important for users.

**Maintainability:** When the system is used, new requirements may emerge. When these requirements are emerged, the system should be changeable to accommodate these requirements for maintaining the usefulness of the system. If the system is not maintainable, then the system can not be modified for new requirements. In this situation, a new system should be developed for provide new requirements. The maintainability is important for avoiding from cost.

**Performance**: The system should use the minimum part of memory. The processes of the system should use the processor most efficiently. User should finish operation in the least time interval.

**Reparability**: When the system is used, system failures are inevitable. The disruption caused by failure can be minimized if the system can be repaired quickly. So, the system should be possible to find the problem, access the component that has failed and make changes to fix that component. When the system has a repair, the source code should be modified.

**4.2 Command Prompt**

Command Prompt is a command Line Interpreter application available in most Windows Operating systems. We use Command Prompt to execute entered commands. Most of the commands are used to automate tasks via scripts and batch files to perform advanced administrative functions and troubleshoot and solve certain kinds of Windows issues.

The command line prompt is one of the most powerful tools within the Windows operating system. It allows users to do almost anything we can do with a GUI, but simply in words. We can work with directories, files, folders, and the Internet. Using command prompt, we can make the computer do anything by just typing the commands.

**4.3 Image Processing**

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

* + - In our program we used Dlib, a pre-trained program trained on the HELEN dataset to detect human faces using the pre-defined 68 landmarks.
    - 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.

**4.4 Machine Learning**

Machine learning is the [scientific](https://en.wikipedia.org/wiki/Branches_of_science)  [study](https://en.wikipedia.org/wiki/Branches_of_science) of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use in order to perform a specific task effectively without usi explicit instructions, relying on patterns and inference instead. It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", inorder to make predictions or decisions without being explicitly told.

**4.5. Dataset**

For Drowsiness Detector: The dataset used for this model is created by us. To create the dataset, we wrote a script that captures eyes from a camera and stores in our local disk. We separated them into their respective labels ‘Open’ or ‘Closed’. The data was manually cleaned by removing the unwanted images which were not necessary for building the model. The data comprises around 7000 images of people’s eyes under different lighting conditions. After training the model on our dataset, we have attached the final weights and model architecture file “models/cnnCat2.h5”.

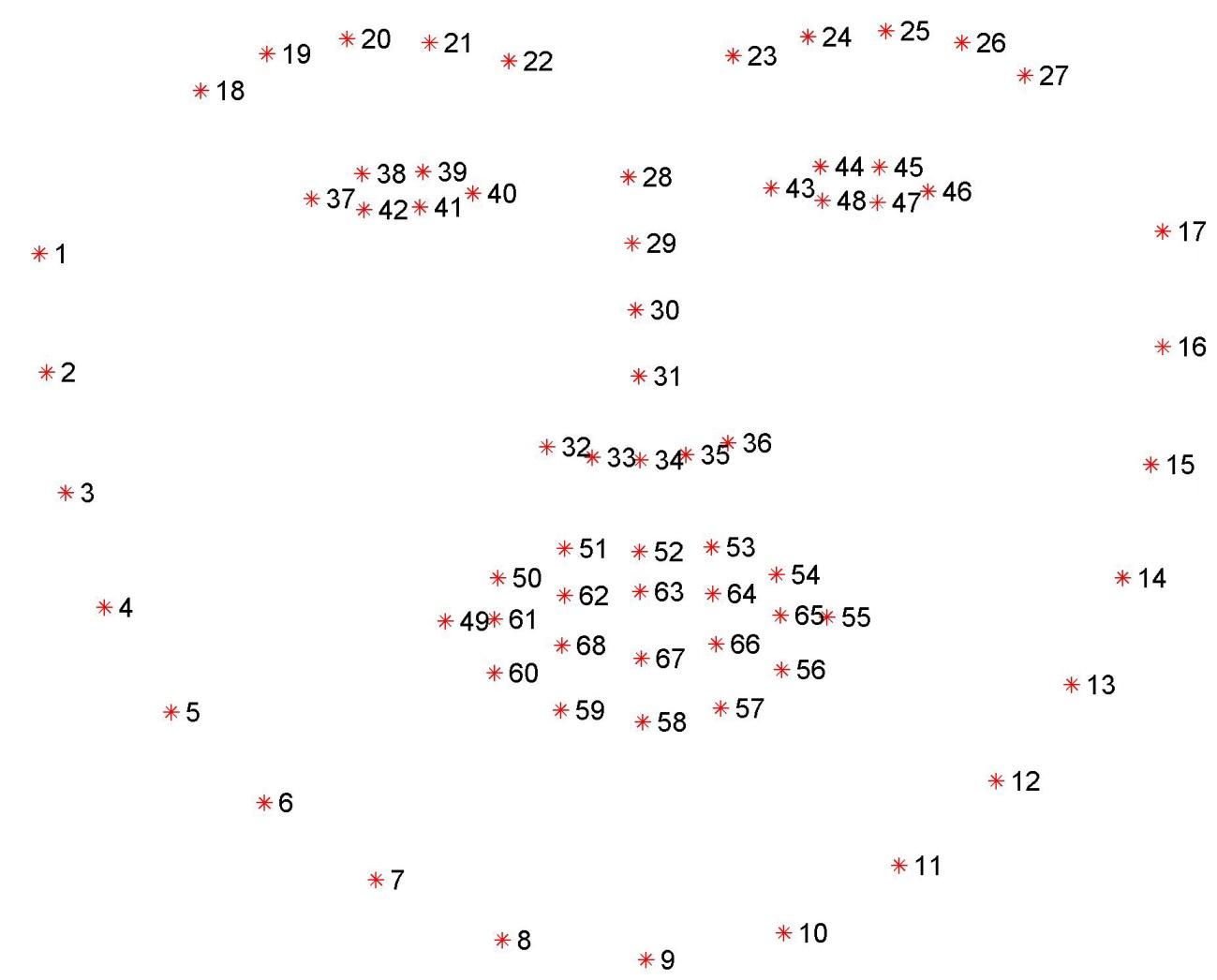
Now, you can use this model to classify if a person’s eye is open or closed.

For Yawn Detector: shape\_predictor\_68\_face\_landmarks.dat

Existing facial databases cover large variations including: different subjects, poses, illumination, occlusions etc. However, the provided annotations appear to have several limitations.

1. The majority of existing databases provide annotations for a relatively small subset of the overall images.
2. The accuracy of provided annotations in some cases is not so good (probably due to human fatigue).
3. The annotation model of each database consists of different number of landmarks.

These problems make cross-database experiments and comparisons between different methods almost infeasible. To overcome these difficulties, we propose a semi-automatic annotation methodology for annotating massive face datasets. This is the first attempt to create a tool suitable for annotating massive facial databases.



**4.6 Data Preparation**

In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into a Deep Learning model which will classify whether the person’s eyes are ‘Open’ or ‘Closed’. The approach we will be using for this Python project is as follows :

**Step 1 –** Take image as input from a camera.

**Step 2 –** Detect the face in the image and create a Region of Interest (ROI).

**Step 3 –** Detect the eyes from ROI and feed it to the classifier.

**Step 4 –** Classifier will categorize whether eyes are open or closed.

**Step 5 –** Calculate score to check whether the person is drowsy.

### 4.6.1 The Model Architecture

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.

The CNN model architecture consists of the following layers:

* Convolutional layer; 32 nodes, kernel size 3
* Convolutional layer; 32 nodes, kernel size 3
* Convolutional layer; 64 nodes, kernel size 3
* Fully connected layer; 128 nodes

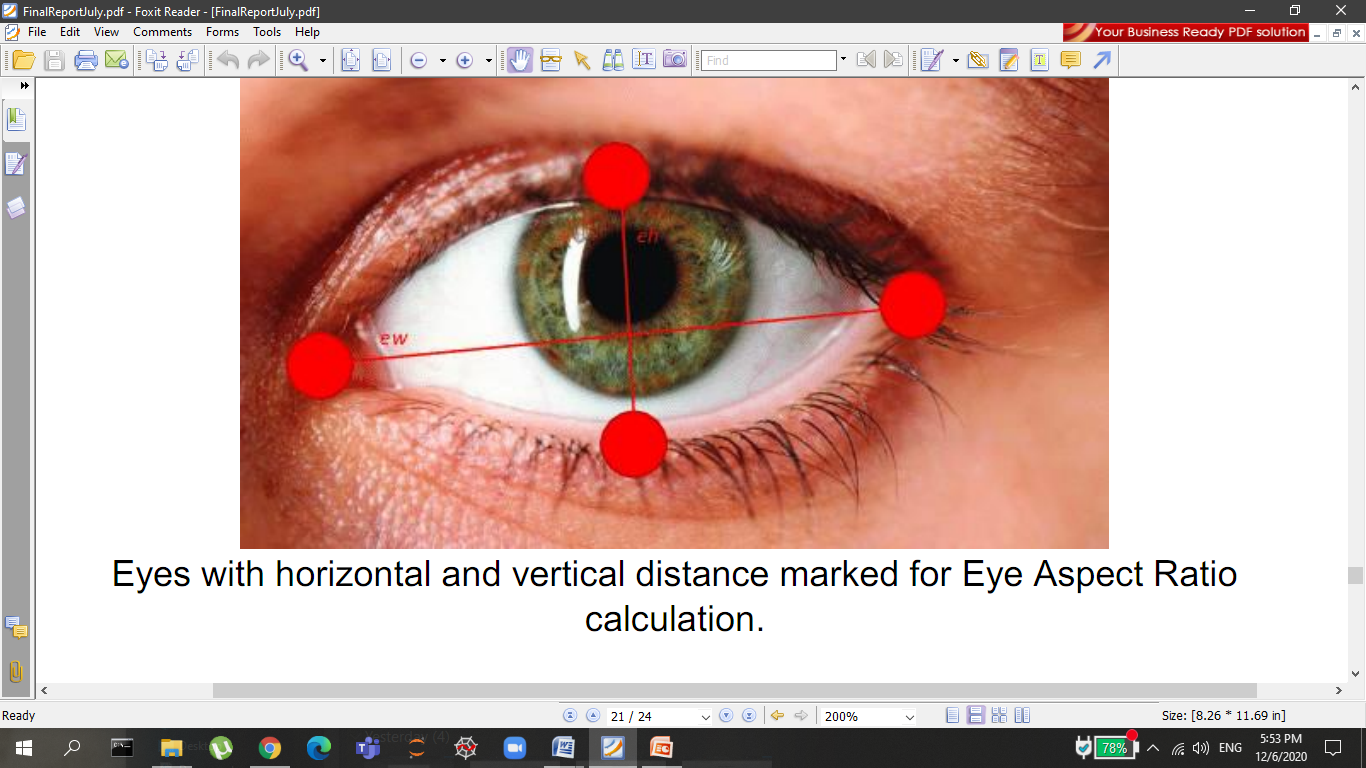
The final layer is also a fully connected layer with 2 nodes. In all the layers, a Relu activation function is used except the output layer in which we used Softmax.



### Prerequisites

The requirement for this Python project is a webcam through which we will capture images. You need to have Python (3.6 version recommended) installed on your system, then using pip, you can install the necessary packages.

1. **OpenCV –** pip install opencv-python (face and eye detection).
2. **TensorFlow –** pip install tensorflow (keras uses TensorFlow as backend).
3. **Keras –** pip install keras (to build our classification model).
4. **Pygame –** pip install pygame (to play alarm sound).
5. Let’s now understand how our algorithm works step by step.
6. **Step 1 – Take Image as Input from a Camera**
7. With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, **cv2.VideoCapture(0)** to access the camera and set the capture object (cap). **cap.read()** will read each frame and we store the image in a frame variable.
8. **Step 2 – Detect Face in the Image and Create a Region of Interest (ROI)**
9. To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don’t need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier **face = cv2.CascadeClassifier(‘ path to our haar cascade xml file’)**. Then we perform the detection using **faces = face.detectMultiScale(gray)**. It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face.
10. **Step 3 – Detect the eyes from ROI and feed it to the classifier**
11. The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in **leye** and **reye** respectively then detect the eyes using**left\_eye = leye.detectMultiScale(gray)**. Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code.
12. **l\_eye** only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into **r\_eye**.



1. **Step 4 – Classifier will Categorize whether Eyes are Open or Closed**
2. We are using [CNN](https://en.wikipedia.org/wiki/Convolutional_neural_network) classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using**r\_eye = cv2.cvtColor(r\_eye, cv2.COLOR\_BGR2GRAY)**. Then, we resize the image to 24\*24 pixels as our model was trained on 24\*24 pixel images **cv2.resize(r\_eye, (24,24))**. We normalize our data for better convergence **r\_eye = r\_eye/255**(All values will be between 0-1). Expand the dimensions to feed into our classifier. We loaded our model using **model = load\_model(‘models/cnnCat2.h5’)** . Now we predict each eye with our model  
   **lpred = model.predict\_classes(l\_eye)**. If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.
3. **Step 5 – Calculate Score to Check whether Person is Drowsy**
4. The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of the person.

A threshold is defined for example if score becomes greater than 15 that means the person’s eyes are closed for a long period of time. This is when we beep the alarm using **sound.play()**

**4.7 Test Cases**



**5. RESULT**

A very necessary window based application project for “Accident Prevention System”. The software uses a user interface which allows the user to easily manage the details of the Customer.The Software functionality is based on the core concepts of Python Programming language. The GUI of this project is done by using the essential classes of sklearn, pygame, opencv . The Software is robust and error-free. All the exceptions have been handled properly so the user may not face any difficulty. This is a project that deals with every requirement of the user.

**6. CONCLUSION**

In this Python project Accident Prevention System, we have built a drowsy driver alert and yawn detector system that we can implement in numerous ways. We used OpenCV to detect faces and eyes using a haar cascade classifier and then we used a CNN model to predict the status. 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.

**6.1 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.

**6.2 Project Risks**

* Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty.
* Many problems can infect a software project.
* A risk is a potential problem -- it might happen, it might not.
* But, regardless of the outcome, it's a really good idea to identify it, assess its probability of occurrence, and estimate its impact.
* Risk management aims at dealing with all kinds of risks that might affect a project.

## Type of risk

Risk identification is a systematic attempt to specify threats to the project plan.

step toward avoiding them when possible and controlling them when necessary. By identifying known and predictable risks the project manager takes

One method for identifying risks is to create a risk item checklist.

The checklist can be used for risks identification and focuses on some subset of known and predictable risks in the following generic subcategories:

* Product size - risks associated with the overall size of the software to be built or modified.
* Business impact - risks associated with constraints imposed by management or the marketplace.
* Customer characteristics -risks associated with the constraints of the customer and the developer’s ability to communicate with the customer in a timely manner.
* Process definition -risks associated with the degree to which the software process has been define and is following by the development organization.
* Development environment -risks associated with the availability and quality of the tools to be used to build the product.
* Technology to be built -risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
* Staff size and experience -risks associated with the overall technical and project experience of the software engineers who do the work.

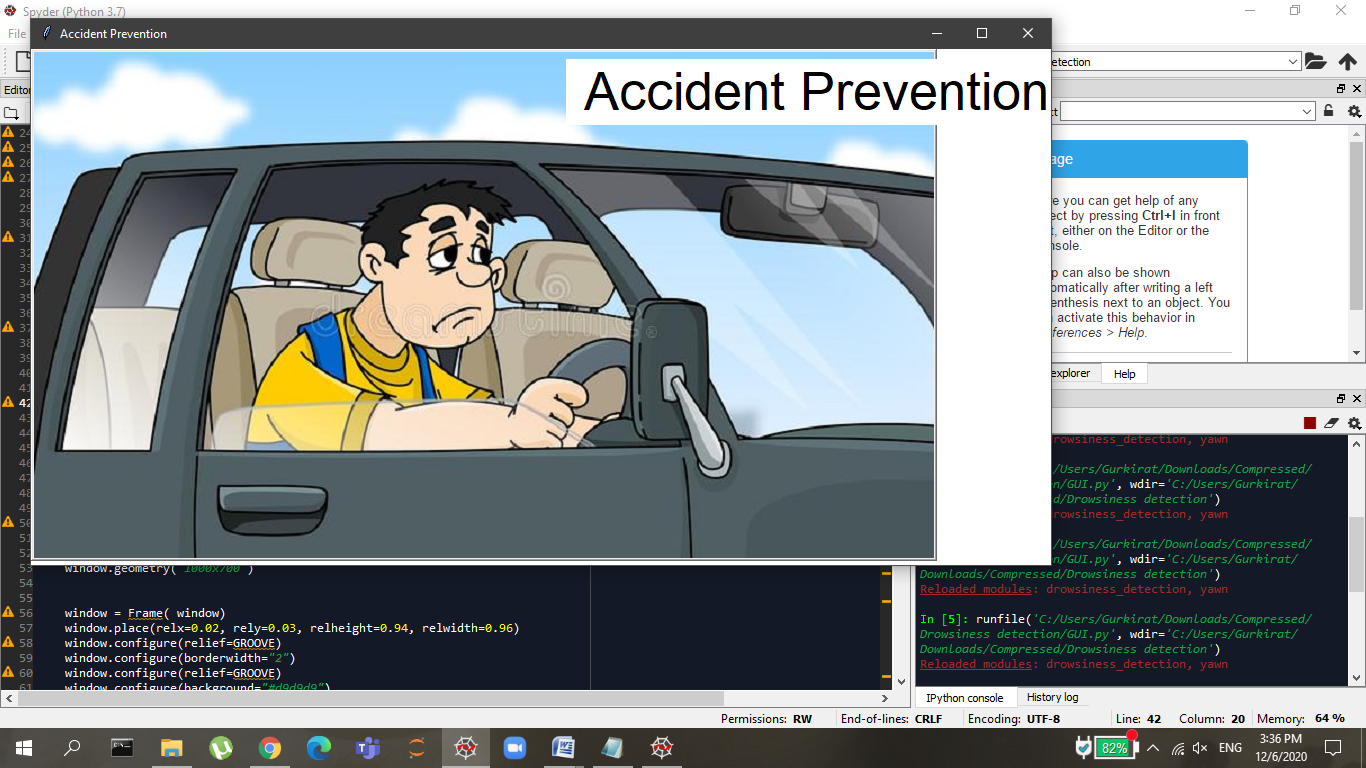
**LIMITATIONS**

* GUI is only in English, which creates a language barrier..
* Yawn Detection and Drowsiness Detection cannot be executed together

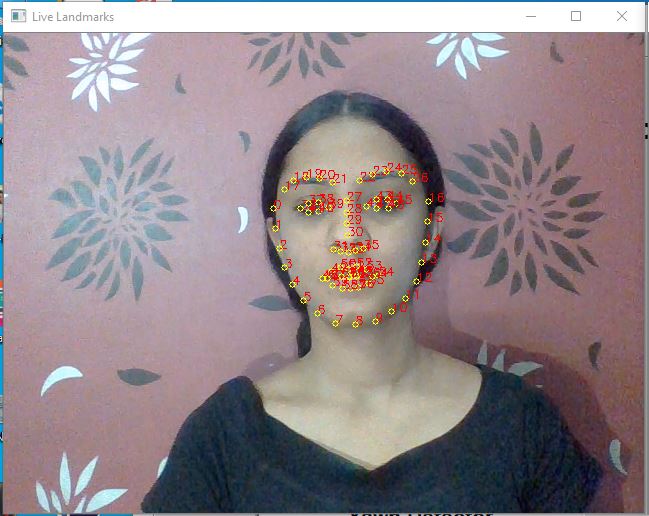
**REFERENCES**

* Books
* Doug Hellmann, The Python Standard Library by Example (2 Volume Set), Copyright 1998.
* S. R. Paul, *NHTSA's Drowsy Driver Technology Program*, 2005.
* L. Wang, X. Wu, and M. Yu, “Review of driver fatigue/drowsiness detection methods,” *Journal of Biomedical Engineering*, vol. 24, no. 1, pp. 245–248, 2007.View at:
* G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, “Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness,” *Neuroscience and Biobehavioral Reviews*, vol. 44, pp. 58–75, 2014.View at:  |
* T. Myllylä, V. Korhonen, E. Vihriälä et al., “Human heart pulse wave responses measured simultaneously at several sensor placements by two MR-compatible fibre optic methods,” *Journal of Sensors*, vol. 2012, Article ID 769613, 8 pages, 2012.View at:  |
* M. Simon, E. A. Schmidt, W. E. Kincses et al., “EEG alpha spindle measures as indicators of driver fatigue under real traffic conditions,” *Clinical Neurophysiology*, vol. 122, no. 6, pp. 1168–1178, 2011.View at:  |
* S. K. L. Lal and A. Craig, “Reproducibility of the spectral components of the electroencephalogram during driver fatigue,” *International Journal of Psychophysiology*, vol. 55, no. 2, pp. 137–143, 2005.View at:  |
* B. T. Jap, S. Lal, P. Fischer, and E. Bekiaris, “Using EEG spectral components to assess algorithms for detecting fatigue,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 2352–2359, 2009.View at:  |
* J. Krajewski, D. Sommer, U. Trutschel et al., “Steering wheel behavior based estimation of fatigue,” in *Proceedings of the 5th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pp. 118–124, 2009.View at:
* Y. Wu, J. Li, D. Yu, and H. Cheng, “Research on quantitative method about driver reliability,” *Journal of Software*, vol. 6, no. 6, pp. 1110–1116, 2011.View at:  |
* A. Eskandarian and A. Mortazavi, “Evaluation of a smart algorithm for commercial vehicle driver drowsiness detection,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 553–559, June 2007.View at:
* C. X. Huang, W. C. Zhang, C. G. Huang, and Y. J. Zhong, “Identification of driver state based on ARX in the automobile simulator,” *Technology & Economy in Areas of Communications*, vol. 10, no. 2, pp. 60–63, 2008.View at:
* Z. Zhang and J.-S. Zhang, “Driver fatigue detection based intelligent vehicle control,” in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 2, pp. 1262–1265, IEEE, Hong Kong, August 2006.View at:  |
* Q. Wang, J. Yang, M. Ren, and Y. Zheng, “Driver fatigue detection: a survey,” in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, vol. 2, pp. 8587–8591, Dalian, China, 2006.View at:  |
* Z. Zhang and J. Zhang, “A new real-time eye tracking for driver fatigue detection,” in *Proceedings of the IEEE 6th International Conference on ITS Telecommunications*, pp. 8–11, June 2006.View at:  |
* R. Parsai and P. Bajaj, “Intelligent monitoring system for driver's alertness (a vision based approach),” in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 4692 of *Lecture Notes in Computer Science*, pp. 471–477, Springer, Berlin, Germany, 2007.View at:  |
* M. Eriksson and N. P. Papanikolopoulos, “Eye-tracking for detection of driver fatigue,” in *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC '97)*, pp. 314–319, 1997.View at:
* S. M. Shi, L. S. Jin, R. B. Wang, and B. L. Tong, “Driver mouth monitoring method based on machine vision,” *Journal of JiLin University*, vol. 34, no. 2, pp. 232–236, 2004.View at:
* W. Wang, J. Xi, and H. Chen, “Modeling and recognizing driver behavior based on driving data: a survey,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 245641, 20 pages, 2014.View at:  |
* Q. Tan, M. Yang, T. Luo et al., “A novel interdigital capacitor pressure sensor based on LTCC technology,” *Journal of Sensors*, vol. 2014, Article ID 431503, 6 pages, 2014.View at:  |
* Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in *Computational Learning Theory*, vol. 904 of *Lecture Notes in Computer Science*, pp. 23–37, Springer, Berlin, Germany, 1995.View at:  |
* P. Viola and M. J. Jones, “Robust real-time object detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.View at:  |
* D. F. Dinges and R. Grace, *PERCLOS: A Valid Psychophysiological Mesure of Alertness As Assessed by Psychomotor Vigilance*, Federal Highway Administration, Office of Motor Carriers, 1998.
* W. Kong, Z. Zhou, L. Zhou, Y. Dai, G. Borghini, and F. Babiloni, “Estimation for driver fatigue with phase locking value,” *International Journal of Bioelectromagnetism*, vol. 14, no. 3, pp. 115–120, 2012.View at:
* J. P. Dukuzumuremyi, B. Zou, C. P. Mukamakuza, D. Hanyurwimfura, and E. Masabo, “Discrete cosine coefficients as images features for fire detection based on computer vision,” *Journal of Computers*, vol. 9, no. 2, pp. 295–300, 2014.View at:  |
* R. Li and C. F. Li, “Adaboost face detection based on improved covariance feature,” *Journal of Computers*, vol. 9, no. 5, pp. 1077–1082, 2014.View at:  |
* .Sagonas, E. Antonakos, G, Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 faces In-the-wild challenge: Database and results**.** Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild". 2016.
* C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. A semi-automatic methodology for facial landmark annotation. Proceedings of IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR-W), 5th Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2013). Oregon, USA, June 2013.
* C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic. 300 Faces in-the-Wild Challenge: The first facial landmark localization Challenge. Proceedings of IEEE Int’l Conf. on Computer Vision (ICCV-W), 300 Faces in-the-Wild Challenge (300-W). Sydney, Australia, December 2013.

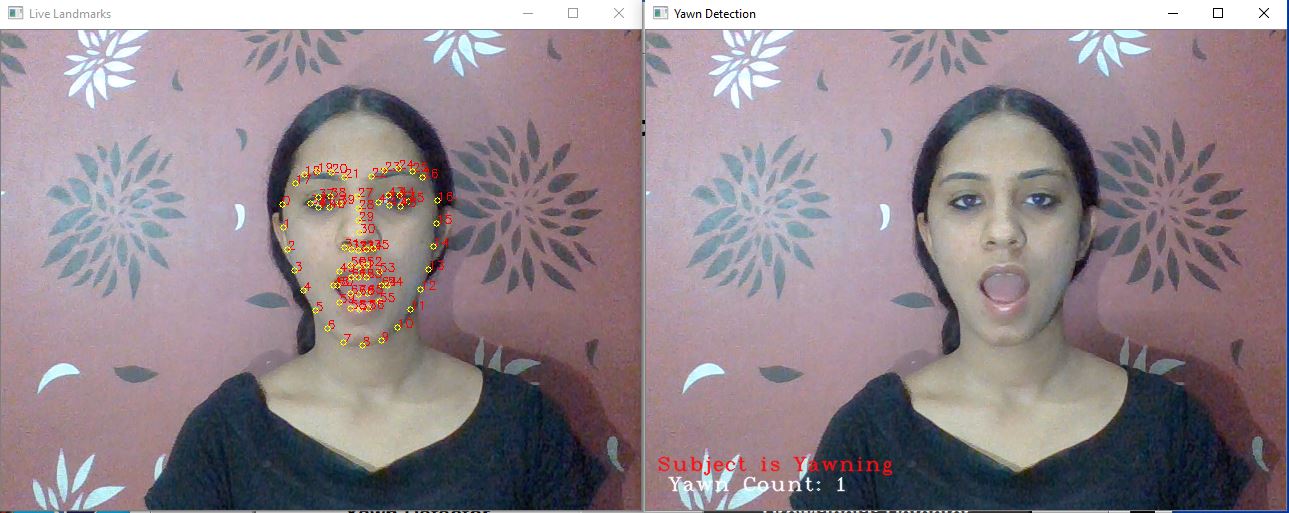
**APPENDIX A (SCREENSHOTS)**

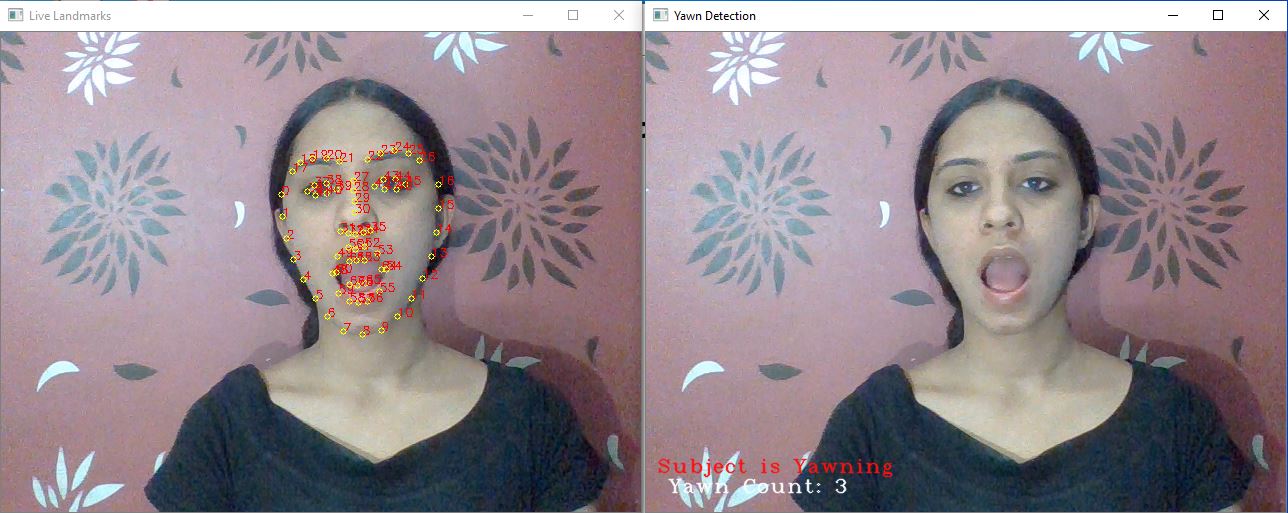
****

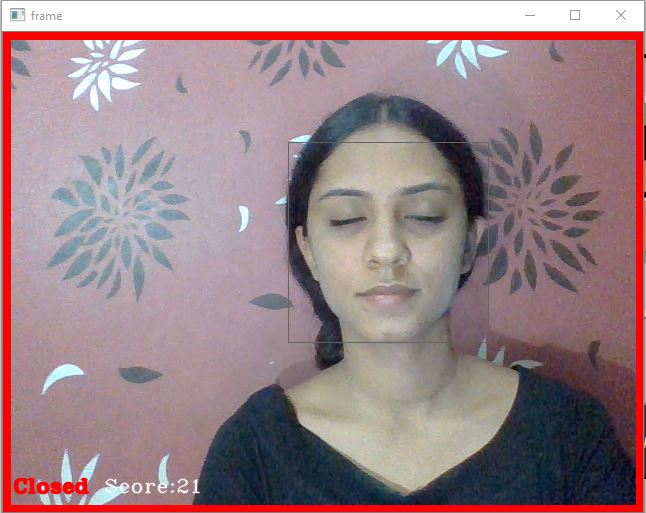
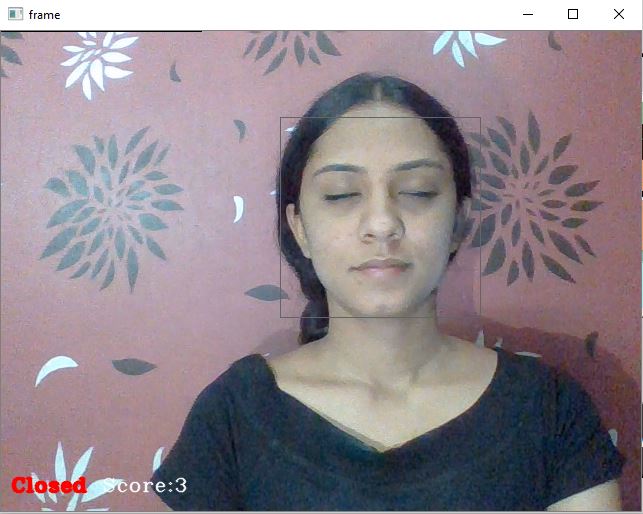
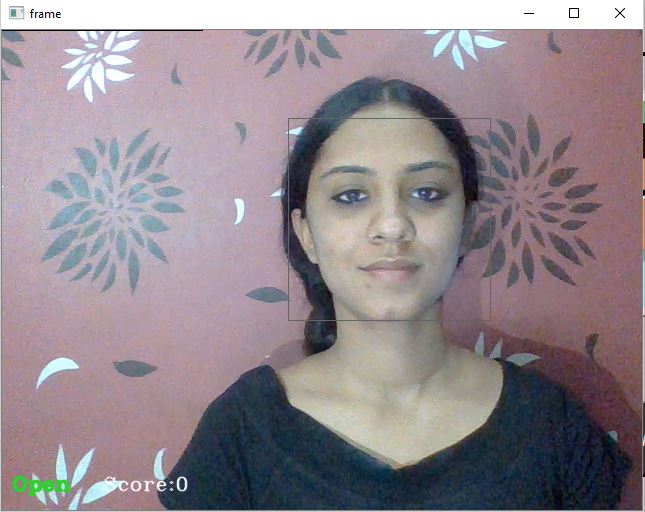
****



Facial Landmarks for Yawn Detector & Yawn Count is 1 after detecting lips wide open





**Drowsiness Detector**

**APPENDIX B (CODE)**

Drowsiness.py

import cv2

import os

from keras.models import load\_model

import numpy as np

from pygame import mixer

import time

def fun():

mixer.init()

sound = mixer.Sound('alarm.wav')

face = cv2.CascadeClassifier('haar cascade files\haarcascade\_frontalface\_alt.xml')

leye = cv2.CascadeClassifier('haar cascade files\haarcascade\_lefteye\_2splits.xml')

reye = cv2.CascadeClassifier('haar cascade files\haarcascade\_righteye\_2splits.xml')

lbl=['Close','Open']

model = load\_model('models/cnncat2.h5')

path = os.getcwd()

cap = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_COMPLEX\_SMALL

count=0

score=0

thicc=2

rpred=[99]

lpred=[99]

while(True):

ret, frame = cap.read()

height,width = frame.shape[:2]

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

faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))

left\_eye = leye.detectMultiScale(gray)

right\_eye = reye.detectMultiScale(gray)

cv2.rectangle(frame, (0,0) , (200,0) , (0,0,0) , thickness=cv2.FILLED )

for (x,y,w,h) in faces:

cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) , 1 )

for (x,y,w,h) in right\_eye:

r\_eye=frame[y:y+h,x:x+w]

count=count+1

r\_eye = cv2.cvtColor(r\_eye,cv2.COLOR\_BGR2GRAY)

r\_eye = cv2.resize(r\_eye,(24,24))

r\_eye= r\_eye/255

r\_eye= r\_eye.reshape(24,24,-1)

r\_eye = np.expand\_dims(r\_eye,axis=0)

rpred = model.predict\_classes(r\_eye)

if(rpred[0]==1):

lbl='Open'

if(rpred[0]==0):

lbl='Closed'

break

for (x,y,w,h) in left\_eye:

l\_eye=frame[y:y+h,x:x+w]

count=count+1

l\_eye = cv2.cvtColor(l\_eye,cv2.COLOR\_BGR2GRAY)

l\_eye = cv2.resize(l\_eye,(24,24))

l\_eye= l\_eye/255

l\_eye=l\_eye.reshape(24,24,-1)

l\_eye = np.expand\_dims(l\_eye,axis=0)

lpred = model.predict\_classes(l\_eye)

if(lpred[0]==1):

lbl='Open'

if(lpred[0]==0):

lbl='Closed'

break

if(rpred[0]==0 and lpred[0]==0):

score=score+1

cv2.putText(frame,"Closed",(10,height-20), font, 1,(0,0,255),2,cv2.LINE\_AA)

# if(rpred[0]==1 or lpred[0]==1):

else:

score=score-1

cv2.putText(frame,"Open",(10,height-20), font, 1,(50,205,50),2,cv2.LINE\_AA)

if(score<0):

score=0

cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE\_AA)

if(score>15):

#person is feeling sleepy so we beep the alarm

cv2.imwrite(os.path.join(path,'image.jpg'),frame)

try:

sound.play()

except: # isplaying = False

pass

if(thicc<16):

thicc= thicc+2

else:

thicc=thicc-2

if(thicc<2):

thicc=2

cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)

cv2.imshow('frame',frame)

if cv2.waitKey(1) == 13: #13 is the Enter Key

break

cap.release()

cv2.destroyAllWindows()

Yawn.py

import cv2

import dlib

import numpy as np

from pygame import mixer

import os

def fun1():

PREDICTOR\_PATH = "shape\_predictor\_68\_face\_landmarks.dat"

predictor = dlib.shape\_predictor(PREDICTOR\_PATH)

#cascade\_path='haarcascade\_frontalface\_default.xml'

#cascade = cv2.CascadeClassifier(cascade\_path)

detector = dlib.get\_frontal\_face\_detector()

mixer.init()

sound = mixer.Sound('alarm.wav')

path = os.getcwd()

thicc=2

def get\_landmarks(im):

rects = detector(im, 1)

if len(rects) > 1:

return "error"

if len(rects) == 0:

return "error"

return np.matrix([[p.x, p.y] for p in predictor(im, rects[0]).parts()])

def annotate\_landmarks(im, landmarks):

im = im.copy()

for idx, point in enumerate(landmarks):

pos = (point[0, 0], point[0, 1])

cv2.putText(im, str(idx), pos,

fontFace=cv2.FONT\_HERSHEY\_SCRIPT\_SIMPLEX,

fontScale=0.4,

color=(0, 0, 255))

cv2.circle(im, pos, 3, color=(0, 255, 255))

return im

def top\_lip(landmarks):

top\_lip\_pts = []

for i in range(50,53):

top\_lip\_pts.append(landmarks[i])

for i in range(61,64):

top\_lip\_pts.append(landmarks[i])

top\_lip\_all\_pts = np.squeeze(np.asarray(top\_lip\_pts))

top\_lip\_mean = np.mean(top\_lip\_pts, axis=0)

return int(top\_lip\_mean[:,1])

def bottom\_lip(landmarks):

bottom\_lip\_pts = []

for i in range(65,68):

bottom\_lip\_pts.append(landmarks[i])

for i in range(56,59):

bottom\_lip\_pts.append(landmarks[i])

bottom\_lip\_all\_pts = np.squeeze(np.asarray(bottom\_lip\_pts))

bottom\_lip\_mean = np.mean(bottom\_lip\_pts, axis=0)

return int(bottom\_lip\_mean[:,1])

def mouth\_open(image):

landmarks = get\_landmarks(image)

if landmarks == "error":

return image, 0

image\_with\_landmarks = annotate\_landmarks(image, landmarks)

top\_lip\_center = top\_lip(landmarks)

bottom\_lip\_center = bottom\_lip(landmarks)

lip\_distance = abs(top\_lip\_center - bottom\_lip\_center)

return image\_with\_landmarks, lip\_distance

#cv2.imshow('Result', image\_with\_landmarks)

#cv2.imwrite('image\_with\_landmarks.jpg',image\_with\_landmarks)

#cv2.waitKey(0)

#cv2.destroyAllWindows()

cap = cv2.VideoCapture(0)

yawns = 0

yawn\_status = False

while True:

ret, frame = cap.read()

image\_landmarks, lip\_distance = mouth\_open(frame)

height,width = frame.shape[:2]

prev\_yawn\_status = yawn\_status

if lip\_distance > 20:

yawn\_status = True

cv2.putText(frame, "Subject is Yawning", (10,height-40),

cv2.FONT\_HERSHEY\_COMPLEX\_SMALL, 1,(15,15,255),1,cv2.LINE\_AA)

output\_text = " Yawn Count: " + str(yawns + 1)

cv2.putText(frame, output\_text, (10,height-20),

cv2.FONT\_HERSHEY\_COMPLEX\_SMALL, 1,(255,255,255),1,cv2.LINE\_AA)

else:

yawn\_status = False

if prev\_yawn\_status == True and yawn\_status == False:

yawns += 1

if (yawns > 3):

try:

sound.play()

except: # isplaying = False

pass

cv2.imshow('Live Landmarks', image\_landmarks )

cv2.imshow('Yawn Detection', frame )

if cv2.waitKey(1) == 13: #13 is the Enter Key

break

cap.release()

cv2.destroyAllWindows()

GUI.py

import tkinter

import PIL

import os

import PIL.Image, PIL.ImageTk

from sklearn.metrics import pairwise

import pickle

import tensorflow as tf

from tkinter import \*

from tkinter import messagebox

import numpy as np

import os

import cv2

from PIL import Image, ImageDraw

import PIL.ImageOps

from keras.models import load\_model

import tkinter as tk

from PIL import ImageTk, Image

from drowsiness\_detection import fun

from yawn import fun1

sroot = Tk()

sroot.title('Accident Prevention')

sroot.minsize(height=516,width=1020)

sroot.configure(bg='white')

facial="C:/Users/Gurkirat/Desktop/driver.jpg"

img0 = ImageTk.PhotoImage(Image.open(facial))

panel = Button(sroot,image=img0)

panel.image = img0

panel.place(x=0,y=0)

#chilanka

Label(sroot,text=" Accident Prevention ",font='Timesnewroman 40 ',bg='white',fg='black').place(x=535,y=10)

def main():

window=tkinter.Tk()

face="C:/Users/Gurkirat/Desktop/face.jpg"

img = ImageTk.PhotoImage(Image.open(face))

voice="C:/Users/Gurkirat/Desktop/voice.jpg"

img = ImageTk.PhotoImage(Image.open(voice))

window.title('Accident Prevention')

window.geometry('1000x700')

window = Frame( window)

window.place(relx=0.02, rely=0.03, relheight=0.94, relwidth=0.96)

window.configure(relief=GROOVE)

window.configure(borderwidth="2")

window.configure(relief=GROOVE)

window.configure(background="#d9d9d9")

window.configure(highlightbackground="#d9d9d9")

window.configure(highlightcolor="black")

window.configure(width=925)

window.configure(background="#d9d9d9")

window.configure(highlightbackground="#d9d9d9")

window.configure(highlightcolor="black")

lblHeading = Label(window,text = "\t\t\t\t Accident Prevention \t\t\t\t",font=('Calibri',30,'bold'), bg="#d9d9d9",height=3).pack()

b1=Button(window,padx=16,pady=4,bd=4,fg="white",font=('arial',16,'bold'),width=22,height=2,bg="gray25",text='Yawn Detector',command=fun1)

b1.place(x=100,y=420)

b2=Button(window,padx=16,pady=4,bd=4,fg="white",font=('arial',16,'bold'),width=22,height=2,bg="gray25",text='Drowsiness Detector',command=fun)

b2.place(x=550,y=420)

face="C:/Users/Gurkirat/Desktop/yawn.jpg"

img1 = ImageTk.PhotoImage(Image.open(face))

panel = Button(window, image = img1,command=fun1)

panel.image = img1

panel.place(x=100,y=110)

voice="C:/Users/Gurkirat/Desktop/drow.jpg"

img2 = ImageTk.PhotoImage(Image.open(voice))

panel2 = Button(window, image = img2,command=fun)

panel2.image = img2

panel2.place(x=550,y=110)

window.mainloop()

def call\_mainroot():

sroot.destroy()

main()

sroot.after(2500,call\_mainroot)

mainloop()