**DRIVER DROWSINESS DETECTION**

**A Project Report**

***Submitted by:***

**SAHIL RANA**

**(AACS0004A/17)**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

***in partial fulfillment for the award of the degree***

***of***

**at**

****

**COMPUTER SCIENCE AND ENGINEERING**

**ARNI SCHOOL OF TECHNOLOGY**

**KATHGARH (INDORA), HIMACHAL (INDIA)- 176401**

**DECEMBER 2019**

**ARNI UNIVERSITY**

**ACKNOWLEDGEMENT**

“A Qualification of Success is That We Do Not Bring Harmonious and Beneficial Results to Ourselves but also Share Those Benefits with Others.”

First and Foremost, I also owe my sincerest gratitude towards Sh. Anurag Rana Head and Associate Professor supervisor for his valuable advice and healthy criticism throughout my project which help me immensely to complete my work successfully.

I would like to thank Dean Academics and members of the departmental research committee for their valuable suggestion and healthy criticism during my presentation of the work.

The constant guidance and encouragement received from Sh. Anurag Rana Head And Associate Professor of Computer Science & Engineering, (Arni University), has been of help in carrying our present work and is acknowledged with reverential thanks.

I would also like to thank everyone who has knowingly and unknowingly helped me throughout my project.

Last but not the least,a word of thanks for the authors of all those books and papers which I have consulted during my project work as well as for preparing the report.

At the end thanks to the Almighty who made me capable of undertaking and completing this great job.

**Sahil kumar Signature of Student**

**Reg. ID: AACS0004A/17**

**Batch: 2017-21**

**DECLARATION**

I hereby declare that the project entitled "Development of driver drowsiness detection using Machine Learning Algorithm" submitted for the B.Tech. (CSE) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

**Place: Arni University Signature of the Student**

**Date: 07/08/2019**

**CERTIFICATE**

This is certified that the synopsis of Mr. Sahil Kumar, B.Tech. CSE student of Computer Science & Engineering, Dept, Arni School Of Technology, Arni University Kathgarh (Indora) Kangra, H.P (India) has been finalized the work during the academic year 2021. The student has successfully delivered his synopsis seminar entitled **"** **Development of DRIVER DROWSINESS DETECTION ”** on dated 15/05/2021, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering).

Mr. Anurag Rana

**Project Supervisor**

**Counter Signed By**

Signature of Head of the Department Signature of Dean/ Associate Dean

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Place: Date: \_\_\_\_/\_\_\_\_\_\_/20\_\_\_\_\_ .**

**ABSTRACT**

Detection of drowsiness of the driver is a vehicle safety technology, which helps to put off accidents which are caused by the driver being dozy. A variety of studies have recommended that around 20% of all road accidents are due to drowsiness of the driver. The developments of technologies for detecting or preventing drowsiness while driving is a major confront in accident evasion systems. Because of the peril of the tiredness while driving, different new methods need to be developed for counteracting the effect. The paper is based on an example for detection of drowsiness systems. The intent of this paper is design of an automated system for safety of drivers from improper driving. The system is designed such that it will precisely scrutinize the eye blink. In this paper, the eye blink of the driver is detected by using an eye blink sensor which is IR based. The disparity across the eye will vary as per eye blink. The output is high, if the eye is closed or else output is low. It indicates the closing or opening position of an eye. Their output is given to the circuit to signify the alarm. The controller will send a warning signal so that it is displayed on a liquid crystal display screen. The buzzer, which is placed near the driver, will be activated and alters the driver when he falls asleep during driving. The alcohol sensor is also used to detect whether the driver is drunk which avoids accidents caused by the drunken drivers. According to the intensity of light, the lights will be ON or OFF inside the vehicle, this saves power consumption. Tilt sensor is also used to detect whether the vehicle met with an accident or not.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Contents** | **Page No** |
|  | Acknowledgement | II |
|  | Declaration | III |
|  | Certificate | IV |
|  | Abstract | V |
|  | System Requirement | 8 |
|  | **Chapter No. 1:- Introduction to Python + Machine Learning**   * 1. Introduction to Python   2. Introduction to Machine Learning | 4-15 |
|  | **Chapter No. 2:- Tools to be used for Development**  2.1 Introduction to PyCharm**-**Professional IDE for Python Developers  2.2 Introduction to XAMPP  2.3 Introduction to Anaconda | 16 - 18 |
| *8.* | **Chapter No. 3:- Introduction to project**  3.1 Project Description  3.2 Objective  3.3 About the project  3.4 The dataset  3.5 The Model Architecture  3.6 Advantage  3.7 Application  3.8 Disadvantage  3.9 Hardware requirements  3.10 Software requirements  3.11 Existing System Study | 19-21 |
| *9.* | **CHAPTER N0. 4:- Testing**  4.1 Functional Testing  4.2 Structural Testing  4.3 Level of Testing | 22-24 |
| *10.* | **CHAPTER N0. 5:-Different Modules**  5.1 OpenCV  5.2 TensorFlow  5.3 Keras  5.4 Pygame | 25 |
| *11.* | **CHAPTER NO . 6 :- Data Flow Diagram**  6.1 First flow chart  6.2 Second flow chart | 26-27 |
| *12.* | **CHAPTER NO . 7 :- Objective Drowsiness Detection**  7.1 Objective of drowsiness detection | 28 |
| *13.* | **CHAPTER NO . 8 :- Drowsiness Manipulation** | 29 |
| *14.* | **CHAPTER NO.9 :- METHODS FOR MEASURING DROWSINESS.**  9.1 Subjective measures  9.2 Vehicle-Based measures  9.3 Steering wheel Movement  9.4 Standard deviation of Lane Position  9.5 Behavioral measures  9.6 Physiological measures | 30-32 |
| *15.* | **CHAPTER NO.10- PROPOSED SYSTEM** | 33-37 |
| *16.* | **CHAPTER NO.11-FUTURE SCOPE** | 38-39 |
| *17.* | **CHAPTER NO.12-OUTPUT SCREENSHOTS** | 40 |
| *18.* | **CHAPTER NO.13-CODING** | 41-43 |
| *19* | **CHAPTER NO.14-CONCLUSION** | 44 |
| *20* | **CHAPTER NO.15- REFRENCES** | 45 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **S.NO** | **FIGURE NUMBER** | **PAGE NO** |
| 1 | **FIG NO- 1.1,2,3,4,5** | 5,6,12,13,14 |
| 2 | **FIG NO- 2.1,2** | 17,18 |
| 5 | **FIGNO- 5.1** | 20 |
| 6 | **FIG NO-6.1,6.2** | 27,28 |
| 9 | **FIG NO-9.1,9.2** | 32,33 |
| 10 | **FIG NO- 10.1,2,3,4,5,6,** | 34,35,36,37,38,39 |
| 11 | **FIG NO- 11.1** | 40 |
| 12 | **FIG NO 12.1,2** | 41 |

**SYSTEM REQUIREMENTS**

1. **Hardware Requirements:-**

* Intel(R) Celeron(R) N2840 @ 2.16GHz
* 500 GB Hard Disk
* 4 GB of RAM
* Printer
* CDs (Compact Disk) & Pen Drives

1. **Software Requirements:-**

* Operating System: Window 10
* Internet Explorer, Google Chrome
* JetBrains PyCharm Community Edition 2020
* XAMPP
* Anaconda

**CHAPTER NO. 1**

**INTRODUCTION TO PYTHON + MACHINE LEARNING**

**1.1 Introduction to Python:-**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python is a multiparadigm, general-purpose, interpreted, high-level programming language. Python allows programmers to use different programming styles to create simple or complex programs, get quicker results and write code almost as if speaking in a human language. Some of the popular systems and applications that have employed Python during development include Google Search, YouTube, Bit Torrent, Google App Engine, Eve Online, Maya and iRobot machines.



Fig 1.1

**1.1.1 History of Python:-**

Python’s initial development was spearheaded by **Guido van Rossum** in the late 1980s. Today, it is developed by the Python Software Foundation. Because Python is a multi paradigm language, Python programmers can accomplish their tasks using different styles of programming: object oriented, imperative, functional or reflective. Python can be used in Web development, numeric programming, game development, serial port access and more.

There are two attributes that make development time in Python faster than in other programming languages:

Python is an interpreted language, which precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code. Python focuses so much on this abstraction that its code can be understood by most novice programmers.

Python code tends to be shorter than comparable codes. Although Python offers fast development times, it lags slightly in terms of execution time. Compared to fully compiling languages like C and C++, Python programs execute slower. Of course, with the processing speeds of computers these days, the speed differences are usually only observed in benchmarking tests, not in real-world operations. In most cases, Python is already included in Linux distributions and Mac OS X Machines.

**1.1.2 Features of Python:-**

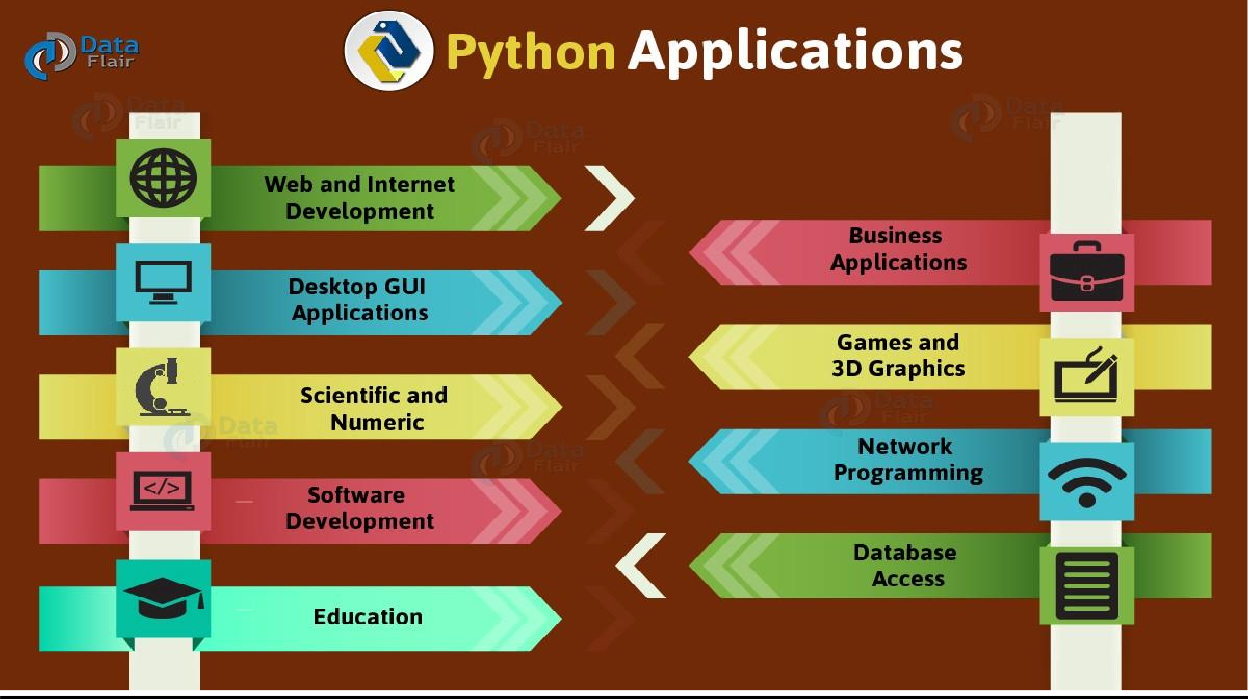
* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.
  + 1. **Applications of Python:-**

fig 1.2

* **GUI-Based Desktop Applications:**

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

1. **Image Processing and Graphic Design Applications:**

Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Light wave and Maya, also use Python in variable proportions.

1. **Scientific and Computational Applications:**

The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as Free CAD, and finite element method software, such as Abaqus , are coded in python.

* **Games:**

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney‟s Toontown Online, Vega Strikeetc.

* **Web Frameworks and Web Applications:**

Python has been used to create a variety of web-frameworks including CherryPy, Django, Turbo Gears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with databases and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; O doo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

* **Enterprise and Business Applications:**

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lips, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

* **Operating Systems:**

Python is often an integral part of Linux distributions. For instance, Ubuntu’s Ubiquity Installer and Fedora’s and Red Hat Enterprise Linux’s Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

* **Language Development:**

Python’s design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple’s Swift, Coffee Script, Cobra, and OCaml all share similarity with Python.

**1.1.4 Basic Concepts of Python:-**

**Beginning with Python programming**

1. **Basics:**

There are two major Python versions- Python 2 and Python 3. Both are quite different.

* **Finding an Interpreter:** Before we start Python programming, we need to have an interpreter to interpret and run our programs.
* **Writing first program:** Following is first program in Python:

# Script Begins

print("HELLO")

# Scripts Ends

Output: WMK

1. **Input/Output:**

* Taking input in Python
* Taking input from console in Python
* Taking multiple inputs from user in Python
* Python Input Methods for Competitive Programming
* Vulnerability in input() function – Python 2.x
* Python | Output using print() function
* How to print without a newline in Python?
* Python | end parameter in print()
* Python | sep parameter in print()
* Python | Output Formatting

1. **Data Types:**

In programming, data type is an important concept. Variables can store data of different types, and different types can do different things. Python has the following data types built-in by default, in these categories:

* Strings
* List
* Tuples
* Sets
* Dictionary
* Arrays Set 1, Set 2

1. **Variables:**

* Variables, expression condition and function
* Maximum possible value of an integer in python?
* Global and local variables in python
* Packing and unpacking arguments in python
* End parameter in Python
* Type conversion in python
* Byte objects vs string in python
* Print single and multiple variable
* Swap variable
* Private variables
* \_\_name\_\_ (A Special variable) in Python

1. **Operators:**

* Basic operator in python
* Logical and bitwise not operator on boolean
* Ternary operator
* Division operator in python
* Operator Overloading in Python
* Any & all in python
* Inplace and standard operators in python
* Operator function in python | Set – 1
* Operator function in python | Set – 2
* Inplace operator | Set -1
* Inplace operator | Set -2
* Logic Gates in Python
* Python | a += b is not always a = a + b
* Difference between == and is operator in Python
* Python Membership and Identity Operators | in, not in, is, is not Operators

1. **Control Flow:**

* Loops
* Loops and Control Statements (continue, break and pass) in Python
* Looping technique in python
* range vs xrange in python
* Programs for printing pyramid technique in python
* Chaining comparison in python
* else with for
* switch function
* Coroutine in Python
* Using iteration in python effectively
* Iterators in Python
* Iterators function in python| Set – 1
* Iterators function in python | set -2
* Python \_\_iter\_\_() and \_\_next\_\_() | Converting an object into an iterator
* Python | Difference between iterable and iterator
* Generators in python
* Generators expression in python

1. **Functions:**

* Functions in Python
* class method vs static method in Python
* Write an empty function in Python – pass statement
* Yield instead of Return
* Return Multiple Values
* Partial Functions in Python
* First Class functions in Python
* Precision Handling
* \*args and \*\*kwargs
* Python closures
* Function Decorators
* Decorators in Python
* Decorators with parameters in Python
* Memoization using decorators in Python
* Help function in Python
* Python | \_\_import\_\_() function
* Python | range() does not return an iterator
* Coroutine
* Python bit functions on int (bit\_length, to\_bytes and from\_bytes)

1. **Object Oriented Concepts:**

* Python3 Intermediate Level Topics
* Class, Object and Members
* Data Hiding and Object Printing
* Inheritance, examples of object, issubclass and super
* Polymorphism in Python
* Class and static variable in python
* Class method and static method in python
* Changing class members
* Constructors in Python
* Destructors in Python
* First class function
* str() vs repr()
* str() vs vpr()
* Metaprogramming with metaclasses
* Class and instance attribute
* Reflection
* Barrier objects
* Timer objects
* Garbage collection

1. **Exception Handling**

* Exception handling
* User defined Exception
* Built-in Exception
* clean up action
* Nzec error
* Python | 404 Error handling in Flask
* try and except in Python
* Defining Clean Up Actions in Python

1. **Modules In Python:**

* Consider a module to be the same as a code library.
* A file containing a set of functions you want to include in your application.
* Numeric Functions
* Logarithmic and Power Functions
* Trigonometric and Angular Functions
* Special Functions and Constants
* Inplace Operators – Set 1 Set 2
* Calendar Module
* Calendar Functions – Set 1 Set 2
* Complex Numbers – Introduction
* Complex Numbers -Important Functions and Constants
* Trigonometric and Hyperbolic Functions
* Time Functions in Python – Set 1, Set-2
* Random Numbers
* struct() module
* Python Urllib Module
* print
* eval()
* fraction()
* Pyautogui – Mouse Keyboard Automation
* Generate QR Code using pyqrcode module

1. **Libraries And Function:**

* Timeit function
* Numpy- Set 1, Set 2
* Get and Post
* Regular Expressions – Set 1, Set 2
* OS module
* Copy module – Deep vs Shallow copy
* Import module
* Reload module
* Collection module
* Dequeue
* Named Tuple
* Heap
* Enum module
* Theano in Python
* Statistical Functions – Set 1, Set 2
* Bisect module
* Python math library | gamma() function
* Python math library | expm1() method
* Decimal Functions – Set 1, Set 2
* NetworkX
* getpass() and getuser()
* Reading and Generating QR codes
* Fnmatch
* Python | Matplotlib.pyplot ticks
* Unicodedata – Unicode Database
* Text Wrap – Text wrapping and filling
* Generating Secure random numbers
* pickle — Python object serialization
* Understanding Python Pickling with example
* copyreg — Register pickle support functions
* Python GUI – tkinter

1. **Python GUI (Graphical User Interface):**

* Python GUI – Tkinter
* Distance-time GUI calculator using Tkinter
* Simple GUI calculator using Tkinter
* Simple registration form using Tkinter
* Create a stopwatch using python
* Designing GUI applications Using PyQt
* CGI Programming in Python
* Color game using Tkinter in Python
* Make Notepad using Tkinter
* Message Encode-Decode using Tkinter
* Real time currency convertor using Tkinter
  1. **Introduction to Machine Learning:-**

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future.

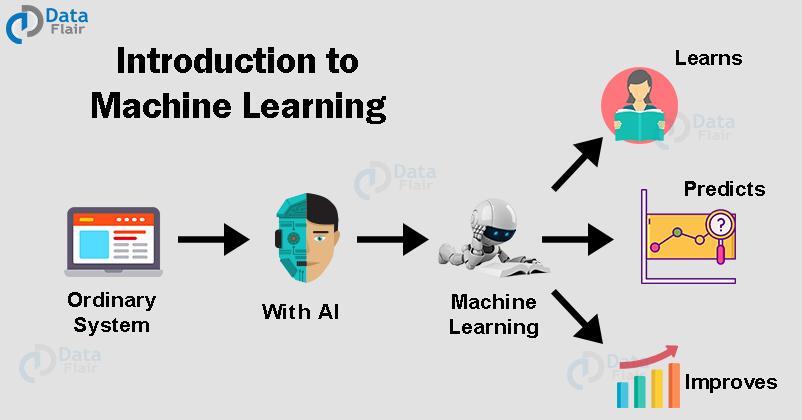


Fig 1.3

## 1.2.1 Machine Learning – Types of Techniques:-

## There are three types of ML Techniques:

1. **Supervised Learning:-** Supervised learning as the name suggests getting supervised by someone. It is a learning in which the machine uses data which is already tagged with the correct answer. After that, the machine is provided with a new set of data. With the help of supervised learning, the algorithm analyzes the training data (set of training examples) and produces a correct outcome from labeled data. Here the machine has already learned the things from previous data. So, now is the time it uses the learning wisely.

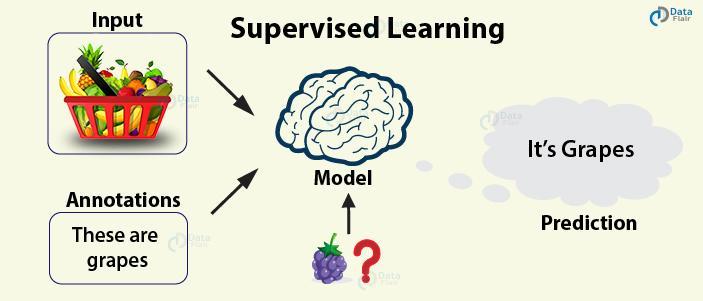


Fig 1.4

**For example –** if we take a fruit basket, the machine will first classify the fruit with its shape and color and would confirm the fruit name. If one searches for grapes, then machine learning from its training data (basket containing fruits) will use the prior knowledge.

It will then apply the knowledge to test data and will then provide you with the results.

1. **Unsupervised Learning:-** In unsupervised learning, the training of the machine is done using the information which is neither classified nor labeled. The machine learning algorithm acts on information without guidance. It groups unsorted information according to similarities, patterns, and differences without any prior training or supervision.

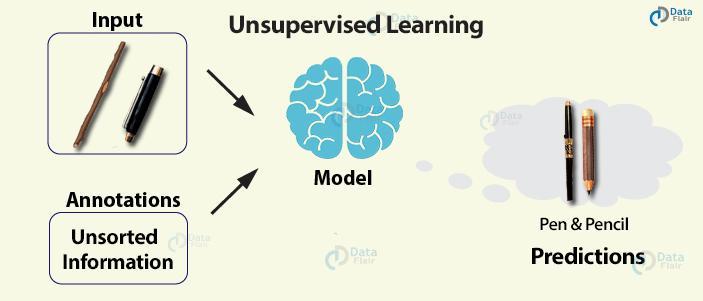
Since there is no training given to the machine, the machine itself finds the hidden structure in unlabeled data and interprets it.

Fig 1.5

**For example –** a wooden stick with a cap can be a pen and with no cap a pencil. With no learning and no training, the machine tries to interpret itself.

1. **Reinforcement Learning:-** Reinforcement learning is a very interesting kind of learning. There’s no answer key which can tell what’s right. But, the reinforcement learning agent still decides how to act to perform its task. This machine learning technique is all about taking actions that are suitable and maximize the reward in a particular situation. It is when the learner receives rewards and punishments for their actions.

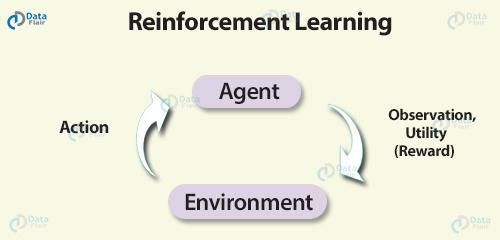


fig 1.6

**For example –** In a given scenario, the reward could be utility and the agent could be told to receive as much utility as possible in order to “win”.

**1.2.2 Applications of Machine Learning:-**

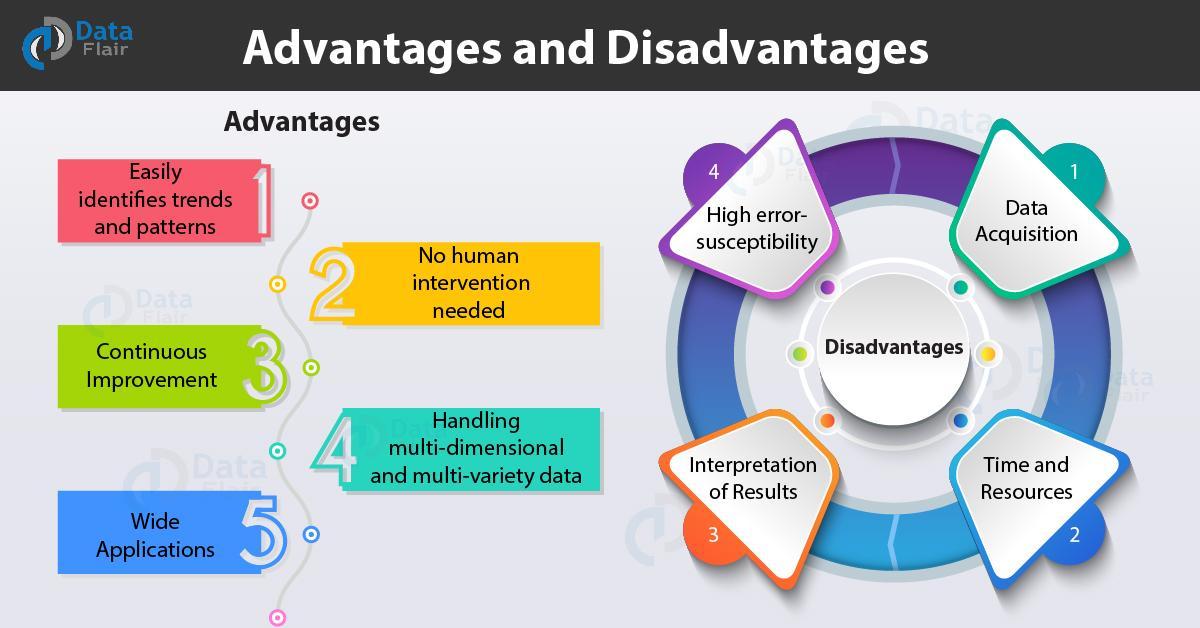


fig 1.7

**Advantages of Machine learning:-**

1. **Easily identifies trends and patterns:-**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

1. **No human intervention needed (automation):-**

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

1. **Continuous Improvement:-**

As [ML algorithms](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

1. **Handling multidimensional and multi-variety data:-**

Machine Learning algorithms are good at handling data that are multidimensional and multi-variety, and they can do this in dynamic or uncertain environments.

1. **Wide Applications:-**

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

**Disadvantages of Machine Learning:-**

With all those advantages to its powerfulness and popularity, Machine Learning isn’t perfect. The following factors serve to limit it:

1. **Data Acquisition:-**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

1. **Time and Resources:-**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

1. **Interpretation of Results:-**

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

1. **High error-susceptibility:-**

[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**CHAPTER NO. 2**

**TOOLS TO BE USED FOR DEVELOPMENT**

**2.1 Introduction to Pycharm-Professional IDE for Python Developers:-**

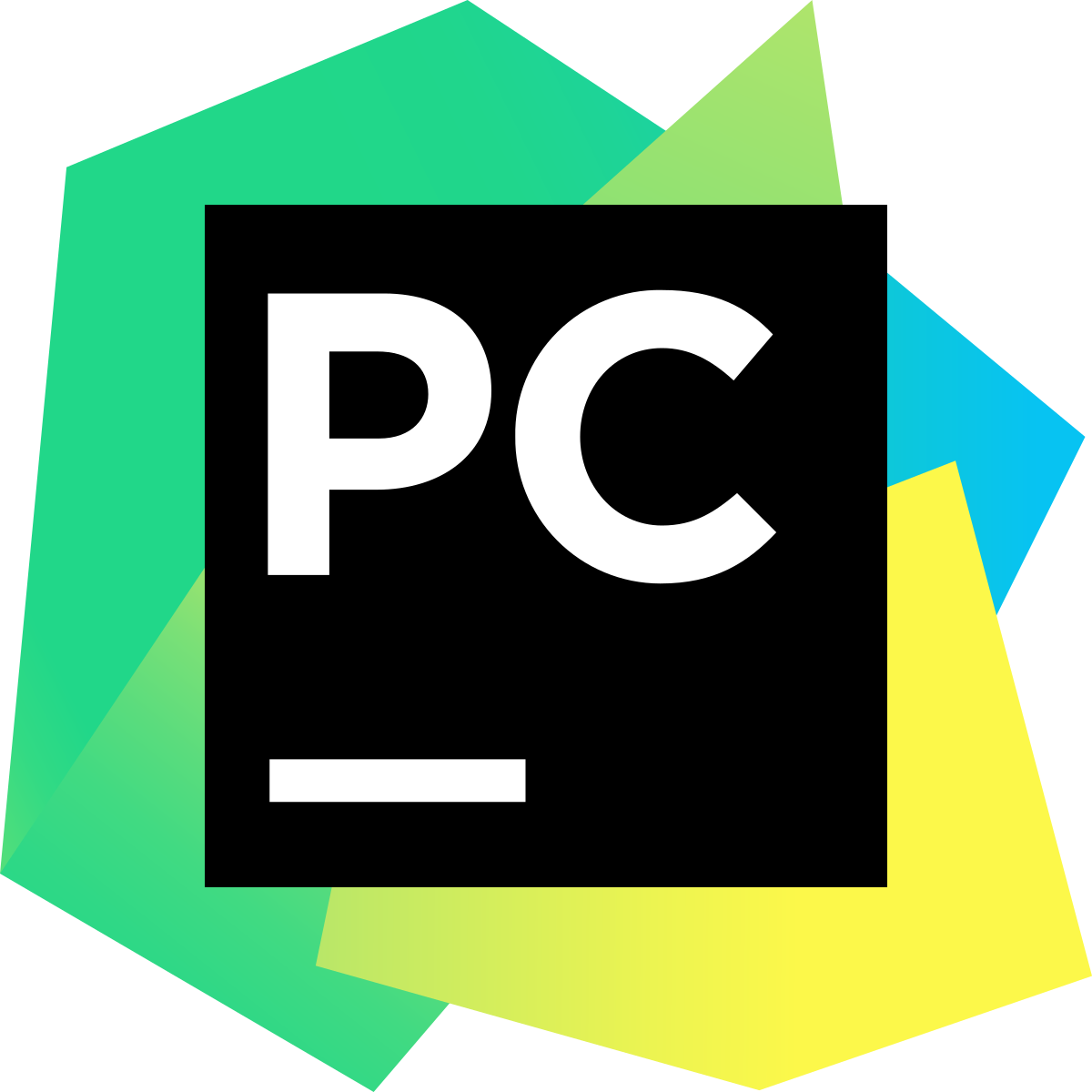


Fig 2.1

PyCharm is in the Professional Python Developers toolbox as an important tool to make their lives easier and more productive. As any tool, it can be used to different extent. Some users will use basic features and will feel happy about it while others, more curious, will explore possibilities and get the maximum out of it.

It’s your choice, but we want to make sure that you know enough to decide on using a given feature or configuration, or not, with confidence. We will invest time and efforts to show you quick tips & tricks that will help you to speed up your flow, we will invite industry experts to showcase projects and best practices, we will have tutorials showcasing different frameworks and how to best adapt to the latest changes in the Python ecosystem, and much more!

**JetBrains**

Since its foundation in the early 2000s, JetBrains has been working hard to make the best and most effective developer tools on earth. Fast forward to the present days and we are proud of our product line with products that, together, are used by over 8 million users all over the world. PyCharm is one of the products in our portfolio and is used by millions of developers to build projects that are changing the world that we know now.

* 1. **Introduction to XAMPP:-**

**XAMPP** is a [free and open source](https://en.wikipedia.org/wiki/Free_software) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends, consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language). XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.



fig 2.2

* 1. **Introduction to Anaconda:-**

Anaconda is an open-source distribution for python and R. It is used for [data science](https://www.edureka.co/blog/how-to-learn-data-science/), [machine learning](https://www.edureka.co/blog/introduction-to-machine-learning/), [deep learning](https://www.edureka.co/blog/deep-learning-with-python/), etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science.



Fig 2.3

Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to easily collect data from various sources using various machine learning and AI algorithms. It helps in getting an easily manageable environment setup which can deploy any project with the click of a single button.

Now that we know what anaconda is, let’s try to understand how we can install anaconda and set up an environment to work on our systems.

**CHAPTER No. 3**

**Introduction to Project**

* 1. **Project Description:-**

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving. A new approach towards automobile safety and security with autonomous region primarily based automatic automotive systems is projected during this conception. We have a tendency to propose distinct however closely connected ideas. Drowsy driver detection system and a traffic detection system with external vehicle intrusion doding primarily based conception. In recent times automobile fatigue connected crashes have very enlarged. So as to attenuate these problems, we’ve incorporated a driver alert system by watching each driver's eyes still as sensing still because the driver state of affairs based primarily based native setting recognition based AI systems are projected.

Nowadays, more and more professions require long-term concentration. Drivers must keep a close eye on the road, so they can react to sudden events immediately. Driver fatigue often becomes a direct cause of many traffic accidents. Therefore, there is a need to develop the systems that will detect and notify a driver of her/him bad psychophysical condition, which could significantly reduce the number of fatigue-related car accidents. However, the development of such systems encounters many difficulties related to fast and proper recognition of a driver’s fatigue symptoms. One of the technical possibilities to implement driver drowsiness detection systems is to use the vision-based approach. This article presents the currently used driver drowsiness detection systems. Here we are detecting the driver's drowsiness by estimating his vision system.

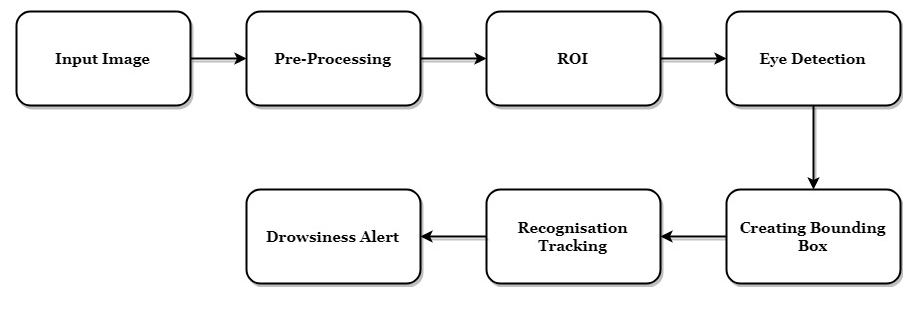


fig 3.1

* 1. **Objective:-**

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. The primary objective of the proposed system is to overcome one or more problems of the existing systems.

* To capture and preprocess the driver image
* To find eye detection fastly using non-intrusive method
* To find eye state fastly
* To wake up the driver to avoid accidents
* To protect the people and vehicle
* To provide a system that concentrates on a non-intrusive vision based system
* To provide a Driver Drowsy Detection System that will work well in Day and night irrespective of lighting condition
* To provide for increased efficiency by avoiding frame loss
* To provide a Driver Drowsy Detection System that uses customized image processing algorithm to concentrate on the eyes detection
* To provide knowledge based method to authenticate the eyes movement using the image-processing algorithm to reduce the chance of misdetection or wrong alarm
* To protect the inmates, driver, people outside the vehicle, vehicle, the surrounding environment and other vehicles which are coming in the opposite direction
* Finding a vehicle driver who is sleepy and tired to stop the driving in order to avoid accidents.
  1. **About the Project:-**

In this Python project, we will be using OpenCV for gathering the images from a webcam and feed them into a [***Deep Learning***](https://data-flair.training/blogs/deep-learning-tutorial/) 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 an 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 –** Classifiers will categorize whether eyes are open or closed.

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

* 1. **The Dataset:-**

The dataset used for this model is 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.

* 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.

* 1. **Advantages:-**
* Region of interest is clear to identify
* Bounding box creation and tracking
  1. **Applications:-**
* Real time tracking applications
* Classification of small particles

**3.8 Disadvantages:-**

* With 80% accuracy, it is obvious that there are limitations in the system. The most significant limitation is that there cannot be any reflective object behind the driver. The more uniform the background is, the more robust the system becomes.
* For testing, rapid head movement was not allowed, since it can be equivalent to simulating a tired driver. For small head movements, the system rarely loses track of the eyes. When the head is turned too much sideways there were some false alarms.
* The system has problems little when the person is wearing eyeglasses. Localizing the eyes is not a problem, but the point is to determine whether the eyes are opened or closed.

**3.9 Hardware Requirements:-**

* Camera
* Personal computer

**3.10 Software Requirements:-**

* Python 2.7 or above versions
* Anaconda software

**3.11 Existing System Study**

By using non intrusive machine vision based concepts, drowsiness of the driver detected system is developed. Many existing systems require a camera which is installed in front of the driver. It points straight towards the face of the driver and monitors the driver‟s eyes in order to identify the drowsiness. For large vehicles such as heavy trucks and buses this arrangement is not pertinent. Bus has a large front glass window to have a broad view for safe driving. If we place a camera on the window of front glass, the camera blocks the frontal view of the driver so it is not practical. If the camera is placed on the frame which is just about the window, then the camera is unable to detain the anterior view of the face of the driver correctly. The open CV detector detects only 40% of the face of the driver in normal driving position in video recording of 10 minutes. In the oblique view, the OpenCV eye detector (CV-ED) frequently fails to trace the pair of eyes. If the eyes are closed for five successive frames the system concludes that the driver is declining slumbering and issues a warning signal. Hence the existing system is not applicable for large vehicles. In order to conquer the problem of the existing system, a new detection system is developed in this project work.

**CHAPTER NO. 4**

**TESTING**

* 1. **Functional Testing:-**

Function testing verifies that each function of the software application operates in conformance with the requirement specification. The testing mainly involves black box testing and it is not concerned with the source code. The functionality of the system by providing appropriate input, verifying the output and comparing the actual result with expected result. Functional testing can be performed on survival of the fastest to check that the appropriate output is coming from the input provided. Each Activity in Survival of the fastest includes validation in which the next activity is dependent upon the user choice.

* 1. **Structural Testing:-**

The structural testing is the testing of the structure of the system or component. Structural testing is often referred to as ‘white box’ or ‘glass box’ or ‘clear-box testing’ because in structural testing, testing is done to check what is happening inside the system. In structural testing the testers are required to have the knowledge of the internal implementations of the code. Here the testers require knowledge of how the software is implemented, how it works.

Survival of the fastest structural testing is done to check how the loops in the software are working. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software. Structural testing checks whether the code written is correct or not. The code in tour and travel is written correctly and is error free. It check that the menu option is working properly or not. And all the menu Items including the correct link to the correct from.

* 1. **Level of Testing:-**

There are different levels during the process of Testing.Levels of testing include the different methodologies that can be used while conducting Software Testing. Following are the main levels of Software Testing:

1. **Functional Testing:**

Functional testing can be performed on survival of the fastest to check that the appropriate output is coming from the input provided. Each Activity in Survival of the fastest includes validation in which the next activity is dependent upon the user choice.

1. **Unit Testing:**

Is this testing done by the developers on the individual units of the source code assigned area. The developers use test data that is separate from the test data of the quality assurance team. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

1. **Integration Testing:**

The testing of combined parts of an application to determine if they function correctly together is Integration testing. In this phase of testing we are emphasizing on how two or more applications or functions are working with each other like how their activities depend upon each other, which activity comes under which module.

1. **System Testing:**

This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.

* This type of testing is performed by a specialized testing team.
* System Testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.
* The application is tested thoroughly to verify that it meets the functional and technical specifications.
* The application is tested in an environment which is very close to the production environment where the application will be deployed.
* System Testing enables us to test, verify and validate both the business requirements as well as the Applications Architecture

1. **Alpha Testing:**

This test is the first stage of testing and will be performed amongst the teams (developer and QA teams). Unit testing, integration testing and system testing when combined are known as alpha testing. During this phase, the following will be tested in the application:

* Spelling Mistakes
* Broken Links
* Cloudy Directions
* The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

1. **Beta Testing:**

This test is performed after Alpha testing has been successfully performed. In beta testing a sample of the intended audience tests the application. Beta testing is also known as pre-release testing. Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program a "real-world" test and partly to provide a preview of the next release. In this phase the audience will be testing the following:

* Users will install, run the application and send their feedback to the project team. Typographical errors, confusing application flow, and even crashes.
* Getting the feedback, the project team can fix the problems before releasing the software to the actual users.
* The more issues you fix that solve real user problems, the higher the quality of your application will be.
* Having a higher-quality application when you release to the general public will increase customer satisfaction

1. **Non-Functional Testing**:

This section is based upon the testing of the application from its non-functional attributes. Nonfunctional testing of Software involves testing the Software from the requirements which are non-functional in nature related but important such as performance, security, user interface etc.

Some of the important and commonly used non-functional testing types are mentioned as follows:

* **Performance Testing:**

It is mostly used to identify any bottlenecks or performance issues rather than finding the bugs in software. There are different causes which contribute in lowering the performance of software:

* Network delay
* Client side processing.
* Database transaction processing.
* Load balancing between servers.
* Data rendering.
* Performance testing is considered as one of the important and mandatory testing type in terms of following aspects:
* Speed (i.e. Response Time, data rendering and accessing)
* Capacity
* Stability
* Scalability Section

**CHAPTER NO. 5**

**DIFFERENT MODULES**

This project uses different modules to complete its coding which are described below:

1. **OpenCV –** pip install opencv-python (face and eye detection). OpenCV is one of the most popular computer vision libraries. It is used for:**-**

* Reading an image
* Extracting the RGB values of a pixel
* Extracting the Region of Interest (ROI)
* Resizing the Image
* Rotating the Image
* Drawing a Rectangle
* Displaying text

1. **TensorFlow –** pip install tensorflow (keras uses TensorFlow as backend). TensorFlow is an open-source software library. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google’s Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well!
2. **Keras –** pip install keras (to build our classification model). Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development.
3. **Pygame –** pip install pygame (to play alarm sound). This library allows you to create games and rich multimedia programs in **Python**.

**CHAPTER NO. 6**

**DATA FLOW DIAGRAMS**

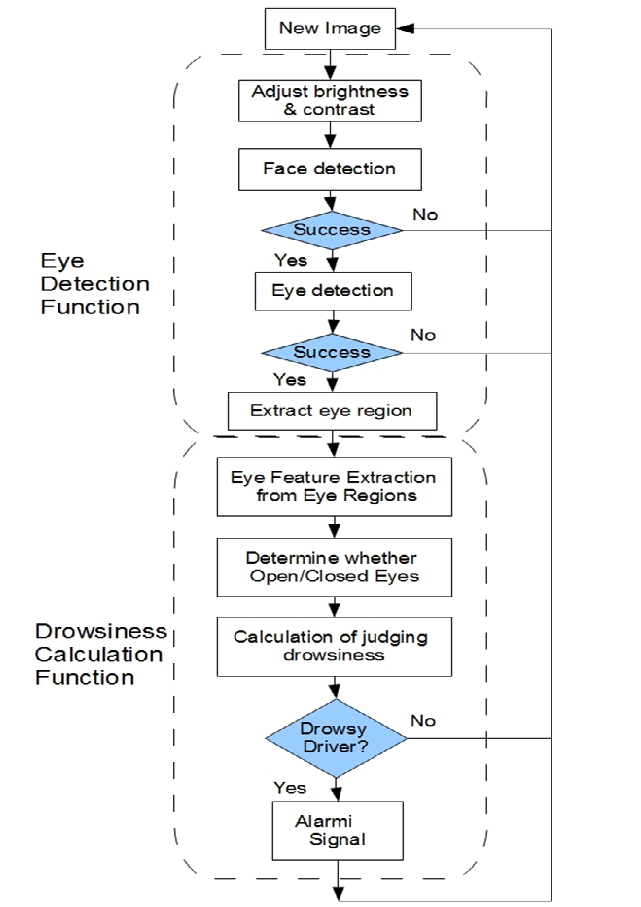
**6.1 First Flow Chart:-**

Fig 6.1

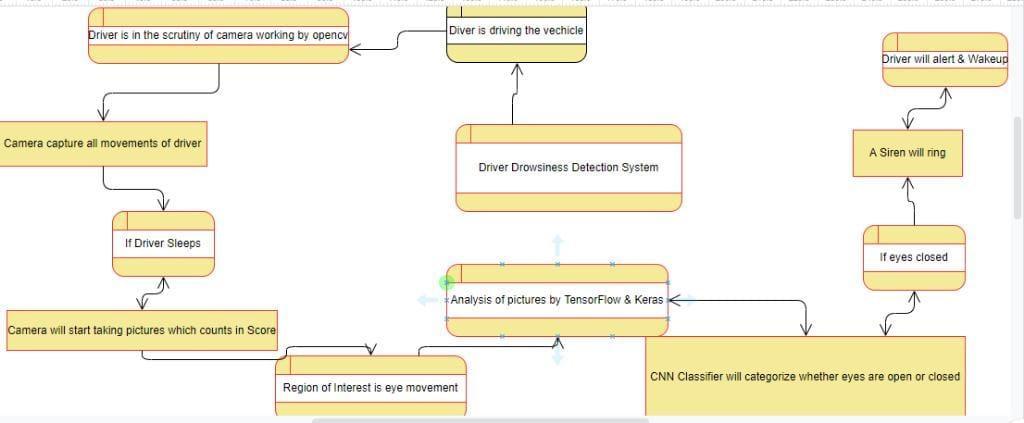
**6.2 Second Flow Chart:-**

Fig 6.2

**CHAPTER NO. 7**

**Objective of Drowsiness Detection**

1. The main aim of our project is to develop non- non-intrusive system which will detect the fatigue or drowsiness of the driver and will issue a warning with the help of an alarm.

2. Driver drowsiness detection is a car safety technology which helps to save the life of the driver by preventing accidents when the driver is getting drowsy.

3. The main objective is to first design a system to detect driver’s drowsiness by continuously monitoring the retina of the eye.

4. The system works in spite of drivers wearing spectacles and in various lighting conditions.

5. To alert the driver on the detection of drowsiness by using buzzer or alarm

6. Speed of the vehicle can be reduced.

7. Traffic management can be maintained by reducing the accidents

**CHAPTER.8**

**DROWSINESS MANIPULATION**

One of the challenges in developing an efficient drowsiness detection system is how to obtain proper drowsiness data

**Driver drowsiness mainly depends on:-**

1. The quality of the last sleep.
2. The circadian rhythm (time of day) .
3. The increase in the duration of the driving task.

In addition, researchers have found that drowsiness-related crashes are more probable in a monotonous environment than in a stimulating environment.

Therefore, there is a very high probability that a partially sleep-deprived driver will become drowsy when driving in a monotonous environment at a constant speed for three hours during a time when their circadian rhythm is low. This should be taken into consideration when designing an experiment relating to recording driver drowsiness.

**CHAPTER.9**

**METHODS FOR MEASURING DROWSINESS**

Researchers have used various methods to measure driver drowsiness. This section provides a review of the four most widely-used methods, among which the first method is measured either verbally or through questionnaire and the remaining three by means of various sensors.

1. **Subjective measures:**

Subjective measures that evaluate the level of drowsiness are based on the driver’s personal estimation and many tools have been used to translate this rating to a measure of driver drowsiness. The most commonly used drowsiness scale is the Karolinska Sleepiness Scale (KSS), a nine-point scale that has verbal anchors for each step.

1. **Vehicle-Based measures :**

Another method to measure driver drowsiness involves vehicle-based measurements. In most cases, these measurements are determined in a simulated environment by placing sensors on various vehicle components, including the steering wheel and the acceleration pedal; the signals sent by the sensors are then analyzed to determine the level of drowsiness.

1. **Steering wheel Movement (SWM):**

Measured using a steering angle sensor and it is a widely used vehicle-based measure for detecting the level of driver drowsiness . Using an angle sensor mounted on the steering column, the driver’s steering behavior is measured. When drowsy, the number of micro-corrections on the steering wheel reduces compared to normal driving . Fairclough and Graham found that sleep deprived drivers made fewer steering wheel reversals than normal drivers. To eliminate the effect of lane changes, the researchers considered only small steering wheel movements (between 0.5° and 5°), which are needed to adjust the lateral position within the lane . Hence, based on small SWMs, it is possible to determine the drowsiness state of the driver and thus provide an alert if needed. In a simulated environment, light side winds that pushed the car to the right side of the road were added along a curved road in order to create variations in the lateral position and force the drivers to make corrective SWMs . Car companies, such as Nissan and Renault, have adopted SWMs but it works in very limited situations. This is because they can function reliably only at particular environments and are too dependent on the geometric characteristics of the road and to a lesser extent on the kinetic characteristics of the vehicle.

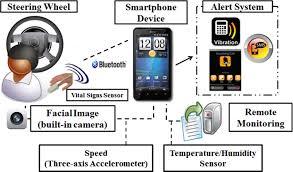


Fig 9.1

1. **Standard deviation of Lane Position:**

Is another measure through which the level of driver drowsiness can be evaluated . In a simulated environment, the software itself gives the SDLP and in case of field experiments the position of lane is tracked using an external camera

**3. Behavioral measures :**

A drowsy person displays a number of characteristic facial movements, including rapid and constant blinking, nodding or swinging their head, and frequent yawning . Computerized, non-intrusive, behavioral approaches are widely used for determining the drowsiness level of drivers by measuring their abnormal behaviors . Most of the published studies on using behavioral approaches to determine drowsiness, focus on blinking . PERCLOS (which is the percentage of eyelid closure over the pupil over time, reflecting slow eyelid closures, or “droops”, rather than blinks) has been analyzed in many studies . This measurement has been found to be a reliable measure to predict drowsiness and has been used in commercial products such as Seeing Machines and Lexus . Some researchers used multiple facial actions, including inner brow rise, outer brow rise, lip stretch, jaw drop and eye blink, to detect drowsiness . However, research on using other behavioral measures, such as yawning and head or eye position orientation.

**4. Physiological measures:**

As drivers become drowsy, their head begins to sway and the vehicle may wander away from the center of the lane. The previously described vehicle-based and vision based measures become apparent only after the driver starts to sleep, which is often too late to prevent an accident.

However, physiological signals start to change in earlier stages of drowsiness. Hence, physiological signals are more suitable to detect drowsiness with few false positives; making it possible to alert a drowsy driver in a timely manner and thereby prevent many road accidents.

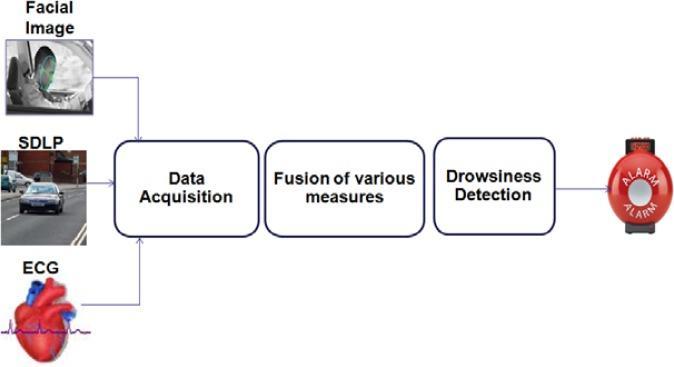
****

fig 9.2

**CHAPTER.10**

**PROPOSED SYSTEM**

In order to overcome this eye blink sensor is used. A spectacle with eye blink sensor is used to detect the driver drowsiness and alerts the driver with buzzer, if the driver is affected by drowsiness. The various hardware components of project are mentioned below :

1. LPC2148 microcontroller
2. Eyeblink Sensor
3. Alcohol Sensor
4. Tilt Sensor
5. LDR
6. LCD & Buzzer
7. GSM and GPS

**1. Power supply**

In this system, the power supply of 12V is used for the transmitter section and receiver section. For conversion of the A.C. to D.C rectifiers are being used. To step down the voltage, a step down transformer is used.

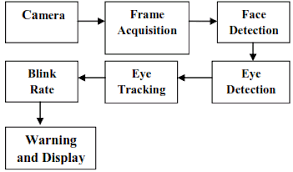


Fig 10.1

**2.Microcontroller**

Here the microcontroller section is the control unit of the project. It consists of a Microcontroller with its related circuitry like Crystal with capacitors, Pull up resistors, and reset circuitry and so on. The heart of the project is the microcontroller because it controls the interfaced devices. By using a written program it communicates with the devices. Here 16-bit/32-bit ARM7TDMI-S controller with a small leadless quad flat package. It supports up to 40KB RAM and 512KB on chip flash ROM and supports crystal frequency of 60MHz used at high speed operation.

**3. Eye blink sensor**

An eye blinking is mandatory in this work, since it is used to impel the device and to activate events. Instructions were written in image processing that if there is no eyelid movement established for the definite phase of preset i.f. time greater than the time of eye blinking of a normal human then considered as “blink” . In this paper time is to be set as 5 seconds or more than it, as “blink event” is distinct from “normal eye blinking”. The test is to be conducted for normal blinking of the human eye.

1. **IR Sensor**

The IR transmitter and IR receiver are arranged in parallel. When the signal is given, the IR sensor starts functioning and the IR transmitter emits the infrared rays to the receiver. The comparator is coupled with an IR receiver. The operational amplifier is attached to the comparator.To the inverting input terminal of the comparator the reference voltage is given, the comparator is linked to the receiver. When a disruption is present in the IR rays between sender (transmitter) and recipient (receiver), the IR receiver will not conduct. Hence the voltage at the inverting input terminal is lower than the voltage at the non inverting input. Therefore the output of the comparator is high. The output voltage of the comparator is given to the microcontroller. When the IR receiver receives the rays from the transmitter, the IR receiver becomes conducting since the voltage at the non inverting terminal is lower than voltage at the inverting terminal. Therefore the output of the comparator is low. Hence the output of the comparator is set to the controller. This circuit is used for counting eyelid movement.

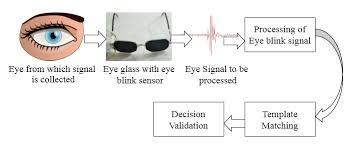


Fig 10.2

**4.ALCOHOL SENSOR**

The concentration of alcohol in our breath is detected by using an alcohol sensor. Sensor provides an analog output based on alcohol concentration. MQ-3 sensor is used as an alcohol sensor . The alcohol sensor has an exceptional sensitivity and faster response time. The isobutane, propane, alcohol, cigarette, smoke, LNG are sensed by the sensor.

**5. TILT Sensor**

According to variations in angular movement of an object, tilt sensors will produce an electrical signal. These sensors are used to enumerate tilt and slope within a narrow range of movement. Sometimes, the inclinometers are referred to as tilt sensors because the sensor generates a signal but the inclinometer generates both readout and a signal. It is used to detect the fall. It acts as a switch.

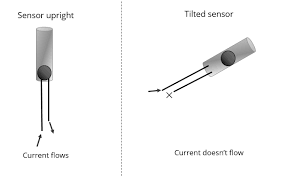


Fig 10.3

**6. LIGHT DEPENDENT RESISTOR**

A Light dependent resistor (LDR) is also called a photoresistor or a cadmium sulfide (CdS) cell is shown in fig 5. It is also known as a photoconductor.

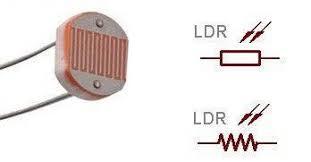




Fig 10.4,5

**7.Global Positioning System**

The location of any object or vehicle is determined by using satellite signals in Global positioning system (GPS) technology [5]. To find the receiver in 3D space three satellite signals are needed and for time accuracy the fourth satellite is used. The longitude, latitude and attitude parameters information are given by the GPS. By using longitude, latitude and attitude parameters location of any object can be estimated. In GPS technology, the communication is amid GPS transceiver and GPS satellite.



Fig 10.6

**8.Global system for mobile communication**

Global system for mobile communication (GSM) is the technology that underpins most of the world‟s mobile phone networks . The GSM technology is extremely booming wireless technology along with a unique story of global attainment with increase in collaboration. GSM is the fastest mounting communication technology. GSM is digital cellular technology. For mobile voice and data service transmission and reception digital cellular technology is used. In GSM technology speed of data transfer is up to 9.6 kbps and allows the broadcast of basic data services such as SMS.

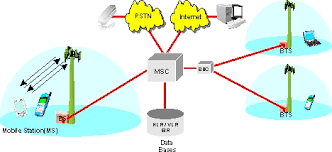
****

fig 10.7

**CHAPTER.11**

**FUTURE SCOPE**

Our Aim Is To Detect Drowsiness And In Future This System Will Be Integral Part Of Safety System In Vehicles And Used To Save Many Lives. In Future This Prototype Will Be Extended To Add Few More Features To Detect Driver Drowsiness. In Future This Prototype Will Be Extended To Add Few More Features To Detect Driver Drowsiness. In Future We Are Planning To Add Yawn Detection Method, Detection Of Sleep By Monitoring Head Movement. Further, This Prototype Will Be Extended To Monitor The Reflected Ray From Eye Using Nano Camera. If The Reflection Ray Is Absent, Then Eye Is Closed Otherwise Eye Is Opened. We Believe That This Will Create A Better Opportunity To Detect Drowsiness.

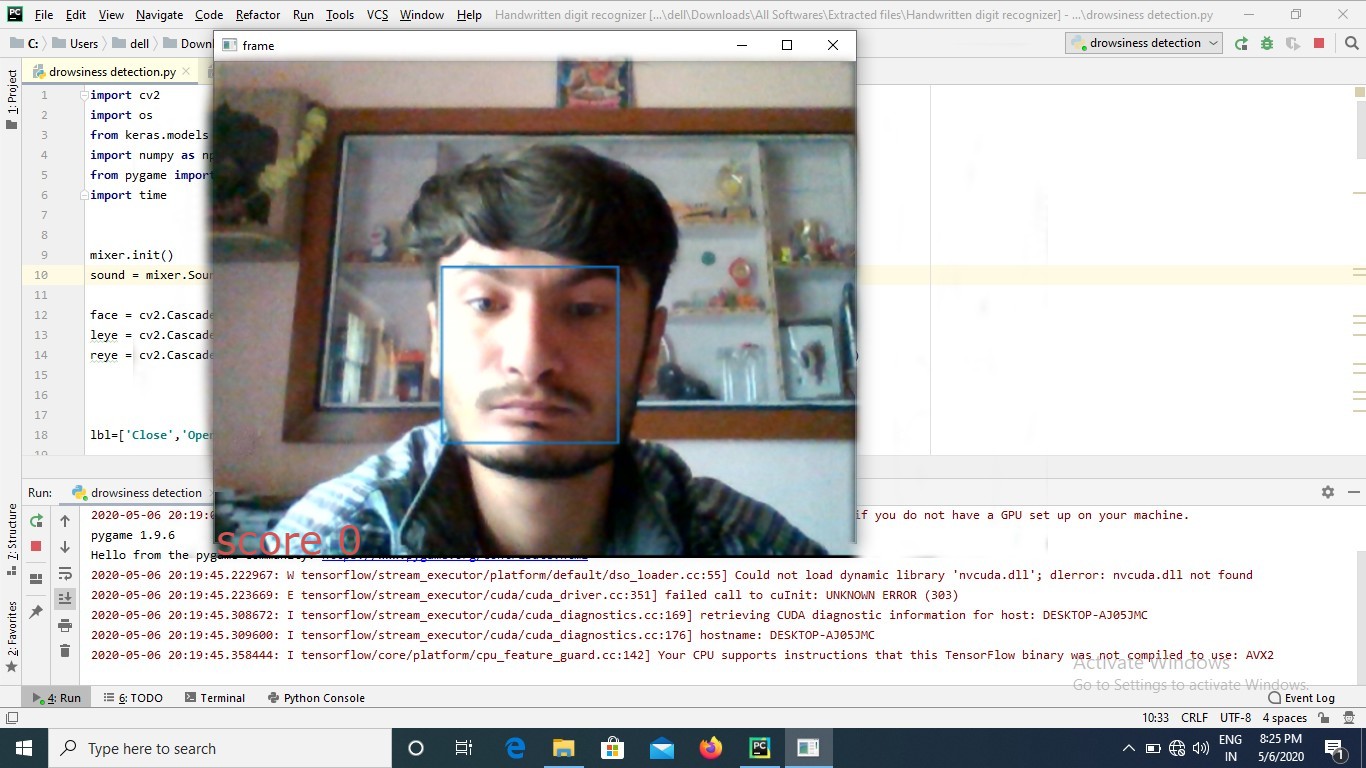
1. This project can be implemented in the form of mobile application to reduce the cost of hardware.
2. This project can be integrated with a car, so that automatic speed control can be imparted if the driver is found sleeping.
3. Moving forward, there are a few things we can do to further improve our results and fine-tune the models.
4. First, we need to incorporate distance between the facial landmarks to account for any movement by the subject in the video.
5. Realistically the participants will not be static on the screen and we believe sudden movements by the participant may signal drowsiness or waking up from micro-sleep.
6. Second, we want to update parameters with our more complex models (NNs, ensembles, etc.) in order to achieve better results.
7. The future works may focus on the utilization of outer factors such as vehicle states, sleeping hours, weather conditions, mechanical data, etc, for fatigue measurement.
8. Driver drowsiness poses a major threat to highway safety, and the problem is particularly severe for commercial motor vehicle operators.
9. Twenty-four hour operations, high annual mileage, exposure to challenging environmental conditions, and demanding work schedules all contribute to this serious safety issue
10. Monitoring the driver’s state of drowsiness and vigilance and providing feedback on their condition so that they can take appropriate action is one crucial step in a series of preventive measures necessary to address this problem.
11. Currently there is no adjustment in zoom or direction of the camera during operation. Future work may be to automatically zoom in on the eyes once they are localized.

****

fig 11.1

**CHAPTER NO. 12**

**OUTPUT SCREENSHOTS**



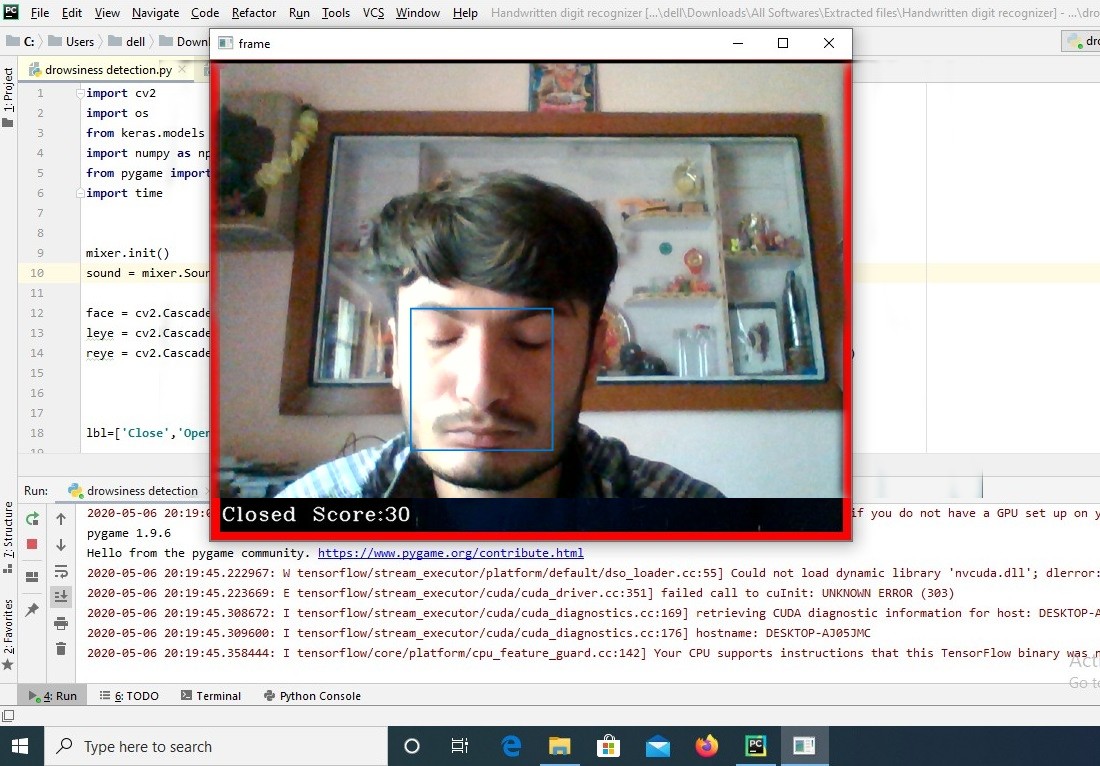


Fig 12.1,2

**CHAPTER NO. 13**

**CODING**

import cv2

import os

from keras.models import load\_model

import numpy as np

from pygame import mixer

import time

mixer.init()

sound = mixer.Sound('alarm.wav')

face = cv2.CascadeClassifier('C:\Python\Python36\Lib\site-packages\cv2\data\haarcascade\_frontalface\_alt.xml')

leye = cv2.CascadeClassifier('C:\Python\Python36\Lib\site-packages\cv2\data\haarcascade\_lefteye\_2splits.xml')

reye = cv2.CascadeClassifier('C:\Python\Python36\Lib\site-packages\cv2\data\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,height-50) , (200,height) , (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,(255,255,255),1,cv2.LINE\_AA)

    # if(rpred[0]==1 or lpred[0]==1):

    else:

        score=score-1

        cv2.putText(frame,"Open",(10,height-20), font, 1,(255,255,255),1,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) & 0xFF == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()

**CHAPTER NO. 14**

**CONCLUSION**

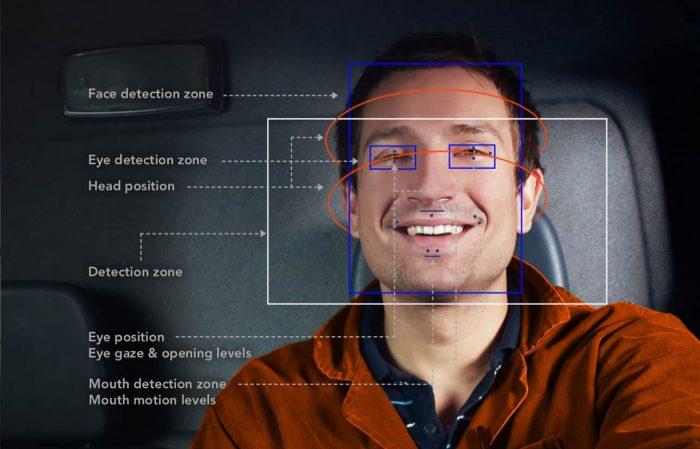
In this Python project, we have built a drowsy driver alert system that you can implement in numerous ways. We used OpenCV to detect faces and eyes using a haar cascade classifier and then we used a CNN model to predict the status. The driver abnormality monitoring system developed is capable of detecting drowsiness, drunken and reckless behaviours of drivers in a short time. The Drowsiness Detection System developed based on eye closure of the driver can differentiate normal eye blink and drowsiness and detect the drowsiness while driving. The proposed system can prevent accidents due to sleepiness while driving. The system works well even in case of drivers wearing spectacles and even under low light conditions if the camera delivers better output. Information about the head and eyes position is obtained through various self-developed image processing algorithms. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. processing judges the driver’s alertness level on the basis of continuous eye closures.

**Chapter 14**

**REFRENCES**

To make this project I have taken source from the following.

1. Google.com, <https://www.google.com/>
2. Youtube.com, <https://www.youtube.com/>
3. Github.com, <https://github.com/>
4. W3school, <https://www.w3schools.com/>

****