A REAL-TIME DRIVER DROWSINESS DETECTION USING EYE ASPECT RATIO ALGORITHM

A PROJECT REPORT

Submitted by

SUBHA SHRI AV	**611820205037**
KIRUTHIKA S	**611820205018**
KAVI PRIYA S	**611820205015**

In partial fulfillment for the award of the degree
Of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

P.S.V. COLLEGE OF ENGINEERING AND TECHNOLOGY
(An ISO 9001:2015 Certified Institution)

KRISHNAGIRI – 635 108
ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this Project titled **“A REAL-TIME DRIVER DROWSINESS DETECTION USING EYE ASPECT RATIO ALGORITHM”** is the Bonafide work of **SUBHA SHRI A V(611820205037), KIRUTHIKA S (611820205018), KAVI PRIYA S(611820205015**) Who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this any other candidate. Has been successfully complete and submitted in partial fulfilment for the award of B.Tech Information Technology of Anna University During the academic year 2020-2024.

SIGNATURE
Dr. M. SRINIVASAN, M.E., Ph.D.,
Head of The Department,

Department of Information Technology,
P.S.V College of Engineering and
Technology,
Krishnagiri - 635 108.

SIGNATURE
Mrs. G. DIVYA, M.Tech.
Supervisor,
Assistant Professor,
Department of Information Technology,
P.S.V College of Engineering and
Technology,
Krishnagiri - 635 108.

The project report submitted for the viva - voice held on..................................at P.S.V. College of Engineering and Technology, Krishnagiri.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I first offer my thanks to all those who are graciously given me the strength and good health during the course of the project at thus pleasing moment of having successfully completed my project, I wish to convey our sincere thanks and gratitude to our beloved chairman,**Dr.P.SELVAM,M.A.,B.Ed.,M.Phil.,Ph.D.,** P.S.V. Group of Institutions.

We also acknowledge our deep sense of gratitude to our secretary **Dr.S.VIVEK,M.A.,M.B.A(U.K).,Ph.D.,** P.S.V. Group of Institutions.

I would like to express by sincere thanks to my beloved principal **Dr.P.LAWRENCE,M.E.,Ph.D.,** P.S.V. College of Engineering and Technology, Krishnagiri, for forwarding us to our project and offering adequate duration in completing my project.

I also express my sincere thanks to, Head of the Department,**Dr.M.SRINIVASAN,M.E.,Ph.D** Department of Information Technology, for providing all the facilities in the successful completion of my project.

I have great pleasure to express my sense of gratitude to my internal guide **Mrs.G.DIVYA,** Assistant Professor and Department of Information Technology, whose guidance and encouragement made this project an interesting educational experience.

Our thanks to librarians **Mr.M.JANAGAN,B.E.,M.E.,BLIS,MLIS.,** and **Mrs.G.SANGEETHA,B.A.,C.L.I.S.C.** for encouraging us.

Last but not least the whole thing will be incomplete if we don’t acknowledge our PARENTS who are everything for us and we are very proud of having my beloved parents.

SUBHA SHRI A V
KIRUTHIKA S
KAVI PRIYA S

ABSTRACT

Every year thousands of people in India lose their lives due to traffic accidents. The role of human factor plays a key role in the accidents. In general, the driver fatigue alone accounts for around 25 percent of the road accidents and up to 60 percent of road accidents result in death or serious injury. A main cause of fatigue is sleeplessness or insomnia. So a drivers' drowsiness state is a major factor in severe road accidents that claims thousands of lives every year. In the recent years, use of intelligent algorithms in vehicles has developed considerably. This project proposes a technique to check the driver fatigue by detecting the facial expressions of the drivers, using face detection technique. It implements a real-time driving drowsiness detection algorithm that considers the individual differences of driver. A Haar cascade classifier is constructed to detect the face region, which avoids the problem of poor accuracy caused by artificial feature extraction. Based on the Dlib toolkit, the landmarks of frontal driver facial emotion in a frame are found. According to the eyes landmarks, a new parameter, called Eyes Aspect Ratio, is introduced to evaluate the drowsiness of driver in the current frame. A unique fatigue state classifier, based on Support Vector Machines, was trained which taking the Eyes Aspect Ratio as input. The trained classifier is application to monitor the state of driver monitor. Because the fatigue driving state is gradually produced, a variable which calculated by number of drowsy frames per unit time is introduced to assess the drowsiness of driver. Through comparative experiments, we demonstrate this algorithm outperforms current driving drowsiness detection approaches in both accuracy and speed. In simulated driving applications, the proposed algorithm detects the drowsy state of driver quickly from 640∗480 resolution images at over 20fps and 98.80% accuracy. This system also generates voice alert to make the driver awake when he/she is detected as drowsy driver and send an mail alert message to authorized person. This system performs well with adequate precision and a good recall.

Keywords: Eyes Aspect Ratio, Dlib toolkit, EAR algorithm, OpenCV, Pyttsx3.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

QT WIDGETS	-	Classes for creating classic desktop -style UIs
QTSQL	-	Classes for database integration using SQL
QTSVG	-	Classes for displaying the contents of SVG files
QTSCRIPT	-	Classes for evaluating QT scripts
QT DESIGNER	-	Classes for extending QT designer
QT XML	-	Classes for handling XML
QT MULTIMEDIA	-	Classes for low -level multimedia programming
QT NETWORK	-	Classes for network programming
QT WEBKIT	-	Classes for rendering and editing HTML
QTCORE	-	Core non-GUI classes used by other modules
DDRS	-	Double data rate version 3
DRAM	-	Dynamic random access memory
XML	-	Extensible markup language
GUI	-	Graphical user interface
QTGUI	-	Graphocal user interface components
HDD	-	Hard disk drive
HOG	-	Histogram of gradients
LGA	-	Land grid array
LBP	-	Local binary pattern
QTOPEN GL	-	OpenGL support classes
PERCLOS	-	Percentage of eyelid closure
RAM	-	Random access memory
SSD	-	Solid state drive
SRAM	-	Static random access memory
SVM	-	Support vector machine

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Road accident is global tragedy with over-rising trend. India suffers from the highest number of deaths –around 1, 05,000 in absolute terms annually-due to road accidents in the world owing to the poor infrastructure and dangerous driving habit. The increasing number of traffic accidents due to a diminished driver's vigilance level has become a serious problem for the society. Statistics show that 20% of all the traffic accidents and up to one-quarter of fatal and serious accidents are due to drivers with diminished vigilance level. Most motor vehicle crashes are caused by driver error (e.g. speeding) or poor operating practices including lack of seat belt use, distractions, fatigue, rash driving, and alcohol or drug use while driving. Most of the road users are quite well aware of the general rules and safety measures while using roads but it is only the laxity on part of road users, which cause accidents and crashes. Of all these issues, the most important issue to be considered is the driver's concentration as well as driver's inattention or distraction. Therefore, human behavior factors affecting the driver's performance are important and should be considered while implementing the reliable systems to assure safe driving. To address this problem, this project proposed an algorithm to implement and thereby recreate the face detection algorithm presented by Viola-Jones. Viola Jones algorithm helps in achieving high detection rates. It also processes the images rapidly. It forms the basis of most of the real time systems as it works only on the present single grey scale image This algorithm is capable of functioning in an unconstrained environment meaning that it can detect all visible faces in any conceivable image. The ideal goal of any face detection algorithm is to perform on par with a human inspecting the same image, but this algorithm will constrain itself to only match the figures posted by Viola Jones. In order to guarantee optimum performance of the developed algorithm the vast majority of images used for training, evaluation and testing are either found on the internet or taken from private collections.

1.2 OBJECTIVE

- To serve intelligent transportation system that can ensure driver safety and reduce the losses caused by drowsy driving.

- To avoid human losses due to accidents that happens due to the carelessness of the driver.

- To implement a better drowsiness detection system to detect the drowsiness of the driver that can outperform all the pre-existing systems.

CHAPTER 2

LITERATURE SURVEY

1. Ruian Liu,et.a; Design of face detection and tracking system,"

According to survey, driver fatigue results in over 50% of the road accidents each year. Using technology to detect driver fatigue/drowsiness is an interesting challenge that would help in preventing accidents. In the past various efforts have been reported in the literature on approaches for drowsiness detection of automobile driver.

2. Picot, A. et.al., "On-Line Detection of Drowsiness Using Brain and Visual Information"

If the eyes are detected to be open, no action is taken. But, if eyes are detected to be closed continuously for two seconds according to this work, that is a particular number of frames depending on the frame rate, then it means that the automobile driver is feeling drowsy and a sound alarm is triggered. However, if the closed states of the eyes are not continuous, then it is declared as a blink.

3. Y. Zhang and C. Hua, ‘‘Driver fatigue recognition based on facial expression analysis using local binary patterns,’’

Zhang and Hua used fatigue facial expression reorganization based on Local Binary Pattern (LBP) features and Support Vector Machines (SVM) to estimate the driver fatigue, but the complexity of this algorithm is bigger than our algorithm.

4. J. Jimenez-Pinto and M. Torres-Torriti, "Driver alert state and fatigue detection by salient points analysis"

Different face detection techniques are characterized by different face detection rates. Analysis of a number of the most popular techniques in this work led to the conclusion that the combination of Viola-Jones technique and PERCLOS method could yield the eye detection rate reaching up to 99% and blinking detection rate up to 97.8%, thus being superior compare to other approaches. This combination has been employed in the reported driver drowsiness detection system.

5. **RituUpadhayay, RakeshKumarYadav “Kernel Principle Component Analysis in Face Recognition System: A Survey”**

Yiaxiao Yun et. al. here in this paper the authors talk about both inner as well as outer side of the car to be traced. Video recording sensors are used and based on the frames received the status of the driver is drawn. The main distinguishing feature of this method is that the authors also focus on the outside of the car to track the vehicles on both side of the car by using sensors and based on the distance the driver is alerted about the situation. Face analysis and detection here works in a simple manner by reading the frames recorded by the camera and recognizing the face. After face is recognized the other parts of the face are recognized based on the facial geometry, i.e. relative position of various organs on the face.

6. **J. Ren, N. Kehtarnavaz and L. Estevez, "Real-time optimization of Viola-Jones face detection for mobile platforms"**

A cascade of classifiers is constructed aiming to achieve increased detection performance while radically reducing computation time. The key insight here is that smaller, more efficient, boosted classifiers can be assembled in such a way that to smaller, more efficient, boosted classifiers can be assembled in such a way that to reject many of the negative sub-windows while detecting almost all positive ones.

7. **G. Bradski and A. Kaehler, “Learning OpenCV: Computer Vision with the OpenCV Library”**

After the classifier function is developed, AdaBoost algorithm is used for selecting the required features and training the data set. This algorithm basically is used for enhancing the performance of the classifier function. AdaBoost is an effective procedure for searching out a small number of good “features” which nevertheless have significant variety. This algorithm is used for selecting the features like eyes or mouth region. After selecting the features of interest, the data sets are trained.

8. PrantoHidayaRusmin, Andrew B. Osmond, AriefSyaichu-Rohmann "Design and Implementation of Driver's Drowsiness Detection System on Digitalized Driver System"

Shinfeng D. Lin presents a sleepy eye's recognition for drowsiness detection without the training stage. In the beginning, an Adaboost classifier with Haar - like features is utilized to find out the face area. Then the eyes region is located by ASM. Finally, the binary pattern and edge detection are adopted to recognize the eye's state. Experimental results prove that the proposed method could accurately.

9. L. Tijerina, W. W. Wierwille, M. J. Goodman, S. Johnston, D. Stoltzfus and M. Gleckler, "A preliminary assessment of algorithms for drowsy and inattentive driver detection on the road"

Percentage of Eyelid Closure (PERCLOS) is the most popular method for drowsiness detection. It mathematically defines the proportion of time when the eyes are 80 percent to 100 percent closed. It monitors the slow closure of an eye lid rather than the fast blinking state of the eyes.

10. Paul Viola, Michael J. Jones, Robust, "Real-time Face Detection"

When the eyes are closed for more than three frames then it is deducible that the driver is feeling drowsy and similarly if the mouth is open for more than three frames, then it is deducible that the driver is yawning. Hence these cases are detected is detected and an alarm sounded.

11. P. Viola et M. Jones, "Rapid object detection using boosted cascade of simple features"

Viola Jones Algorithm forms the basis of this robust system. Viola Jones algorithm helps in achieving high detection rates. It also processes the images rapidly.

CHAPTER 3

SYSTEM DESIGN

3.1 EXISTING SYSTEM

- In the existing work, pre-existing features for facial landmark detection is implemented to identify the state of drowsiness and fatigue.

- 68- facial landmark predefined landmark helps in shape prediction to clearly identify the various regions of the face like eye brows, eye, mouth region etc.

- Various change in parameters of these distinguished points reports various expression of the person.

- The existing system uses Viola Jones algorithm for identifying the drowsiness.

3.1.1 Disadvantages

- The algorithm used here, rely on the detected facial landmarks which make the system less sensitive to landmark misalignment.

- Cannot detect exact drowsiness.

3.2 PROPOSED SYSTEM

- The proposed system that is being developed is focused on image recognition based on drowsiness.
- Drowsiness can be detected by scrutinizing the status of eye.
- This algorithm which will automatically extract eye area and scrutinize this eye part for drowsiness detection based on status of eye.
- The proposed framework is done on the basis of real time face.
- Driver is given a voice alert when drowsiness is detected and also sends mail alert message if drowsiness is detected.

3.2.1 Advantages

- Besides detecting the drowsiness of the driver, the system also alerts the driver to make him awake at the right time.
- It achieves high precision in detection of drowsiness.

CHAPTER 4

SYSTEM FUNCTION

4.1 ARCHITECTURE DESIGN

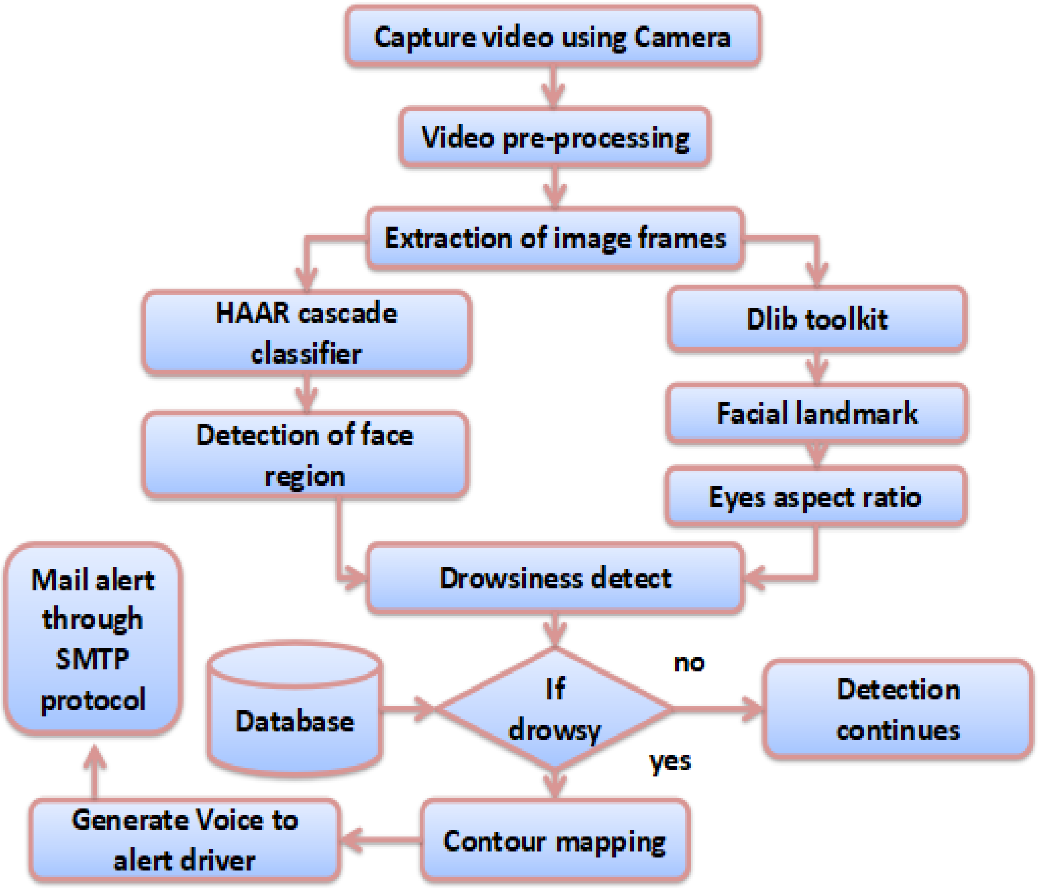


Figure 4.1Architecture diagram for proposed system

4.2 SYSTEM MODULES

There are five modules in this system:

- Real-time camera monitoring module.
- Face region detection module.

- Eyes part scrutinization module.
- Drowsiness detection module.
- Voice alert and mail module.

4.2.1 Real-time Camera Monitoring Module:

In this module, the real-time camera is used to get input image, as the camera will capture live video streaming and convert the video into image frames by extracting the images from video.

4.2.2 Face Region Detection Module:

This module uses HAAR cascade classifier to detect and locate face region in the images.

4.2.3 Eyes Part Scrutinization Module

Here, the Dlib toolkit is used to extract prime facial features present in the face region of the image. Facial landmark algorithm will efficiently locate the region of interest i.e., eye region.

4.2.4 Drowsiness Detection Module

In this module, the drowsiness of the driver will be detected by determining the eyes aspect ratio. Eyes aspect ratio is calculated and by estimating the changes in that parameter’s value, we can determine the drowsy state of the driver.

4.2.5 Voice Alert Module

In this module, the driver will be alerted or warned by generating voice alert when the driver is confirmed or detected to be in drowsy state which may leads to highly dangerous situations. To avoid those serious consequences, the system alerts the driver to make him attentive right after the detection of drowsiness. In addition to the detection of driver drowsiness, it will also alert the driver to save more lives and if driver is in drowsiness, then an alert message will be sent to authorized person.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 HARDWARE SPECIFICATION

Processor	:	INTEL I5 (7th generation)
RAM	:	4 GB RAM
Hard disk	:	1TB
Monitor	:	20’ color monitor

5.2 SOFTWARE SPECIFICATION

Front end	: GUI
Back end	: python
Software tool used	: PyCharm
Platform	: Windows 8

5.3 HAAR CASCADE CLASSIFIER

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

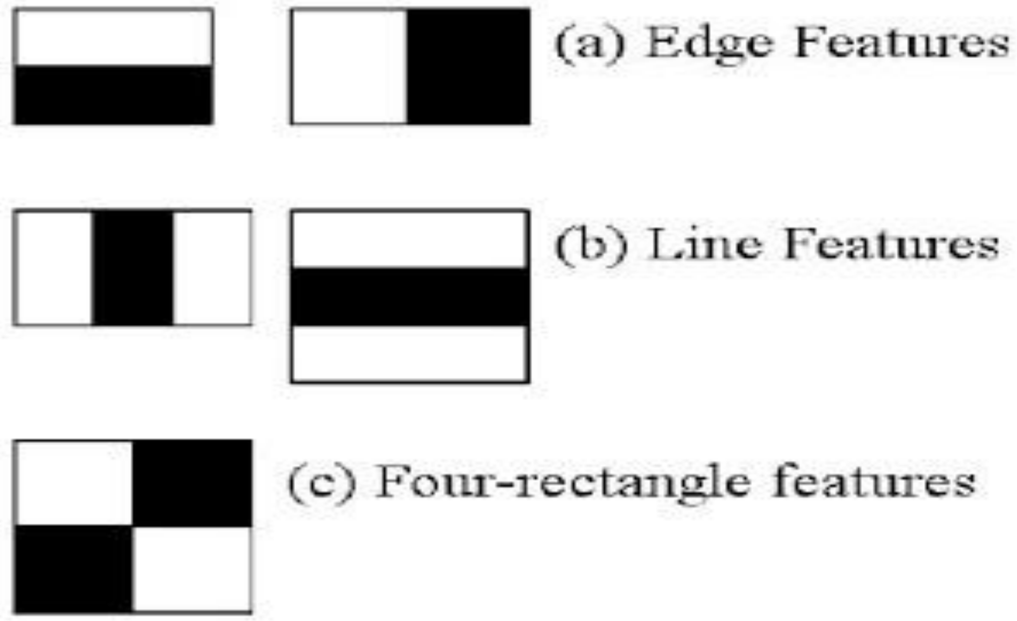


Figure 5.1 HAAR cascade classifier

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So, selecting the best features out of 160000+ features is highly important. It is achieved by Adaboost.

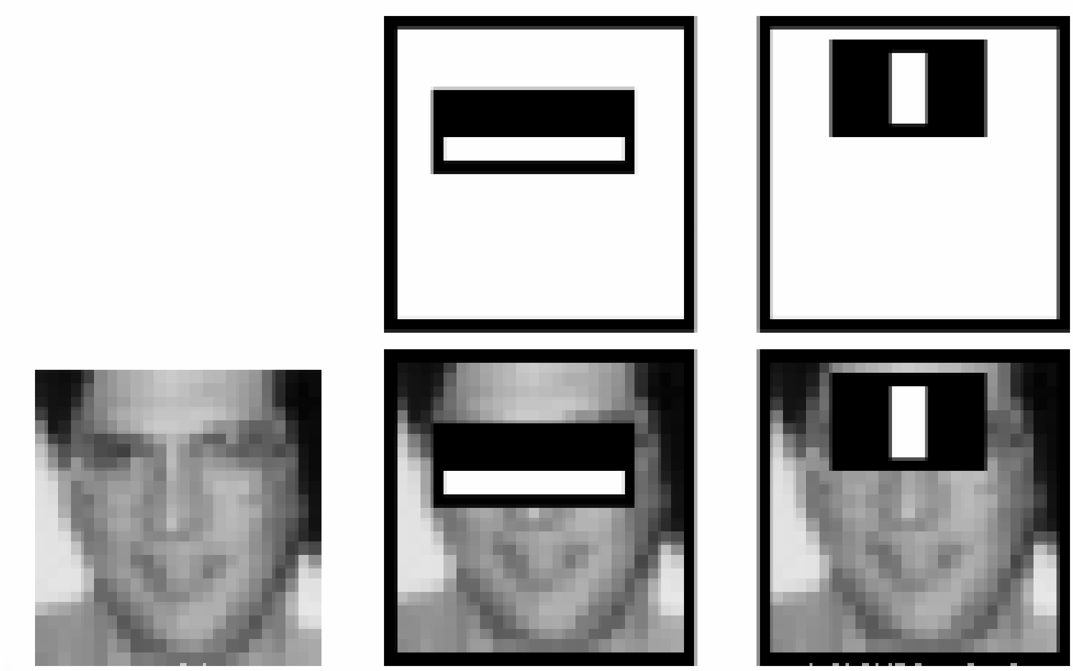


Figure 5.2 Detection process

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features is found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. A reduction from 160000+ features to 6000 features is a big gain.

In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot, and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

For this they introduced the concept of Cascade of Classifiers. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

The authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. (The two features in the above image are actually obtained as the best two features from Adaboost). According to the authors, on average 10 features out of 6000+ are evaluated per sub-window.

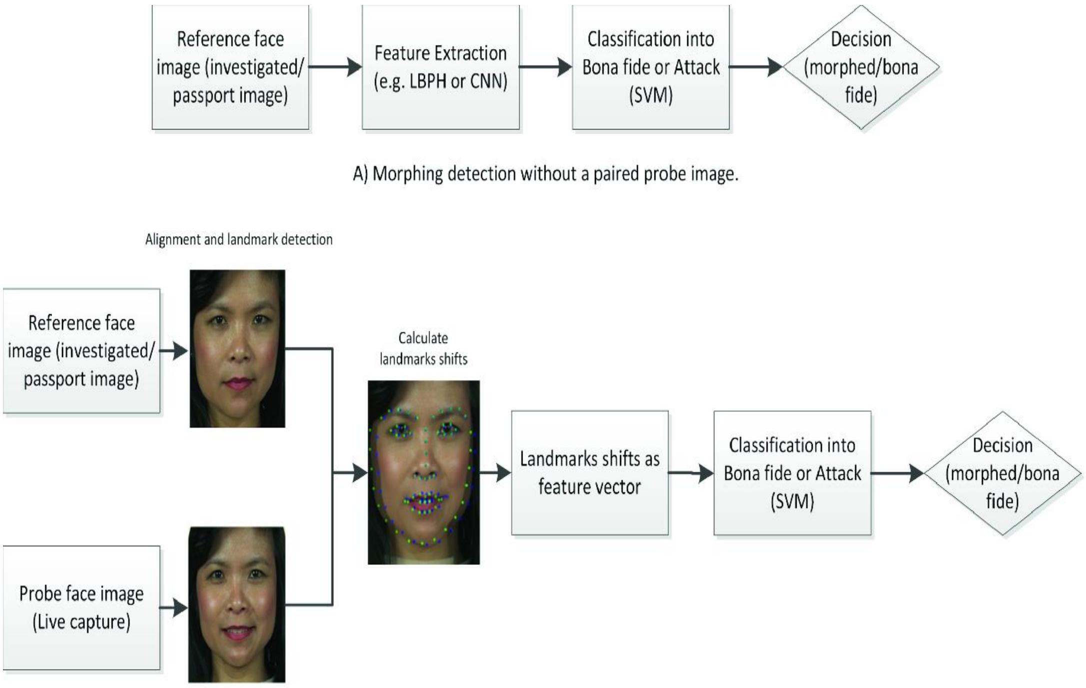
5.4 FACIAL LANDMARK ALGORITHM

Facial landmarks is a technique which can be applied to applications like face alignment, head pose estimation, face swapping, blink detection, drowsiness detection, etc. In this context of facial landmarks, our vital aim is to detect facial structures on the person’s face using a method called shape prediction.
Facial Landmarks Detection has 2 steps:

1. To detect the key facial structures on the person’s face.

2. It involves localizing the face in the image.

We can do Face detection in a number of ways. We can use OpenCV’s built-in Haar Cascade XML files or even TensorFlow. Over here especially, we need to apply a HOG (Histogram of Gradients) object detector specifically for the task of face detection. We can also do it using Deep Learning-based algorithms which are built for face localization. Also, the algorithm will be used for the detection of the faces in the image.


Reference face image (investigated/ passport image)
Feature Extraction (e.g. LBPH or CNN)
Classification into Bona fide or Attack (SVM)
Decision (morphed/bona fide)
Alignment and landmark detection
Reference face image (investigated/ passport image)
Calculate landmarks shifts
Landmarks shifts as feature vector
Classification into Bona fide or Attack (SVM)
Decision (morphed/bona fide)
Probe face image (Live capture)

A) Morphing detection without a paired probe image.

Figure 5.3 Face detection using facial landmark

5.5 DLIB’S FACIAL LANDMARK DETECTOR

1. The pre-trained facial landmark detector inside the dlib library is used to estimate the location of *68 (x, y)-coordinates* that map to facial structures on the face.

2. The indexes of the 68 coordinates can be visualized on the image below:

3. These annotations are part of the 68 point iBUG 300-W dataset which the dlib facial landmark predictor was trained on.

4. It’s important to note that other flavors of facial landmark detectors exist, including the 194 point model that can be trained on the HELEN dataset.

5. Regardless of which dataset is used, the same dlib framework can be leveraged to train a shape predictor on the input training data — this is useful if we would like to train facial landmark detectors or custom shape predictors of our own.

6. We can also use these facial landmarks to extract specific regions of the face, apply face alignment, and even build a blink detection system.

7. Dlib-ml is a cross platform open source software library written in the C++ programming language. Its design is heavily influenced by ideas from design by contract and component-based software engineering. This means it is first and foremost a collection of independent software components, each accompanied by extensive documentation and thorough debugging modes. Moreover, the library is intended to be useful in both research and real world commercial projects and has been carefully designed to make it easy to integrate into a user's C++ application. There are a number of well known machine learning libraries. However, many of these libraries focus on providing a good environment for doing research using languages other than C++. Two examples of this kind of project are the Shogun (Sonnenburg et al., 2006) and Torch (Collobert and Bengio, 2001) toolkits which, while they are implemented in C++, are not focused on providing support for developing machine learning software in that language. Instead they are primarily intended to be used with languages like R, Python, Matlab, or Lua. Then there are toolkits such as Shark (Igel et al., 2008) and dlib-ml which are explicitly targeted at users who wish to develop software in C++. Given these considerations, dlib-ml attempts to help fill some of the gaps in tool support not already filled by libraries such as Shark. It is hoped that these efforts will prove useful for researchers and engineers who wish to develop machine learning software in this language.

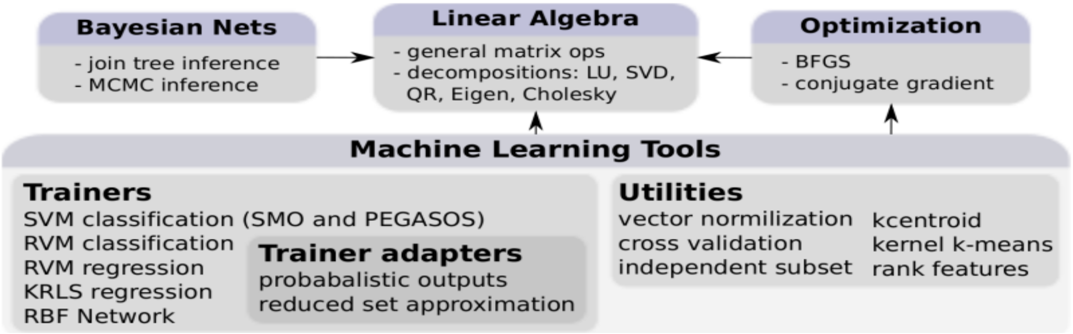


Figure 5.4 Elements of dlib-ml. Arrows show dependencies between components

5.5.1 Elements of the Library

The library is composed of the four distinct components shown in Figure 1. The linear algebra component provides a set of core functionality while the other three implement various useful tools. This paper addresses the two main components, linear algebra and machine learning tools.

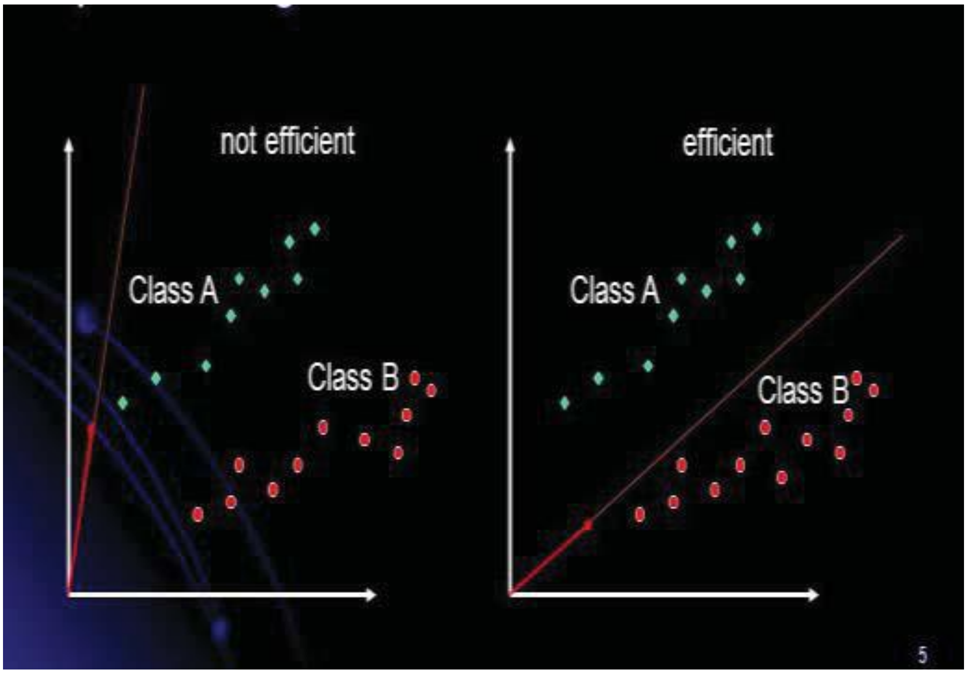


Figure 5.5 PCA seeks directions that are efficient for representing the data.

5.5.2 Linear Algebra

The design of the linear algebra component of the library is based on the template expression techniques. The dlib-ml implementation extends this original design in a number of ways. Most notably, the library can use the BLAS when available, meaning that the performance of code developed using dlib-ml can gain the speed of highly optimized libraries such as ATLAS or the Intel MKL while still using a very simple syntax. Consider the following example involving matrix multiplies, transposes, and scalar multiplications: (1) result = 3*trans(A*B + trans(A)*2*B); (2) result = 3*trans(B)*trans(A) + 6*trans(B)*A; The result of expression (1) could be computed using only two calls to the matrix multiply routine in BLAS but first it is necessary to reorder the terms into form (2) to fit the form expected by the BLAS routines. Performing these transformations by hand is tedious and error prone. Dlib-ml automatically performs these transformations on all expressions and invokes the appropriate BLAS calls. This enables the user to write equations in the form most intuitive to them and leave these details of software optimization to the library.

5.5.3 Machine Learning Tools

A major design goal of this portion of the library is to provide a highly modular and simple architecture for dealing with kernel algorithms. In particular, each algorithm is parameterized to allow a user to supply either one of the predefined dlib-ml kernels, or a new user defined kernel. Moreover, the implementations of the algorithms are totally separated from the data on which they operate. 1756 DLIB-ML:

5.5.4 A Machine Learning Toolkit

This makes the dlib-ml implementation generic enough to operate on any kind of data, be it column vectors, images, or some other form of structured data. All that is necessary is an appropriate kernel. This is a feature unique to dlib-ml. Many libraries allow arbitrary precomputed kernels and some even allow user defined kernels but have interfaces which restrict them to operating on column vectors. However, none allow the flexibility to operate directly on arbitrary objects, making it much easier to apply custom kernels in the case where the kernels operate on objects other than fixed length vectors. The library provides implementations of popular algorithms such as RBF networks and support vector machines for classification. This generic trainer interface, along with the contract programming approach, makes the library easily extensible by other developers. Another good example of a generic kernel algorithm provided by the library is the kernel RLS technique.

5.6 DBMS (DATA BASE MANAGEMENT SYSTEM)

DBMS stands for **D**ata**b**ase **M**anagement **S**ystem. We can break it like this DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can **define DBMS** like this: DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

5.6.1 What is the need of DBMS

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: **Storage of data** and **retrieval of data**.

5.6.2 Storage:

According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage. Let's take a layman example to understand this:In a banking system, suppose a customer is having two accounts, one is saving account and another is salary account. Let's say bank stores saving account data at one place (these places are called tables we will learn them later) and salary account data at another place, in that case if the customer information such as customer name, address etc. are stored at both places then this is just a wastage of storage (redundancy/ duplication of data), to organize the data in a better way the information should be stored at one place and both the accounts should be linked to that information somehow. The same thing we achieve in DBMS.

5.6.3 **Fast Retrieval of data**:

Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible.

5.6.4 Purpose of Database Systems

The main purpose of database systems is to manage the data. Consider a university that keeps the data of students, teachers, courses, books etc. To manage this data, we need to store this data somewhere where we can add new data, delete unused data, update outdated data, retrieve data, to perform these operations on data we need a Database management system that allows us to store the data in such a way so that all these operations can be performed on the data efficiently.

5.7 SMTP PROTOCOL

Email is emerging as one of the most valuable services on the internet today. Most of the internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver’s side.

5.7.1 SMTP Fundamentals

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode.

5.7.2 SMTP Protocol

The SMTP model is of two types:

1. End-to- end method

2. Store-and- forward method

The end to end model is used to communicate between different organizations whereas the store and forward method are used within an organization. A SMTP client who wants to send the mail will contact the destination’s host SMTP directly in order to send the mail to the destination. The SMTP server will keep the mail to itself until it is successfully copied to the receiver’s SMTP.

The client SMTP is the one which initiates the session let us call it as the client- SMTP and the server SMTP is the one which responds to the session request and let us call it as receiver-SMTP. The client- SMTP will start the session and the receiver-SMTP will respond to the request.

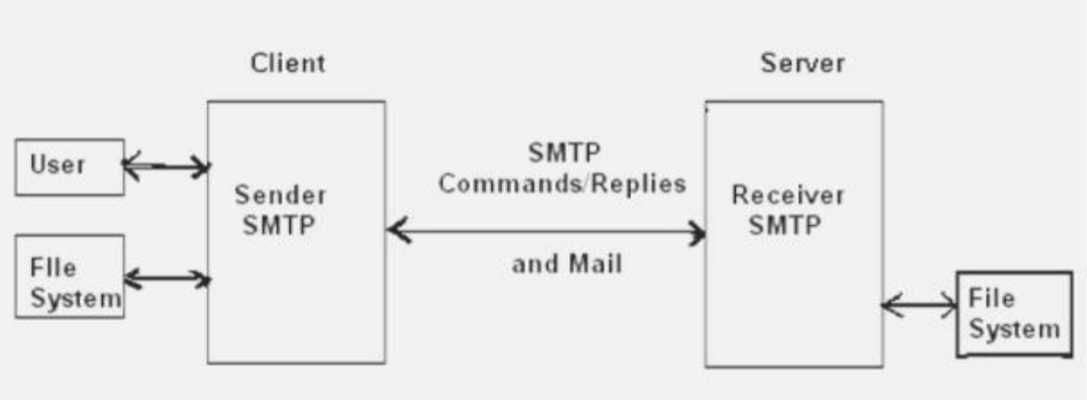


Figure 5.6 SMTP protocol

5.8 EYE ASPECT RATIO(EAR) AND DROWSINESS DETECTOR USING DLIB

In this article, I will show you how to determine facial landmarks using the dlib library, how to calculate EAR(Eye Aspect Ratio), and use the concept of EAR to detect drowsiness.

Before you begin with the code part of this article, you would have to install dlib library in python. There are some prerequisites for installing dlib library and I would recommend you to check this article.

5.8.1 FacialLandmarks that Dlib Detects

The dlib library can be used to detect a face in an image and then find 68 facial landmarks on the detected face.

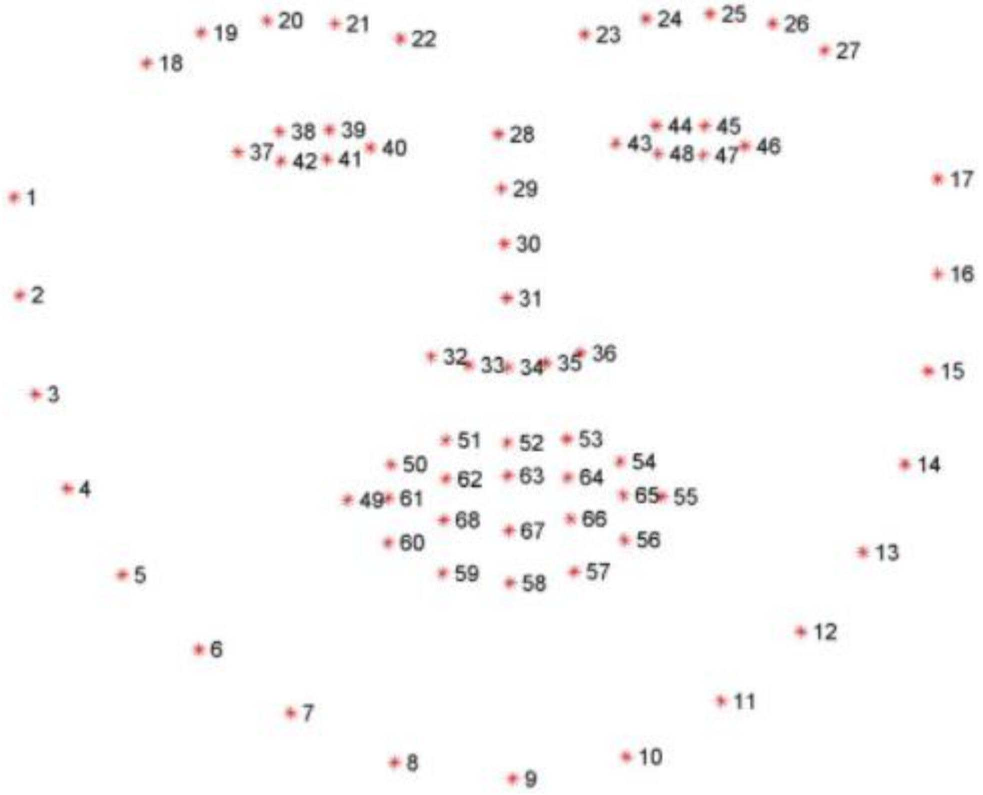


Figure 5.7 68 facial landmarks that dlib finds in a face

5.8.2 How to find Eye Aspect Ratio(EAR)

If you notice, each eye is represented using 6 landmarks points.

The EAR for a single eye is calculated using this formula:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 5.8 EAR Formula

The more the EAR, the more widely eye is open. We would decide a minmum EAR value and used this to decide if the eye is closed or not.

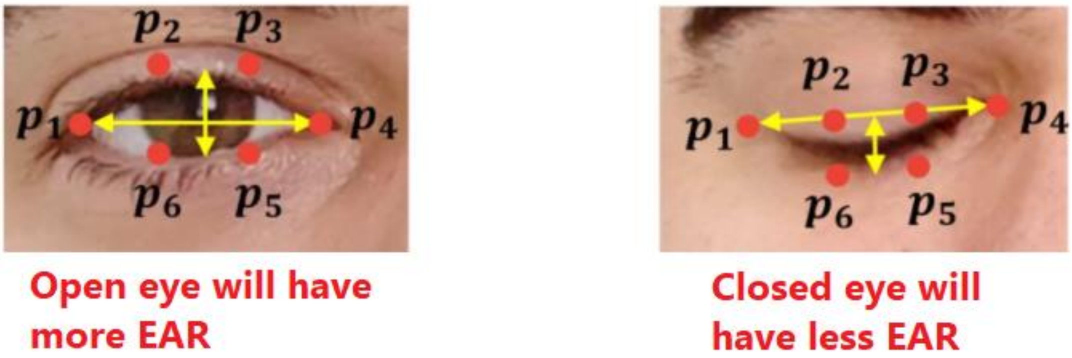
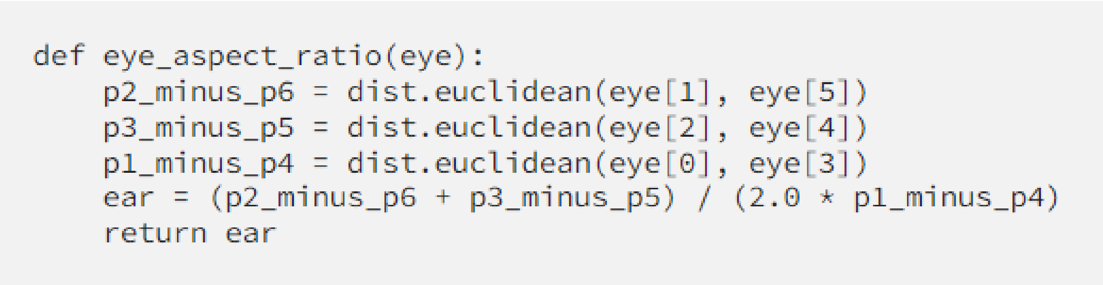


Figure 5.9 Eye is closed or not

Here is the utility function that would return the EAR for a single eye.



```
def eye_aspect_ratio(eye):
    p2_minus_p6 = dist.euclidean(eye[1], eye[5])
    p3_minus_p5 = dist.euclidean(eye[2], eye[4])
    p1_minus_p4 = dist.euclidean(eye[0], eye[3])
    ear = (p2_minus_p6 + p3_minus_p5) / (2.0 * p1_minus_p4)
    return ear
```

Figure 5.10 EAR for a single eye

CHAPTER 6

SYSTEM SOFTWARE

6.1 PYCHARM INTRODUCTION

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects −

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below −

Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

SQLAlchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

Code Coverage in Editor

You can run **.py** files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

Package Management

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

Local History

Local History is always keeping track of the changes in a way that complements like Git. Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.

Refactoring

Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

User Interface of PyCharm Editor

The user interface of PyCharm editor is shown in the screenshot given below. Observe that the editor includes various features to create a new project or import from an existing project.

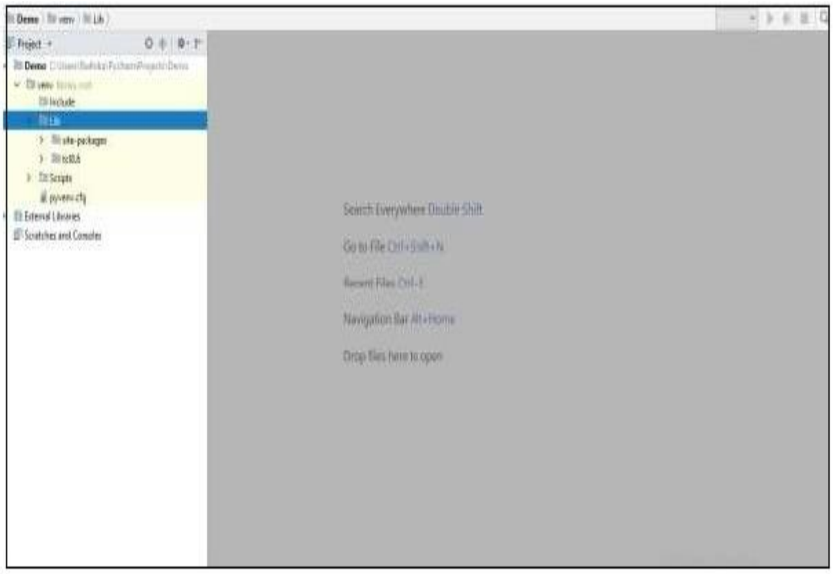


Figure 6.1 Editor window

6.2 PYCHAM INSTALLATION

From the screenshot shown above, you can see the newly created project Demo and the **site-packages** folder for package management along with various other folders.

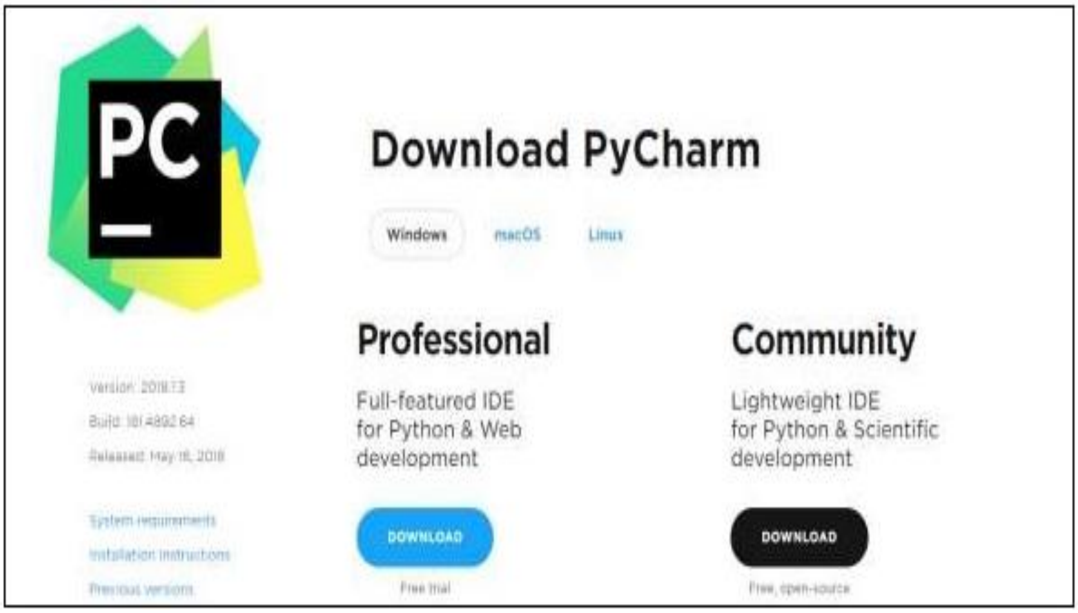
In this chapter, you will learn in detail about the installation process of PyCharm on your local computer.

Steps Involved

You will have to follow the steps given below to install PyCharm on your system. These steps show the installation procedure starting from downloading the PyCharm package from its official website to creating a new project.

Step 1

Download the required package or executable from the official website of you will observe two versions of package for Windows as shown in the screenshot given below – needed for installation. Note that we will continue with community package throughout this tutorial.



Download PyCharm

Windows macOS Linux

Professional

Full-featured IDE
for Python & Web
development

DOWNLOAD

Free trial

Community

Lightweight IDE
for Python & Scientific
development

DOWNLOAD

Free, open-source

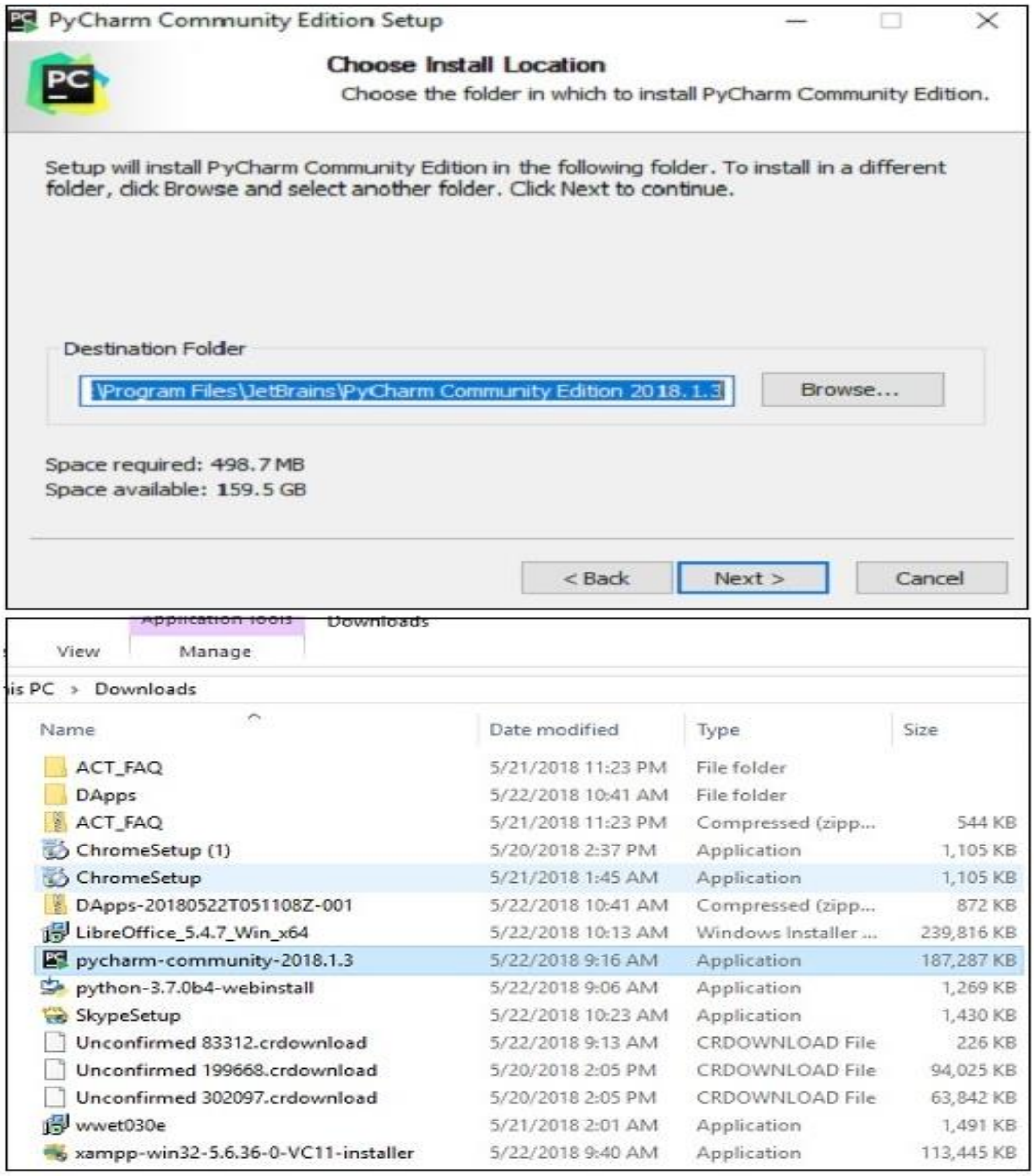
Version: 2018.1.3
Build: 181.4892.64
Released: May 16, 2018

System requirements
Installation instructions
Previous versions

Figure 6.2 PyCharm software

Step 2

Download the community package (executable file) onto your system and mention a destination folder as shown below −



Choose Install Location

Choose the folder in which to install PyCharm Community Edition.

Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: \Program Files\JetBrains\PyCharm Community Edition 2018.1.3

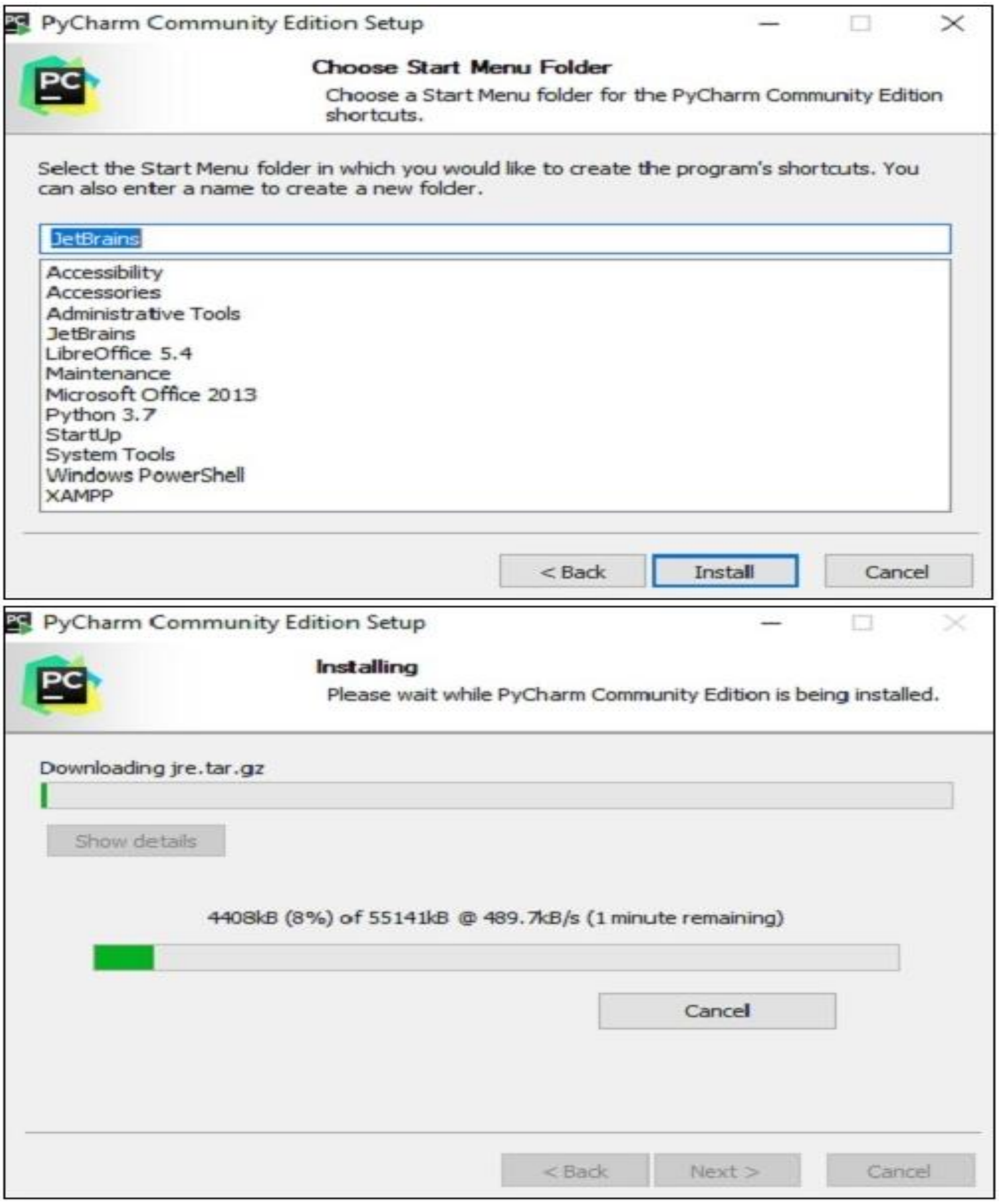
Space required: 498.7 MB
Space available: 159.5 GB

Name	Date modified	Type	Size
ACT_FAQ	5/21/2018 11:23 PM	File folder	
DApps	5/22/2018 10:41 AM	File folder	
ACT_FAQ	5/21/2018 11:23 PM	Compressed (zipp...	544 KB
ChromeSetup (1)	5/20/2018 2:37 PM	Application	1,105 KB
ChromeSetup	5/21/2018 1:45 AM	Application	1,105 KB
DApps-20180522T051108Z-001	5/22/2018 10:41 AM	Compressed (zipp...	872 KB
LibreOffice_5.4.7_Win_x64	5/22/2018 10:13 AM	Windows Installer ...	239,816 KB
pycharm-community-2018.1.3	5/22/2018 9:16 AM	Application	187,287 KB
python-3.7.0b4-webinstall	5/22/2018 9:06 AM	Application	1,269 KB
SkypeSetup	5/22/2018 10:23 AM	Application	1,430 KB
Unconfirmed 83312.crdownload	5/22/2018 9:13 AM	CRDOWNLOAD File	226 KB
Unconfirmed 199668.crdownload	5/20/2018 2:05 PM	CRDOWNLOAD File	94,025 KB
Unconfirmed 302097.crdownload	5/20/2018 2:05 PM	CRDOWNLOAD File	63,842 KB
wwet030e	5/21/2018 2:01 AM	Application	1,491 KB
xampp-win32-5.6.36-0-VC11-installer	5/22/2018 9:40 AM	Application	113,445 KB

Figure 6.3 PyCharm community edition setup

Step 3

Now, begin the installation procedure similar to any other software package.



PyCharm Community Edition Setup

Choose Start Menu Folder

Choose a Start Menu folder for the PyCharm Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessibility
Accessories
Administrative Tools
JetBrains
LibreOffice 5.4
Maintenance
Microsoft Office 2013
Python 3.7
StartUp
System Tools
Windows PowerShell
XAMPP

PyCharm Community Edition Setup

Installing

Please wait while PyCharm Community Edition is being installed.

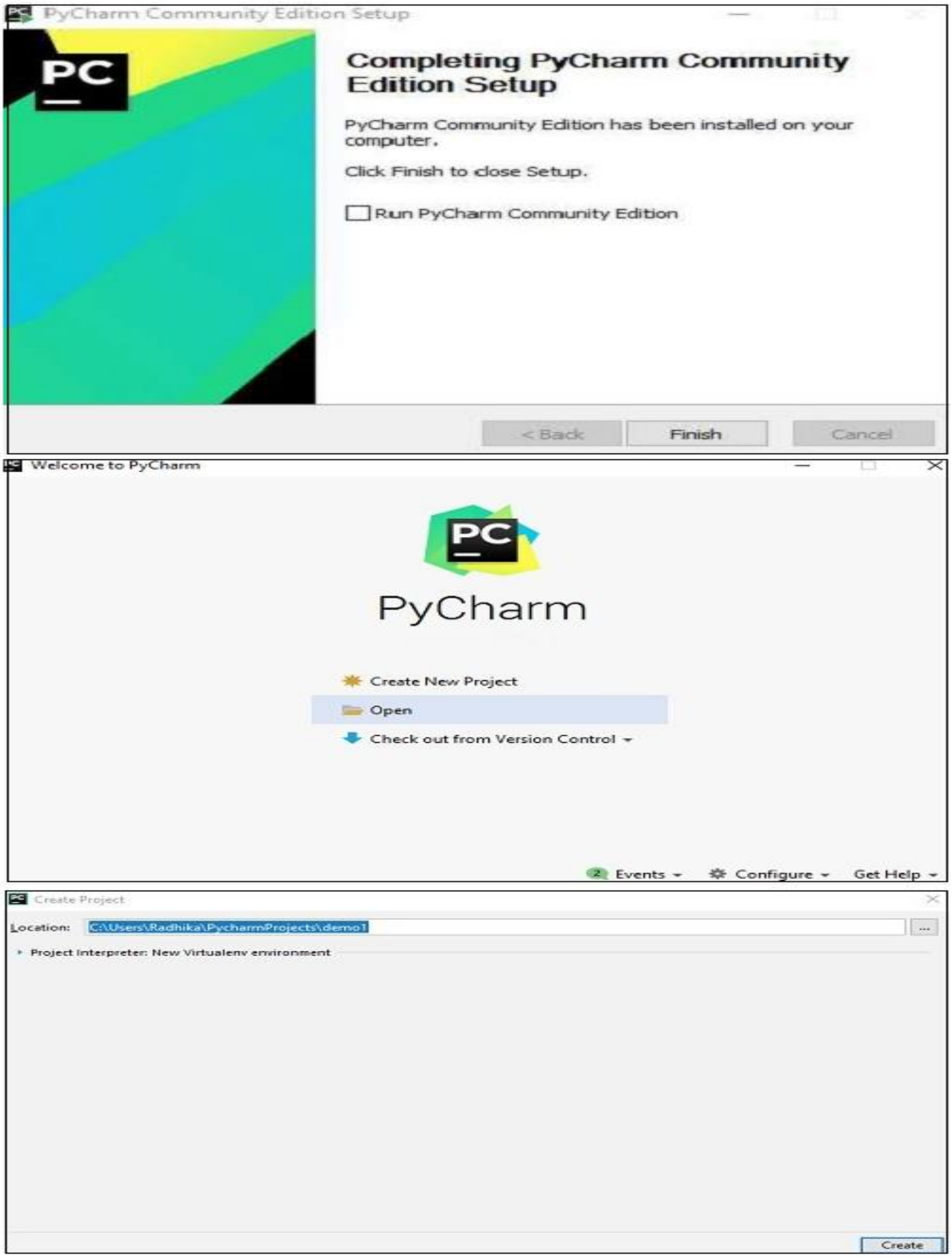
Downloading jre.tar.gz

4408kB (8%) of 55141kB @ 489.7kB/s (1 minute remaining)

Figure 6.4 PyCharm community edition setup

Step 4

Once the installation is successful, PyCharm asks you to import settings of the existing package if any.



Completing PyCharm Community Edition Setup

PyCharm Community Edition has been installed on your computer.

Click Finish to close Setup.

☐ Run PyCharm Community Edition

Location: C:\Users\Radhika\PycharmProjects\demo1

Project Interpreter: New Virtualenv environment

Figure 6.5 PyCharm installation

This chapter will discuss the basics of PyCharm and make you feel comfortable to begin working in PyCharm editor.

When you launch PyCharm for the first time, you can see a welcome screen with entry points to IDE such as −

- Creating or opening the project
- Checking out the project from version control
- Viewing the documentation
- Configuring the IDE

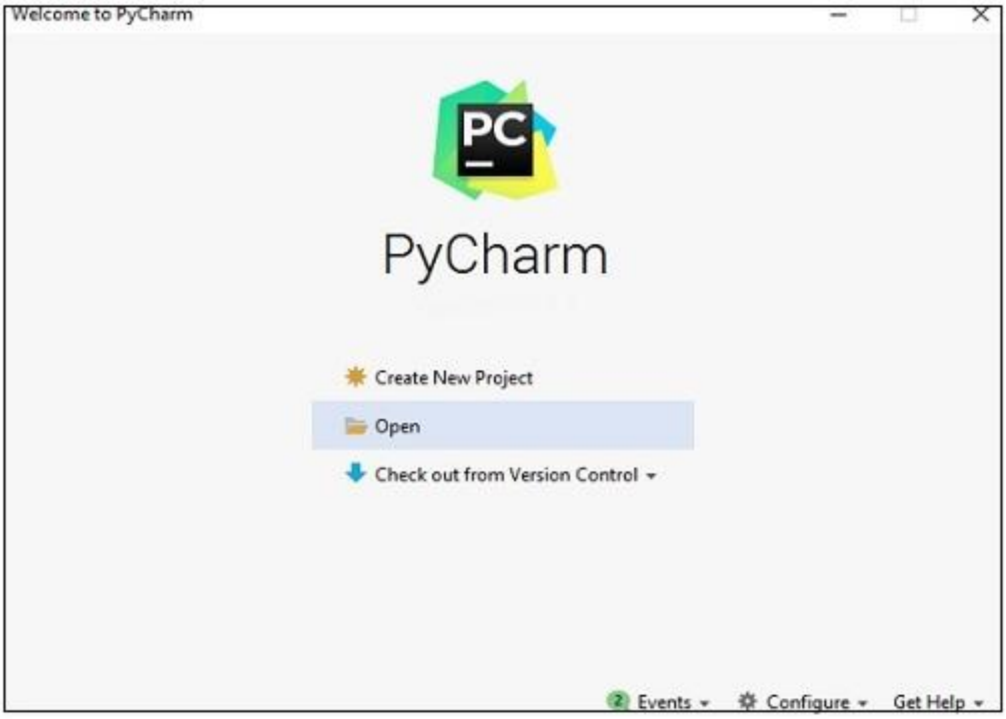


Figure 6.6 PyCharm software page

Recall that in the last chapter, we created a project named **demo1** and we will be referring to the same project throughout this tutorial. Now we will start creating new files in the same project to understand the basics of PyCharm Editor.

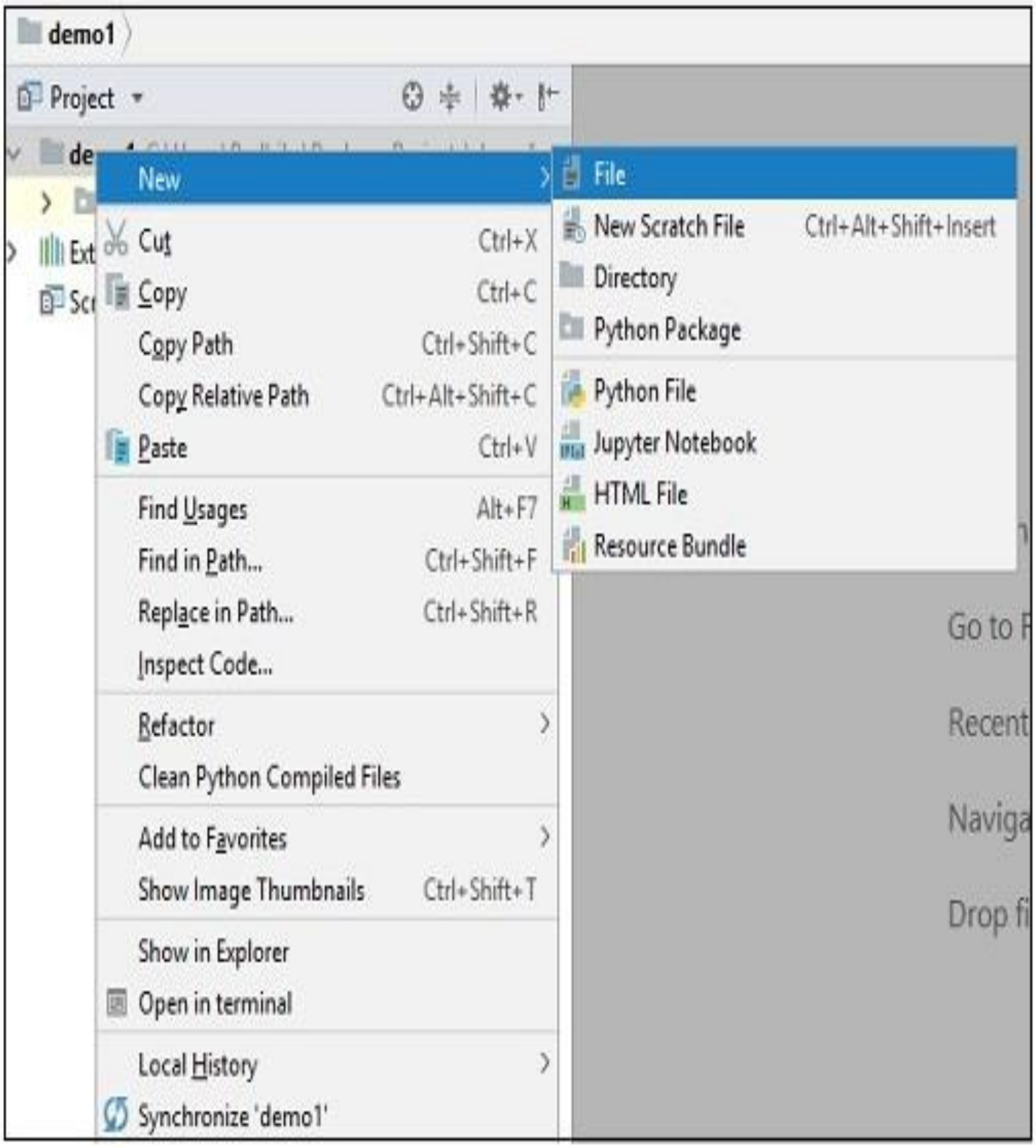
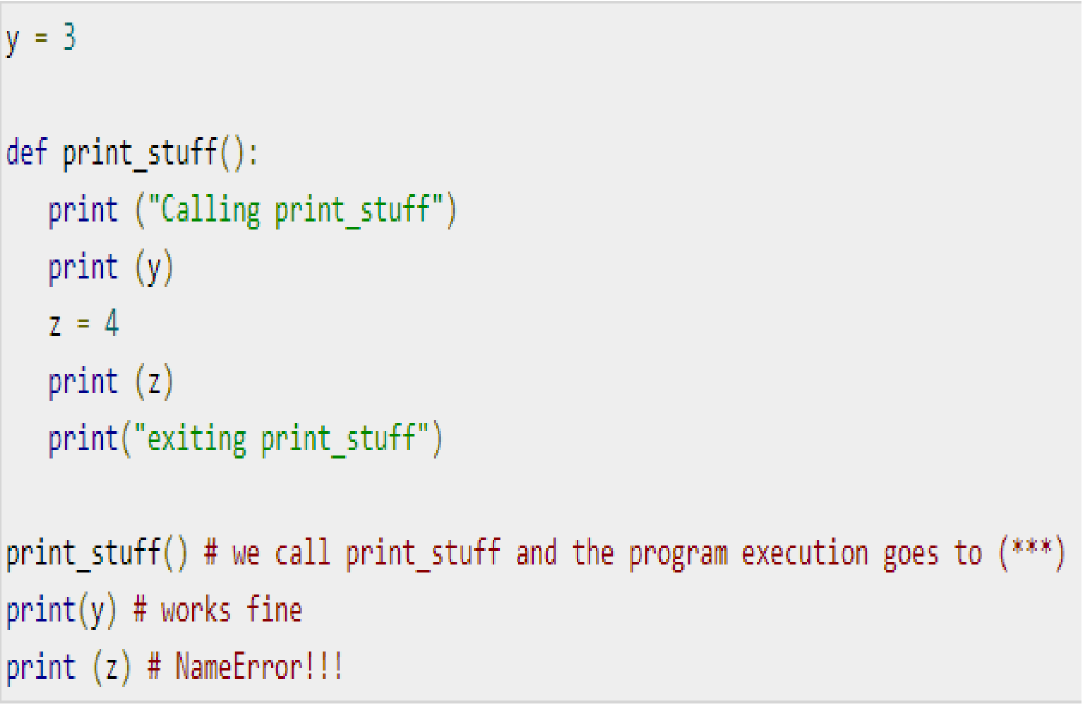


Figure 6.7 PyCharm software

The above snapshot describes the project overview of demo1 and the options to create a new file. Let us create a new file called **main.py**.

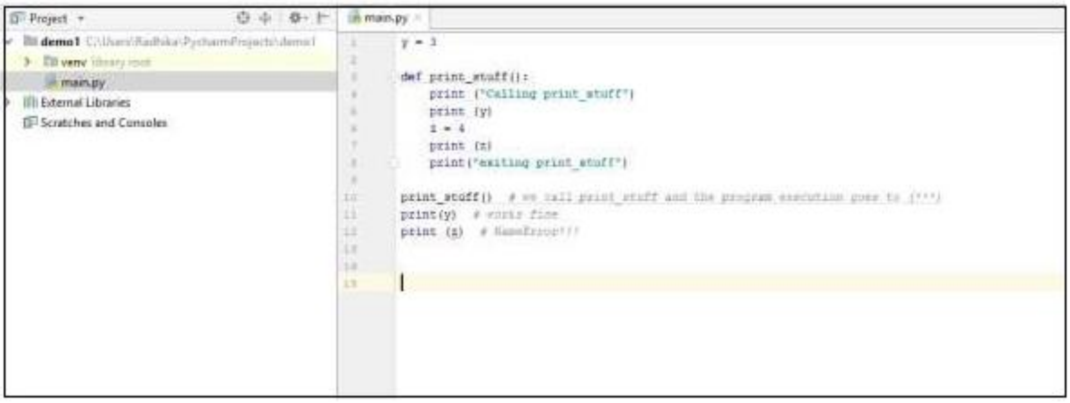
The code included in main.py is as follows



```
y = 3

def print_stuff():
   print ("Calling print_stuff")
   print (y)
   z = 4
   print (z)
   print("exiting print_stuff")

print_stuff() # we call print_stuff and the program execution goes to (***)
print(y) # works fine
print (z) # NameError!!!
```

Figure 6.8 programming page

The code created in the file **main.py** using PyCharm Editor is displayed as shown below –



```
y = 3

def print_stuff():
    print ("Calling print_stuff")
    print (y)
    z = 4
    print (z)
    print("exiting print_stuff")

print_stuff()  # we call print_stuff and the program execution goes to (***)
print(y)  # works fine
print (z)  # NameError!!!
```

Figure 6.9 programming page

This code can be run within IDE environment. The basic demonstration of running a program is discussed below

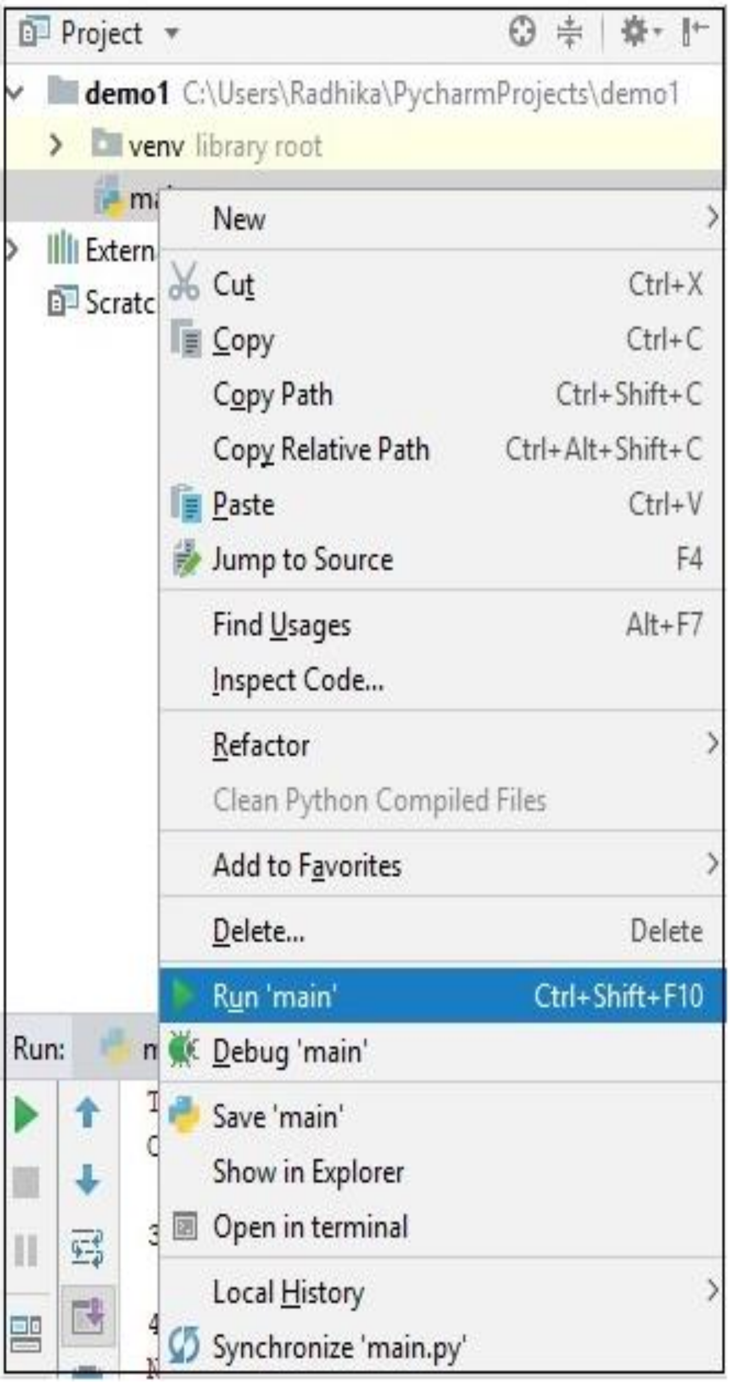
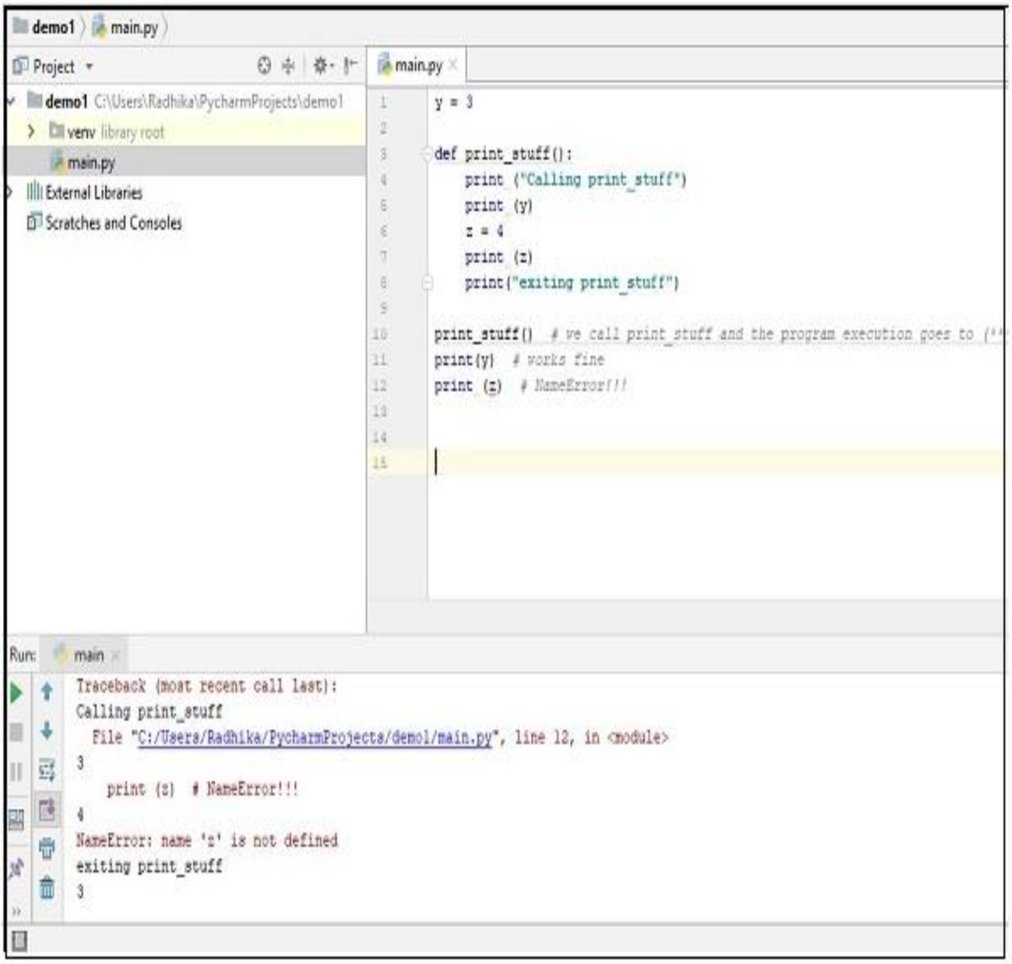


Figure 6.10 project window

Note that we have included some errors within the specified code such that console can execute the code and display output as the way it is intended to.



```
y = 3

def print_stuff():
    print ("Calling print_stuff")
    print (y)
    z = 4
    print (z)
    print("exiting print_stuff")

print_stuff()  # we call print_stuff and the program execution goes to (**
print(y)  # works fine
print (z)  # NameError!!!
```

```
Traceback (most recent call last):
Calling print_stuff
  File "C:/Users/Radhika/PycharmProjects/demo1/main.py", line 12, in <module>
3
    print (z)  # NameError!!!
4
NameError: name 'z' is not defined
exiting print_stuff
3
```

Figure 6.11 Programmingpage

6.3 LANGUAGE USED-PYTHON

Figure 6.12 Python

The system utilizes Python programming language with Google's Tensorflow Machine Learning Library to build and deploy the CNN.

The system utilizes Python programming language with Google's Tensorflow Machine Learning Library to build and deploy the CNN. The performance is analyzed based on real world scenarios tested on the neural network. The network architecture makes use of 9 convolutional and max-pooling layers, followed by 2 fully connected layers. The network is a concoction of classification and detection models.

6.3.1 Applications

- 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 development.

6.3.2 Features

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files. In this project, Python will be written in Thonny Software.

6.3.3 Python Syntax Compared to Other Programming Languages

- Python was designed for readability and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

6.4 PYQT5

PyQt5 is the latest version of a GUI widgets toolkit developed by Riverbank Computing. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. PyQt5 is a blend of Python programming language and the Qt library. This introductory tutorial will assist you in creating graphical applications with the help of PyQt. Our tutorial on earlier version − PyQt4 is available here.

PyQt API is a set of modules containing a large number of classes and functions. While **QtCore** module contains non-GUI functionality for working with file and directory etc., **QtGui** module contains all the graphical controls. In addition, there are modules for working with XML **(QtXml)**, SVG **(QtSvg)**, and SQL **(QtSql)**, etc.

A list of frequently used modules is given below −

- **QtCore** − Core non-GUI classes used by other modules
- **QtGui** − Graphical user interface components

- **QtMultimedia** − Classes for low-level multimedia programming
- **QtNetwork** − Classes for network programming
- **QtOpenGL** − OpenGL support classes
- **QtScript** − Classes for evaluating Qt Scripts
- **QtSql** − Classes for database integration using SQL
- **QtSvg** − Classes for displaying the contents of SVG files
- **QtWebKit** − Classes for rendering and editing HTML
- **QtXml** − Classes for handling XML
- **QtWidgets** − Classes for creating classic desktop-style UIs
- **QtDesigner** − Classes for extending Qt Designer

Supporting Environments

PyQt is compatible with all the popular operating systems including Windows, Linux, and Mac OS. It is dual licensed, available under GPL as well as commercial license. The latest stable version is PyQt5-5.13.2.

Windows

Wheels for 32-bit or 64-bit architecture are provided that are compatible with Python version 3.5 or later. The recommended way to install is using PIP utility −

```
pip3 install PyQt5
```

To install development tools such as Qt Designer to support PyQt5 wheels, following is the command −

```
pip3 install pyqt5-tools
```

You can also build PyQt5 on Linux/macOS from the source code www.riverbankcomputing.com/static/Downloads/PyQt5

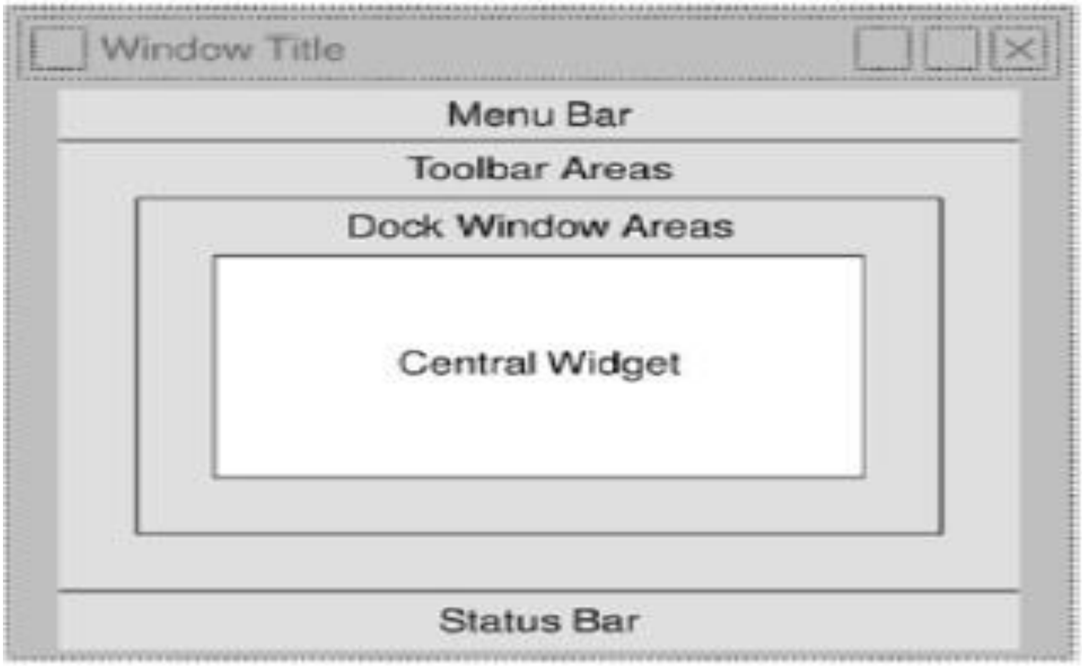


Figure 6.13 Qmainwindow

The PyQt installer comes with a GUI builder tool called **Qt Designer**. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.

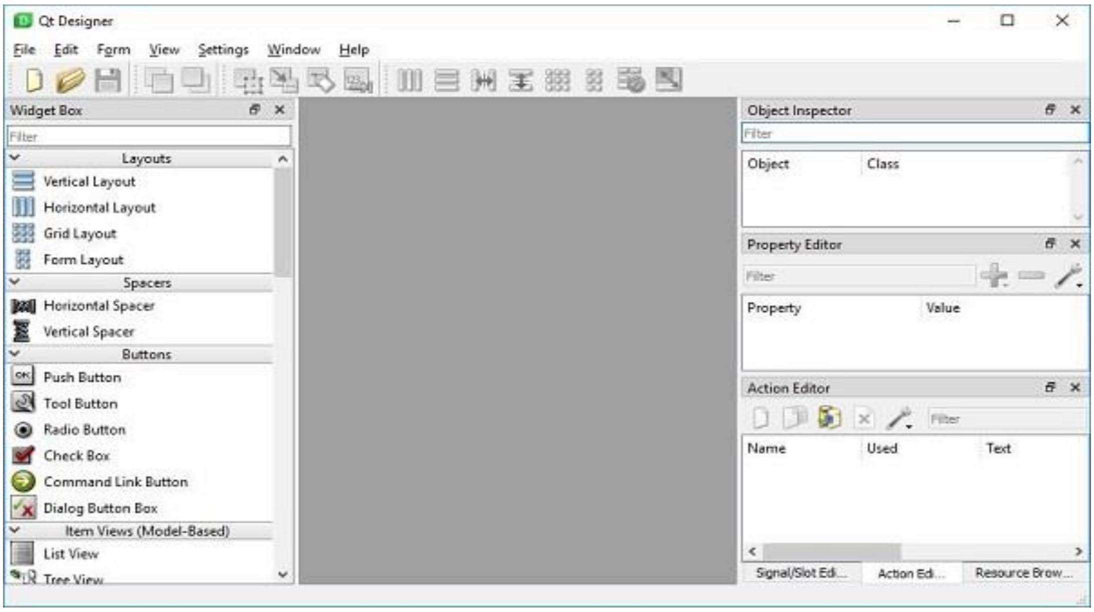
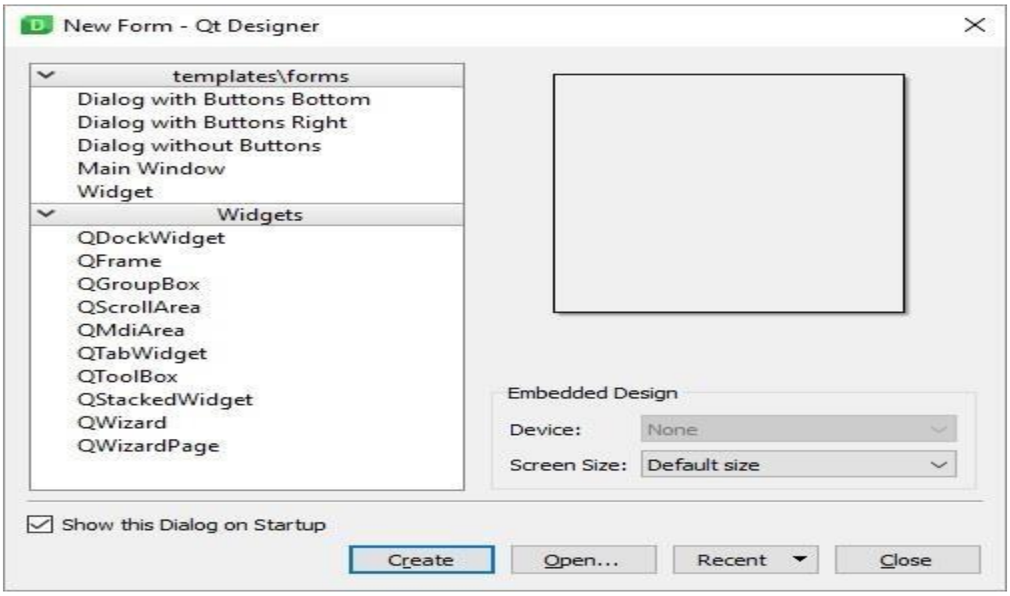


Figure 6.14 PYQT5 main window

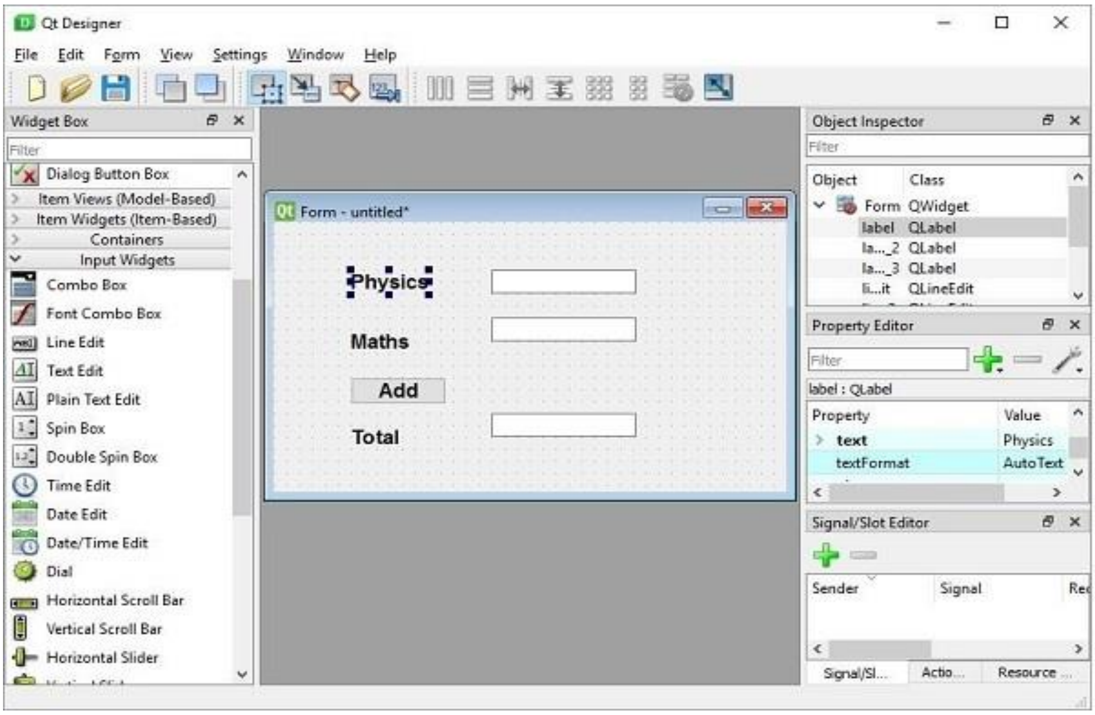
Start designing GUI interface by choosing File → new menu.



Device: None
Screen Size: Default size

Figure 6.15 dialog box

You can then drag and drop required widgets from the widget box on the left pane. You can also assign value to properties of widget laid on the form.



Object	Class
Form	QWidget
label	QLabel
la..._2	QLabel
la..._3	QLabel
li...it	QLineEdit

label : QLabel

Property	Value
text	Physics
textFormat	AutoText

Figure 6.16 form page

This design is translated into Python equivalent by using pyuic5 command line utility. This utility is a wrapper for uic module of Qt toolkit. The usage of pyuic5 is as follows −

```
pyuic5 -x demo.ui -o demo.py
if __name__ == "__main__":
  Import sys
  App = QtGui.QApplication(sys.argv)
  Dialog = QtGui.QDialog()
ui = Ui_Dialog()
ui.setupUi(Dialog)
Dialog.show()
sys.exit(app.exec_())
```

The resultant python script is executed to show the following dialog box −

```
Python demo.py
```

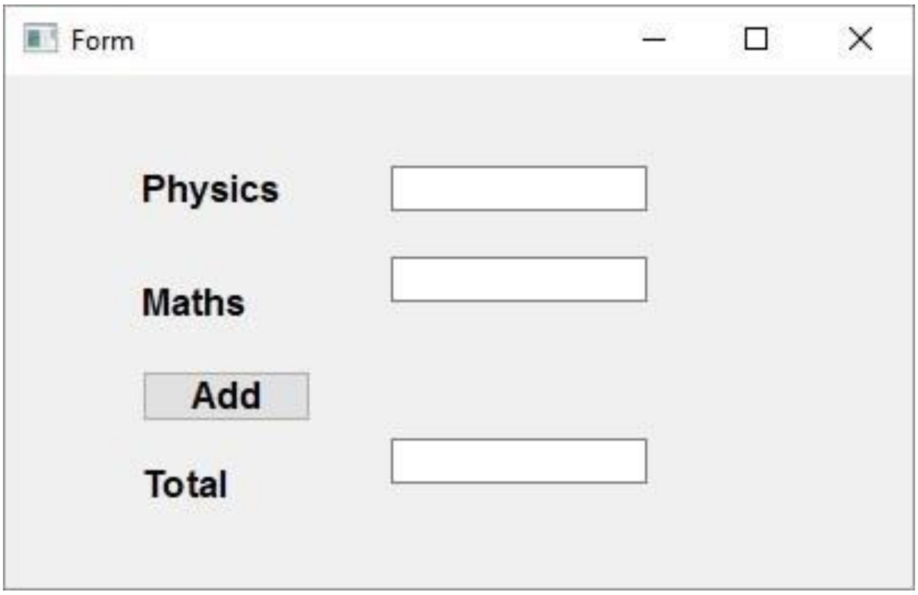


Figure 6.17 form window

The user can input data in input fields but clicking on Add button will not generate any action as it is not associated with any function. Reacting to user-generated response is called as **event handling.**

Unlike a console mode application, which is executed in a sequential manner, a GUI based application is event driven. Functions or methods are executed in response to user's actions like clicking on a button, selecting an item from a collection or a mouse click etc., called **events**.

Widgets used to build the GUI interface act as the source of such events. Each PyQt widget, which is derived from QObject class, is designed to emit **'signal'** in response to one or more events. The signal on its own does not perform any action. Instead, it is 'connected' to a **'slot'**. The slot can be any **callable Python function**.

Using Qt Designer's Signal/Slot Editor
First design a simple form with a LineEdit control and a PushButton.

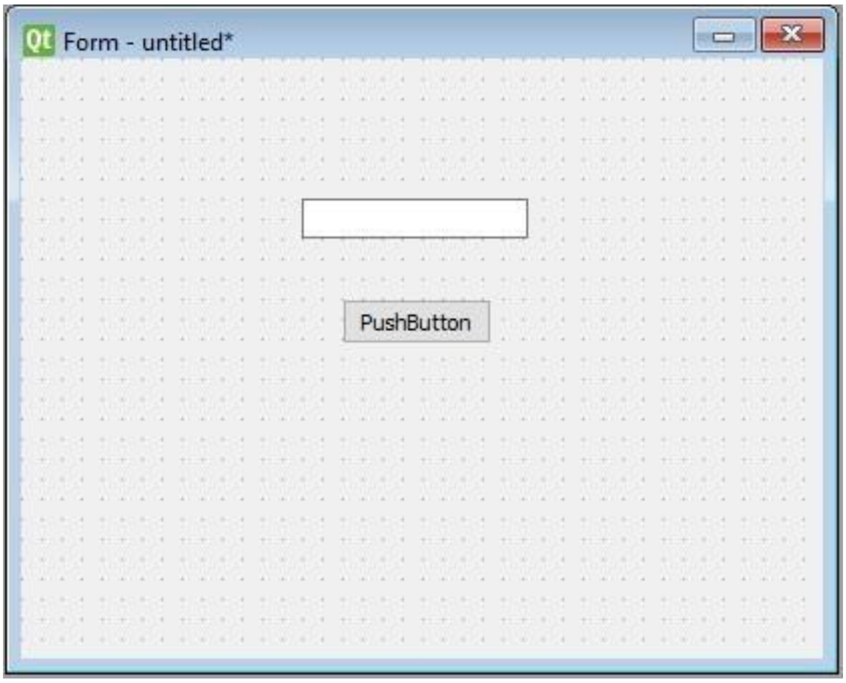


Figure 6.18 design page

It is desired that if button is pressed, contents of text box should be erased. The QLineEdit widget has a clear () method for this purpose. Hence, the button's **clicked** signal is to be connected to **clear ()** method of the text box.

To start with, choose Edit signals/slots from Edit menu (or press F4). Then highlight the button with mouse and drag the cursor towards the textbox

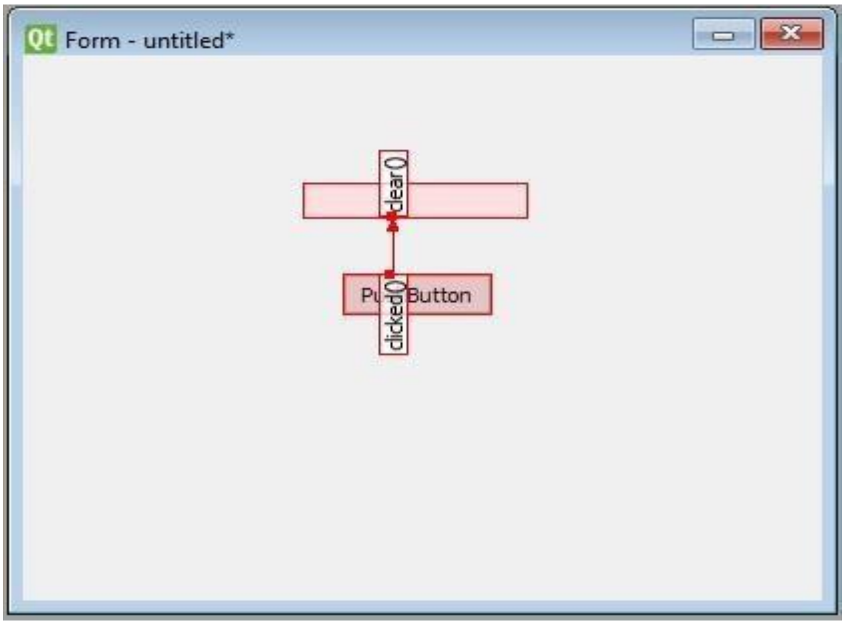
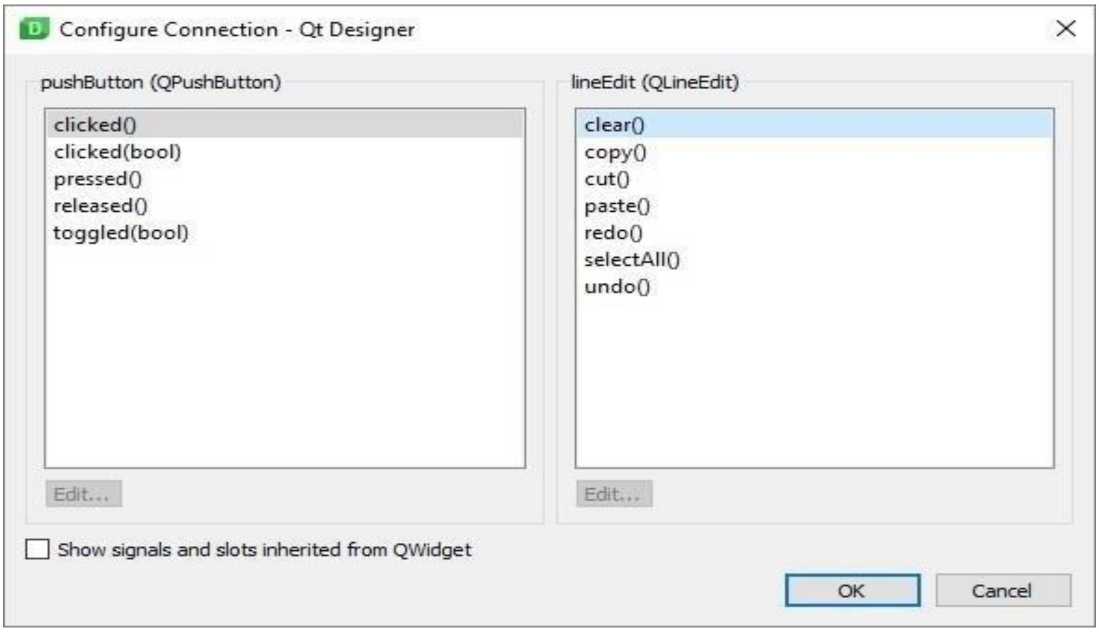


Figure 6.19 Form design page

As the mouse is released, a dialog showing signals of button and methods of slot will be displayed. Select clicked signal and clear () method


Configure Connection - Qt Designer

pushButton (QPushButton)

clicked()
clicked(bool)
pressed()
released()
toggled(bool)

lineEdit (QLineEdit)

clear()
copy()
cut()
paste()
redo()
selectAll()
undo()

□ Show signals and slots inherited from QWidget

Figure 6.20 editing page

CHAPTER 7

HARDWARE REQUIERMENTS & SPECIFICATION

7.1 I5 PROCESSOR

As we all know that Intel is the renowned company that manufactures the processors since 1940s. Every year they introduce different versions of the processors that have more advanced and the new features as compared to the previous one. After the introducing i5 processors, Intel introduced i5 core processors and increases the family members of I series of the processors. It was invented in 2009 as a result of the advanced theories and for the sake to meet the challenges of the modern and latest technologies.

7.1.1 About-Core I5

Developed and manufactured by Intel, the Core i5 is a computer processor, available as dual-core or quad-core. It can be used in both desktop and laptop computers, and is one of four types of processors in the "i" (Intel Core family) series. The first i5 processor was released in September 2009 and new generations of the i5 continue to be released (2020).

The Core i5 processor is available in multiple speeds, ranging from 1.90 GHz up to 3.80 GHz, and it features 3 MB, 4 MB or 6 MB of cache. It utilizes either the LGA 1150 or LGA 1155 socket on a motherboard. Core i5 processors are most often found as quad-core, having four cores. However, a select few high-end Core i5 processors feature six cores.

The most common type of RAM used with a Core i5 processor is DDR3 1333 or DDR3 1600. However, higher performance RAM can be used as well if it's supported by the motherboard.

Figure 7.1 Intel processor

Power usage varies for the Core i5 processors:

- Slower speeds (1.90 GHz to 2.30 GHz) use 11.5 W of power
- Medium speeds (2.60 GHz to 3.10 GHz) use 15 W, 25 W, 28 W or 37 W of power
- Faster speeds (3.20 GHz to 3.80 GHz) use 35 W, 37 W, 45 W, 47 W, 65 W or 84 W of power

7.1.2 Technical Specifications of i5 Processors:

I5 processors basically introduced to manage the different strategies between the low and the advanced working. Some of the technical specifications of the i5 processors that deal with the performing or working capacity and the technicalities of operation of the i5 processors in the computer system are listed below.

1. i5 processors have ability to work with integrated memory and can enhance the performance of the applications. The increase the memory up to 1333 MHz

2. As i5 processors have high speed performing rate so they are able to perform at the maximum CPU rate of 3.6 GHz

3. Turbo technology is present in the device that boosts up the working speed of the computational systems.

4. It also provides the 64 bit architecture for the users for the reliable and much faster working.

5. Micro architecture for the i5 processors was presented by the Nehalem and these processors have a cache rate up to 8 MB.

7.1.3 More advantages of the I5

I5 processors have remarkable advantages that are of great use in the field of computers and technology. Some of them are listed below.

- Dual core processing has the ability to run two independent programs with one hardware.

- I5 processors have improved Pentium base, they have totally new architecture with more integrations and high-speed performance structure.

- Hyper threading technology also enables user to enjoy the high speed and better performance with more reliable outputs. It has 4 tasking threads that allows user to easily execute 3 to 4 programmers at a time.

- Smart memory and cache sequence allows user to enjoy the optimized and efficient data access both direct and sequentially. Effective shortcuts have reduced the access time of the file and system.

- HD graphical features also make these processors distinguished from the others because they are considered as best in their resolution.

7.1.4 Advantages and Applications of Intel Core i5

Mainstream and All-Around Processor

One of the advantages of the Core i5 is that it is a mainstream processor designed for all-around computing requirements. Manufacturers such as Apple and Dell have featured numerous versions of this chip in their high-end computers and laptops.

The aforementioned means that it can handle day-to-day tasks such as web browsing, content consumption, and office productivity. Furthermore, it is capable of handling some tasks that require considerable resources to include average-level video editing and gaming.

Continuously Improving Newer Versions

Core i5 has gone numerous iterations and upgrades since 2009. The latest generations of this processor are typically comparable to the previous generations of the Core i7. In most instances, they are also cheaper and more accessible.

It also has a notable advantage over Core i3. Fundamentally, it has better processing capabilities that translate to better hardware performances. Average consumers and users are better off choosing this processor over a Core i7 and Core i3.

Performance and Flexible Applications

Depending on the specific model or variant, the processor sits within the mid-range to high-range computer usage spectrum. Users can choose from different variants based on their budget and their personal or professional requirements.

Intel Core i5 is essentially positioned as a mid-level and mainstream CPU that is engineered for graphics-intensive gaming, multitasked business productivity applications, and general to above-average high-definition video and audio editing.

7.1.5 Disadvantages and Limitations of Intel Core i5

Performance Limitations and Applications

Of course, it would perform poorly when compared alongside Core i7 and Core i9. These processors have better hardware architecture and specifications because they are positioned for high-level and top-tier computers and use-case scenarios.

It is also important to highlight the fact that the Core i7 is marketed for high-level users while the Core i9 is geared toward enthusiast-level users. The Core i5 would not be able to address the specific resource-intensive requirements of these users.

Intensive Gaming and Video Editing

Remember that this processor can still handle a considerable degree of video editing and gaming. The newer generations could competently accomplish the average needs of video editors and the requirements of some gaming titles.

However, there is a limitation on how much it could perform these tasks. It is typically not ideal for accomplishing intensive games at higher frame rates, as well as editing longer videos. Furthermore, it would be unsuitable for animation and graphics rendering.

A Note on Multitasking Capabilities

The Core line of processors from Intel different in the way they handle processing information. Typical variants of the Core i5 have fewer cores and support fewer simultaneous threads than the Core i7 and Core i9. These variants are not as good at multitasking. A key disadvantage of the Core i5 is that it is not engineered and positioned for demanding work. It would not match the capabilities of current generations of Core i7 and Core i9, as well as equivalent processors from other manufacturers.

7.2 RANDOM ACCESS MEMEORY

RAM is an acronym for ***r**andom **a**ccess **m**emory,* a type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. RAM is found in servers, PCs, tablets, smart phones and other devices, such as printers.

7.2.1 Main Types of RAM

There are two main types of RAM:

1. DRAM (Dynamic Random Access Memory)

2. SRAM (Static Random Access Memory)

DRAM (Dynamic Random Access Memory) – The term dynamic indicates that the memory must be constantly refreshed or it will lose its contents. DRAM is typically used for the main memory in computing devices. If a PC or Smartphone is advertised as having 4GB RAM or 16GB RAM, those numbers refer to the DRAM, or main memory, in the device.

More specifically, most of the DRAM used in modern systems is synchronous DRAM, or SDRAM. Manufacturers also sometimes use the acronym DDR (or DDR2, DDR3, DDR4, etc.) to describe the type of SDRAM used by a PC or server. DDR stands for double data rate, and it refers to how much data the memory can transfer in one clock cycle.

In general, the more RAM a device has, the faster it will perform.

SRAM (Static Random Access Memory) – While DRAM is typically used for main memory, today SRAM is more often used for system cache. SRAM is said to be static because it doesn't need to be refreshed, unlike dynamic RAM, which needs to be refreshed thousands of times per second. As a result, SRAM is faster than DRAM. However, both types of RAM are volatile, meaning that they lose their contents when the power is turned off.

What does RAM do?

RAM is a form of temporary storage that gets wiped when you turn your computer off. RAM offers lightning-fast data access, which makes it ideal for the processes, apps, and programs your computer is actively working on, such as the data needed to surf the internet through your web browser.

To understand RAM, let's use the analogy of a physical desk. Your working space is the top of the desk. That's where you keep everything you frequently use within arm's reach, so you won't waste time searching through your drawers. By contrast, anything you don't use that much or that you want to save for later goes into a desk drawer.

RAM is like the top of your desk at home, where you keep everything you need to access frequently or are currently working on.

What is RAM used for?

RAM is used for immediate data storage and retrieval. Your RAM can process information significantly faster than data on a hard disk — twenty to a hundred times faster, depending on the specific hardware and task.

To accomplish a specific task, computer operating systems load data from the hard disk into RAM to process it. When it’s finished actively working with that data, the computer converts it back into long-term storage.

When you open a program such as Microsoft Word, your computer loads the application into its RAM. If you open a document you already have saved on your computer, your operating system locates the file in long-term storage and copies the information onto its RAM

RAM is used for immediate data storage and retrieval — any task that requires quick access to computing resources.

Another use for RAM is to help your computer load previously-accessed information more quickly. When you first turn on your computer and launch any application, such as PowerPoint or Spotify, it takes a while to load. But if you close a program and then relaunch it, the software opens almost instantly (especially if your PC is optimized for performance). That’s because the data needed to load the app is stored in the significantly faster RAM rather than the hard disk.

How much memory do I need?

The more RAM your computer has, the faster it runs. If your device is old, you might need to upgrade the RAM or other hardware. Every open application (including browser tabs) consumes RAM. When you run out of RAM, your computer has to move things around to free up space on the hard disk, which slows down your computer.

CHAPTER 8

CONCLUSION

HAAR cascade classifier and facial landmark algorithm is used to detect many faces and area of interested features like eyes, nose, mouth etc. Accurate detection of features increases rapidly when a camera of high specifications is used. We are implementing this system in vehicles, where only the driver's face i.e, only one face is detected, when the face is detected and recognized as drowsy, he will get an voice alert . This project presents the real time implementation of drowsiness detection which is invariant to illumination and performs well under various lighting conditions. Correlation coefficient template matching provides a super-fast way to track the eyes and mouth. The proposed system achieves an overall accuracy of 94.58% in four test cases, which is highest in comparison to the recent methods. A high detection rate and reduced false alarms makes sure that this system can efficiently reduce the number of fatalities every year that cause due to drowsiness and carelessness of drivers.

IMPLEMENTATION

1. SOURCE CODE

Page 1:

```
from PyQt5 import QtCore, QtGui, QtWidgets
from page2 import Ui_Form22
class Ui_Form(object):
def setupUi(self, Form):
Form.setObjectName("Form")
Form.resize(800,600)
self.label = QtWidgets.QLabel(Form)
self.label.setGeometry(QtCore.QRect(0, 0, 800,600))
self.label.setText("")
self.label.setPixmap(QtGui.QPixmap("44.webp"))
self.label.setScaledContents(True)
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(Form)
self.label_2.setGeometry(QtCore.QRect(10, 0, 831, 61))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(27)
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setStyleSheet("color: rgb(0,0,0);background-color: rgb(149, 149, 149)")
self.label_2.setObjectName("label_2")
self.pushButton = QtWidgets.QPushButton(Form)
self.pushButton.setGeometry(QtCore.QRect(250, 500, 171, 51))
self.pushButton.clicked.connect(self.next_page)
        font = QtGui.QFont()
```

```
font.setFamily("Times New Roman")
font.setPointSize(14)
font.setBold(True)
font.setWeight(75)
self.pushButton.setFont(font)
self.pushButton.setStyleSheet("border-radius:20px;\n"
"background-color: rgb(149, 149, 149);")
self.pushButton.setObjectName("pushButton")
self.label_3 = QtWidgets.QLabel(Form)
self.label_3.setGeometry(QtCore.QRect(280, 200, 331, 51))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(18)
font.setBold(True)
font.setWeight(75)
self.label_3.setFont(font)
self.label_3.setStyleSheet("color: rgb(255, 255, 255);")
self.label_3.setText("")
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(Form)
self.label_4.setGeometry(QtCore.QRect(280, 260, 331, 51))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(18)
font.setBold(True)
font.setWeight(75)
self.label_4.setFont(font)
self.label_4.setStyleSheet("color: rgb(255, 255, 255);")
self.label_4.setText("")
self.label_4.setObjectName("label_4")
self.label_5 = QtWidgets.QLabel(Form)
self.label_5.setGeometry(QtCore.QRect(280, 320, 331, 51))
```

```
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(18)
font.setBold(True)
font.setWeight(75)
self.label_5.setFont(font)
self.label_5.setStyleSheet("color: rgb(255, 255, 255);")
self.label_5.setText("")
self.label_5.setObjectName("label_5")

self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)

def retranslateUi(self, Form):
        _translate = QtCore.QCoreApplication.translate
Form.setWindowTitle(_translate("Form", "Form"))
self.label_2.setText(_translate("Form", "DRIVER DROWSINESS DETECTION"))
self.label_3.setText(_translate("Form", "KIRUTHIKA S"))
self.label_4.setText(_translate("Form", "SUBHA SHRI A.V"))
self.label_5.setText(_translate("Form", "KAVI PRIYA S"))
self.pushButton.setText(_translate("Form", "NEXT"))

def next_page(self):
print("Next page button clicked")  # Check if the button click is detected
self.Form22 = QtWidgets.QMainWindow()
self.ui = Ui_Form22()
self.ui.setupUi(self.Form22)
self.Form22.show()

if __name__ == "__main__":
import sys
    app = QtWidgets.QApplication(sys.argv)
    Form = QtWidgets.QWidget()
ui = Ui_Form()
ui.setupUi(Form)
Form.show()
sys.exit(app.exec_())
page2:
from PyQt5 import QtCore, QtGui, QtWidgets
from page3 import Ui_Form
class Ui_Form22(object):
def setupUi(self, Form22):
        Form22.setObjectName("Form22")
        Form22.resize(800,600)
self.label = QtWidgets.QLabel(Form22)
self.label.setGeometry(QtCore.QRect(0, 0, 800,600))
self.label.setText("")
self.label.setPixmap(QtGui.QPixmap("33.jpg"))
self.label.setScaledContents(True)
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(Form22)
self.label_2.setGeometry(QtCore.QRect(320, 0, 181, 51))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(27)
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setStyleSheet("color: rgb(0,0,0);background-color: rgb(149, 149, 149)")
self.label_2.setObjectName("label_2")
self.label_3 = QtWidgets.QLabel(Form22)
self.label_3.setGeometry(QtCore.QRect(80, 290, 261, 41))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(22)
font.setBold(True)
font.setWeight(75)
self.label_3.setFont(font)
self.label_3.setStyleSheet("color: rgb(255,255,255);")
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(Form22)
self.label_4.setGeometry(QtCore.QRect(70, 190, 261, 41))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(22)
font.setBold(True)
font.setWeight(75)
self.label_4.setFont(font)
self.label_4.setStyleSheet("color: rgb(255,255,255);")
self.label_4.setObjectName("label_4")
self.lineEdit = QtWidgets.QLineEdit(Form22)
self.lineEdit.setGeometry(QtCore.QRect(350, 190, 251, 41))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
self.lineEdit.setFont(font)
self.lineEdit.setObjectName("lineEdit")
self.lineEdit_2 = QtWidgets.QLineEdit(Form22)
self.lineEdit_2.setGeometry(QtCore.QRect(350, 290, 251, 41))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
self.lineEdit_2.setFont(font)
self.lineEdit_2.setEchoMode(QtWidgets.QLineEdit.Password)
self.lineEdit_2.setObjectName("lineEdit_2")
self.pushButton = QtWidgets.QPushButton(Form22)
self.pushButton.setGeometry(QtCore.QRect(270, 450, 271, 51))
    font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
font.setBold(True)
font.setWeight(75)
self.pushButton.setFont(font)
self.pushButton.setStyleSheet("border-radius:20px;\n"
"background-color: rgb(149, 149, 149);")
self.pushButton.setObjectName("pushButton")
self.pushButton.clicked.connect(self.next_page)
self.lineEdit_6 = QtWidgets.QLineEdit(Form22)
self.lineEdit_6.setGeometry(QtCore.QRect(60, 350, 381, 51))
self.lineEdit_6.setStyleSheet("background-color: rgba(0, 0, 0,0);\n"
                "border:none;\n"
                "border-bottom:2px solid rgba(255,255,255,255);\n"
                "color: rgb(255, 255, 255);\n"
                "paddin-bottom:7px;\n"
                "font: 14pt \"MS Shell Dlg 2\";\n"
                "")
self.lineEdit_6.setObjectName("lineEdit_6")
self.lineEdit_21 = QtWidgets.QLineEdit(Form22)
self.lineEdit_21.setGeometry(QtCore.QRect(60, 400, 381, 51))
self.lineEdit_21.setStyleSheet("background-color: rgba(0, 0, 0,0);\n"
                 "border:none;\n"
                 "border-bottom:2px solid rgba(255,255,255,255);\n"
                 "color: rgb(255,255, 255);\n"
                 "paddin-bottom:7px;\n"
                 "font: 14pt \"MS Shell Dlg 2\";\n"
                 "")
self.lineEdit_21.setObjectName("lineEdit_21")
self.retranslateUi(Form22)
```

```
QtCore.QMetaObject.connectSlotsByName(Form22)
def retranslateUi(self, Form22):
        _translate = QtCore.QCoreApplication.translate
        Form22.setWindowTitle(_translate("Form2", "Form"))
self.label_2.setText(_translate("Form2", "LOG IN "))
self.label_3.setText(_translate("Form2", "PASSWORD:"))
self.label_4.setText(_translate("Form2", "USER_NAME:"))
self.pushButton.setText(_translate("Form2", "LOG IN"))
def next_page(self):
        n1 = self.lineEdit.text()
        n2 = self.lineEdit_2.text()
if n1 == 'project':
print("correct username")
self.lineEdit_6.setText("correct username")
if n2 == '123':
print("correct password")
self.lineEdit_21.setText("correct password")
self.Form = QtWidgets.QMainWindow()
self.ui = Ui_Form()
self.ui.setupUi(self.Form)
self.Form.show()
else:
print("wrong password")
self.lineEdit_21.setText("wrong password")
else:
print("wrong username")
self.lineEdit_6.setText("wrong username")
if __name__ == "__main__":
import sys
    app = QtWidgets.QApplication(sys.argv)
    Form22 = QtWidgets.QWidget()
ui = Ui_Form22()
```

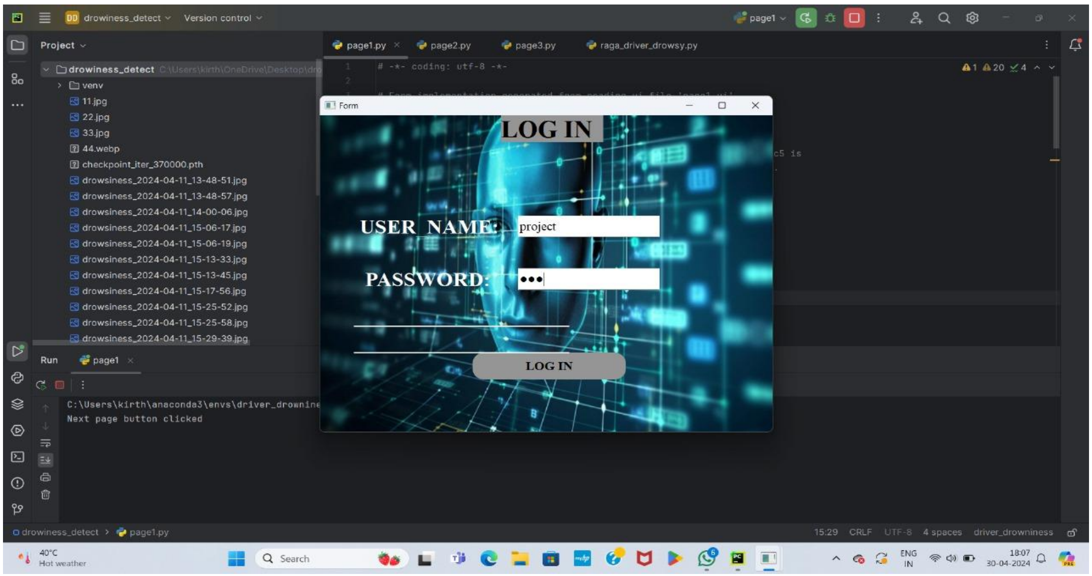
```
ui.setupUi(Form22)
    Form22.show()
sys.exit(app.exec_())
Page3:from PyQt5 import QtCore, QtGui, QtWidgets
from raga_driver_drowsyimport detect
class Ui_Form(object):
def setupUi(self, Form):
Form.setObjectName("Form")
Form.resize(800,600)
self.label = QtWidgets.QLabel(Form)
self.label.setGeometry(QtCore.QRect(0, 0, 800,600))
self.label.setText("")
self.label.setPixmap(QtGui.QPixmap("22.jpg"))
self.label.setScaledContents(True)
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(Form)
self.label_2.setGeometry(QtCore.QRect(250, 0, 291, 61))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(27)
font.setBold(True)
font.setWeight(75)
self.label_2.setFont(font)
self.label_2.setStyleSheet("color: rgb(0,0,0);background-color: rgb(149, 149, 149)")
self.label_2.setObjectName("label_2")
self.pushButton = QtWidgets.QPushButton(Form)
self.pushButton.setGeometry(QtCore.QRect(250, 500, 171, 51))
self.pushButton.clicked.connect(self.next_page)
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
font.setBold(True)
font.setWeight(75)
self.pushButton.setFont(font)
self.pushButton.setStyleSheet("border-radius:20px;\n"
"background-color: rgb(149, 149, 149);")
self.pushButton.setObjectName("pushButton")
self.label_3 = QtWidgets.QLabel(Form)
self.label_3.setGeometry(QtCore.QRect(270, 160, 231, 51))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(18)
font.setBold(True)
font.setWeight(75)
self.label_3.setFont(font)
self.label_3.setStyleSheet("color: rgb(255, 255, 255);")
self.label_3.setText("")
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(Form)
self.label_4.setGeometry(QtCore.QRect(280, 260, 231, 51))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(18)
font.setBold(True)
font.setWeight(75)
self.label_4.setFont(font)
self.label_4.setStyleSheet("color: rgb(255, 255, 255);")
self.label_4.setText("")
self.label_4.setObjectName("label_4")
self.label_5 = QtWidgets.QLabel(Form)
self.label_5.setGeometry(QtCore.QRect(280, 340, 231, 51))
        font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(18)
font.setBold(True)
font.setWeight(75)
self.label_5.setFont(font)
self.label_5.setStyleSheet("color: rgb(255, 255, 255);")
self.label_5.setText("")
self.label_5.setObjectName("label_5")
self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)
def retranslateUi(self, Form):
        _translate = QtCore.QCoreApplication.translate
Form.setWindowTitle(_translate("Form", "Form"))
self.label_2.setText(_translate("Form", "DETECTION"))
self.pushButton.setText(_translate("Form", "DETECT"))
def next_page(self):
print("Detection button clicked")  # Check if the button click is detected
detect()
if __name__ == "__main__":
import sys
    app = QtWidgets.QApplication(sys.argv)
    Form = QtWidgets.QWidget()
ui = Ui_Form()
ui.setupUi(Form)
Form.show()
sys.exit(app.exec_())
Raga drowsy.py
def detect():
from datetime import datetime
from scipy.spatialimport distance
from imutilsimportface_utils
import imutils
import dlib
import cv2
import pyttsx3
engine=pyttsx3.init()
def eye_aspect_ratio(eye):
        A = distance.euclidean(eye[1], eye[5])
        B = distance.euclidean(eye[2], eye[4])
        C = distance.euclidean(eye[0], eye[3])
        ear = (A + B) / (2.0 * C)
return ear
    thresh = 0.25
frame_check = 20
detect = dlib.get_frontal_face_detector()
    predict = dlib.shape_predictor(".\shape_predictor_68_face_landmarks.dat")  # Dat
file is the crux of the code
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_68_IDXS["right_eye"]
    cap = cv2.VideoCapture(0)
    flag = 0
while True:
        ret, frame = cap.read()
        frame = imutils.resize(frame, width=450)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        subjects = detect(gray, 0)
for subject in subjects:
            (x,y,w,h)= face_utils.rect_to_bb(subject)
            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
            shape = predict(gray, subject)
            shape = face_utils.shape_to_np(shape)  # converting to NumPy Array
leftEye = shape[lStart:lEnd]
rightEye = shape[rStart:rEnd]
leftEAR = eye_aspect_ratio(leftEye)
rightEAR = eye_aspect_ratio(rightEye)
            ear = (leftEAR + rightEAR) / 2.0
```

```
leftEyeHull = cv2.convexHull(leftEye)
rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
if ear < thresh:

                flag += 1
print(flag)
if flag >= frame_check:
                     cv2.putText(frame,
"****************ALERT!****************",                              (10,
30),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                     cv2.putText(frame,
"****************ALERT!****************",                              (10,
325),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
                     pyttsx3.speak("Drowsy detect dont sleep")
current_time = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
img_name=f"drowsiness_{current_time}.jpg"
cv2.imwrite(img_name, frame)
from email.mime.multipartimportMIMEMultipart
from email.mime.textimportMIMEText
from email.mime.imageimportMIMEImage
import smtplib
from email.mime.baseimportMIMEBase
from email import encoders
strFrom = 'jeevithaproject123@gmail.com'
strTo = 'subhaarumugam2003@gmail.com'
# Create the root message and fill in the from, to, and subject headers
msgRoot = MIMEMultipart('related')
msgRoot['Subject'] = 'Driver drowsiness'
msgRoot['From'] = strFrom
msgRoot['To'] = strTo
```

```
msgRoot.preamble = 'This is a multi-part message in MIME format.'
msgAlternative = MIMEMultipart('alternative')
msgRoot.attach(msgAlternative)
mail_message_Text = MIMEText('Drowsiness detected')
msgAlternative.attach(mail_message_Text)
sending_image = open(f"drowsiness_{current_time}.jpg", 'rb')
msgImage = MIMEImage(sending_image.read())
sending_image.close()
# Define the image's ID as referenced above
msgImage.add_header('Content-ID', '<image1>')
msgRoot.attach(msgImage)

                        smtp = smtplib.SMTP('smtp.gmail.com', 587)
smtp.starttls()
smtp.login('jeevithaproject123@gmail.com', 'ntqeyqgphxjjsphr')
print("mail id and password correct")
smtp.sendmail(strFrom, strTo, msgRoot.as_string())
print("mail send")
smtp.quit()
else:
            flag = 0
cv2.imshow("Frame", frame)
      key = cv2.waitKey(1) &0xFF
if key == ord("q"):
break
cv2.destroyAllWindows()
cap.stop()
```

2 . SCREENSHOTS

Step 1:

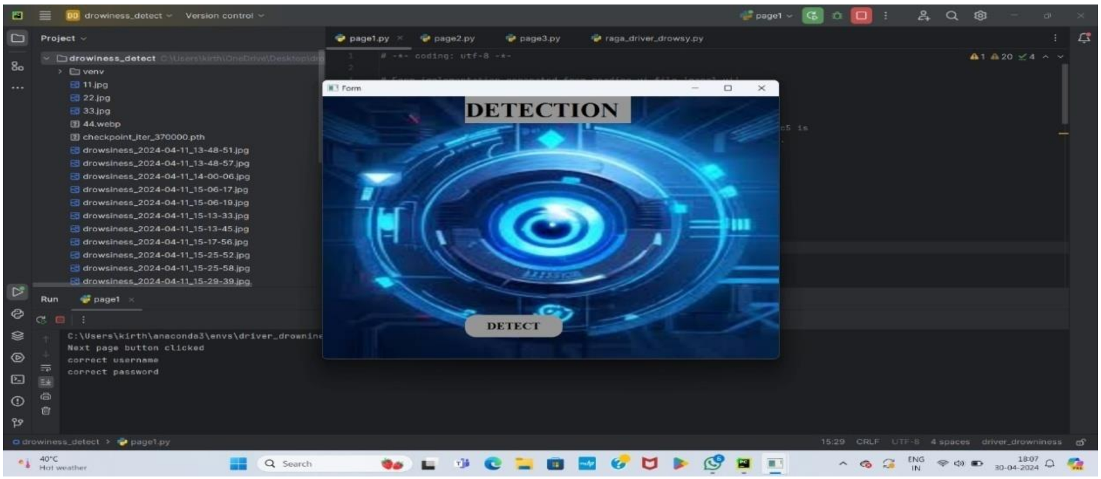


USER NAME: project

PASSWORD: •••

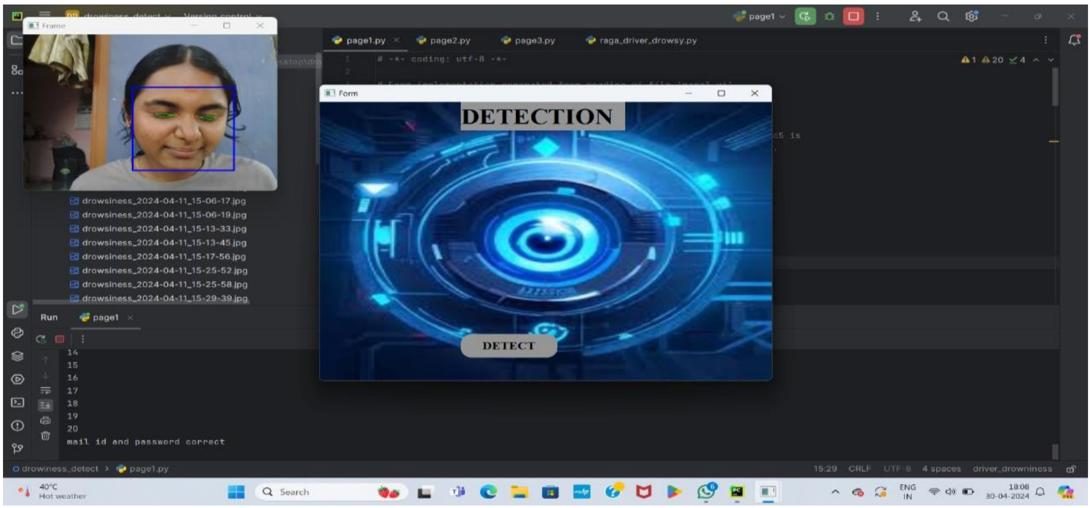
```
C:\Users\kirth\anaconda3\envs\driver_drownine
Next page button clicked
```

Step 2:



```
C:\Users\kirth\anaconda3\envs\driver_drownine
Next page button clicked
correct username
correct password
```

Step 3:

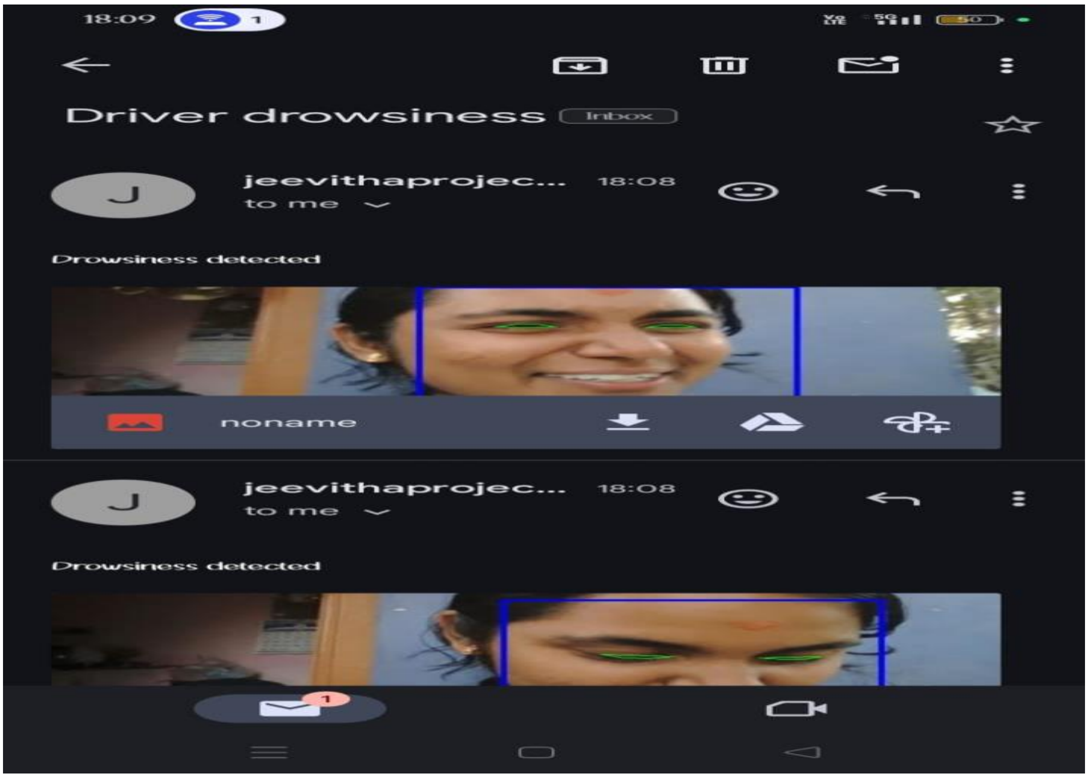


```
drowsiness_2024-04-11_15-06-17.jpg
drowsiness_2024-04-11_15-06-19.jpg
drowsiness_2024-04-11_15-13-33.jpg
drowsiness_2024-04-11_15-13-45.jpg
drowsiness_2024-04-11_15-17-56.jpg
drowsiness_2024-04-11_15-25-52.jpg
drowsiness_2024-04-11_15-25-58.jpg
drowsiness_2024-04-11_15-29-39.jpg
```

```
# -*- coding: utf-8 -*-
```

```
14
15
16
17
18
19
20
mail id and password correct
```

Step 4:



Driver drowsiness Inbox

jeevithaprojec... 18:08
to me

Drowsiness detected

jeevithaprojec... 18:08
to me

Drowsiness detected

REFERENCES

[1] Du, Yong, et al. "Driver fatigue detection based on eye state analysis." Proceedings of the 11th Joint Conference on Information Sciences. 2018.

[2] Choi, In-Ho, Sung Kyung Hong, and Yong-Guk Kim. "Realtime categorization of driver's gaze zone using the deep learning techniques." 2017 International Conference on Big Data and Smart Computing (BigComp). IEEE, 2017.

[3] Tawari, Ashish, Kuo Hao Chen, and Mohan Manubhai Trivedi. "Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation" Intelligent Transportation Systems (ITSC), 2016 IEEE 17th International Conference on. IEEE, 2016.

[4] Singh, R. K., et al. "A real-time heart-rate monitor using noncontact electrocardiogram for automotive drivers." 2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI). IEEE, 2016.

[5] Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57.2 (2016): 137-154.

[6] Basilio, Jorge Alberto Marcial, et al. "Explicit image detection using YCbCr space color model as skin detection." Proceedings of the 2016 American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications, IEEE ICCE 2016.

[7] Mahmood, Arif, and Sohaib Khan. "Correlationcoefficientbased fast template matching through partial elimination." Image Processing, IEEE Transactions on 21.4 (2016): 2099-2108.

[8] P. Viola et M. Jones, "Rapid object detection using boosted cascade of simple features", Proceedings IEEE Conf. on Computer Vision and Pattern Recognition.

[9] Junguk Cho, ShahnamMirzaei, Jason Oberg, Ryan Kastner "FPGA-Based Face Detection System Using Haar Classifiers", 2017, Monterey, California, USA.

[10] G. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library," O'Reilly Media, Inc., 2019.

[11] Tripathy, BirendraNath and Dash, Abinash (2018) Prototype Drowsiness Detection System. B.Tech thesis.

[12] MehrdadSabet, Reza A. Zoorofi, KhosroSadeghniiat-Haghighit, Maryam Sabbaghian "A new System for Driver Drowsiness and Distraction Detection" 20th Iranian Conference on Electrical Engineering (ICEE) may 12-15 2018, Tehran, Iran

[13] Shinfeng D. Lin, Jia-Jen Lin, Chi YaoChung, "Sleepy Eye's Recognition for Drowsiness Detection"2017 International Symposium on Biometric and Security technologies.

[14] PrantoHidayaRusmin, Andrew B. Osmond, AriefSyaichu-Rohmann "Design and Implementation of Driver's Drowsiness Detection System on Digitalized Driver System-2018" 2018 IEEE 3rd International Conference on System Engineering and Technology, 19 - 20 Aug. 2018, Shah Alam, Malaysia.

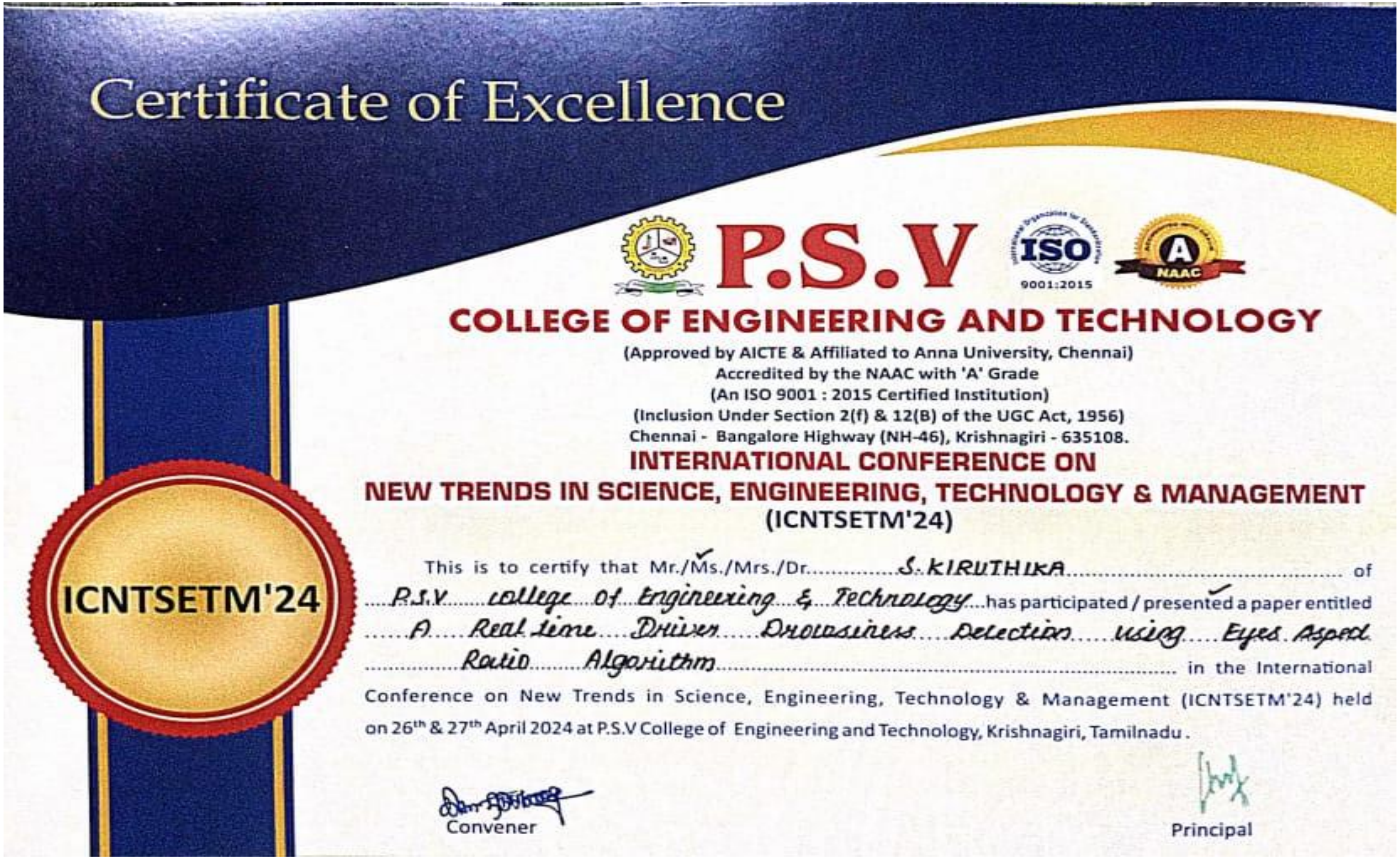
[15] RituUpadhayay, RakeshKumarYadav "Kernel Principle Component Analysis in Face Recognition System: A Survey", M.Tech Scholar CSE ,SET, IFTM University, India, Computer Science &Engg, SET, IFTM University, India

[16] Singh HimaniParmar, MehulJajal, YadavPriyankaBrijbhan "Drowsy Driver Warning System Using Image Processing Electronics & Communication", GEC, Bharuch, Gujarat.

[17] Paul Viola, Michael J. Jones, Robust RealTime Face Detection, International Journal of Cumputer Vision 57(2), 2017.

[18] Gary B. Huang, Manu Ramesh, Tamara Berg, Erik Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, University of Massachusetts, Amherst, Technical Report, October 2019.

[19] Paul Viola, Michael J. Jones, Robust Realtime Face Detection, International Journal of Computer Vision 57(2).



Certificate of Excellence

ICNTSETM'24

P.S.V

ISO 9001:2015

A NAAC

COLLEGE OF ENGINEERING AND TECHNOLOGY

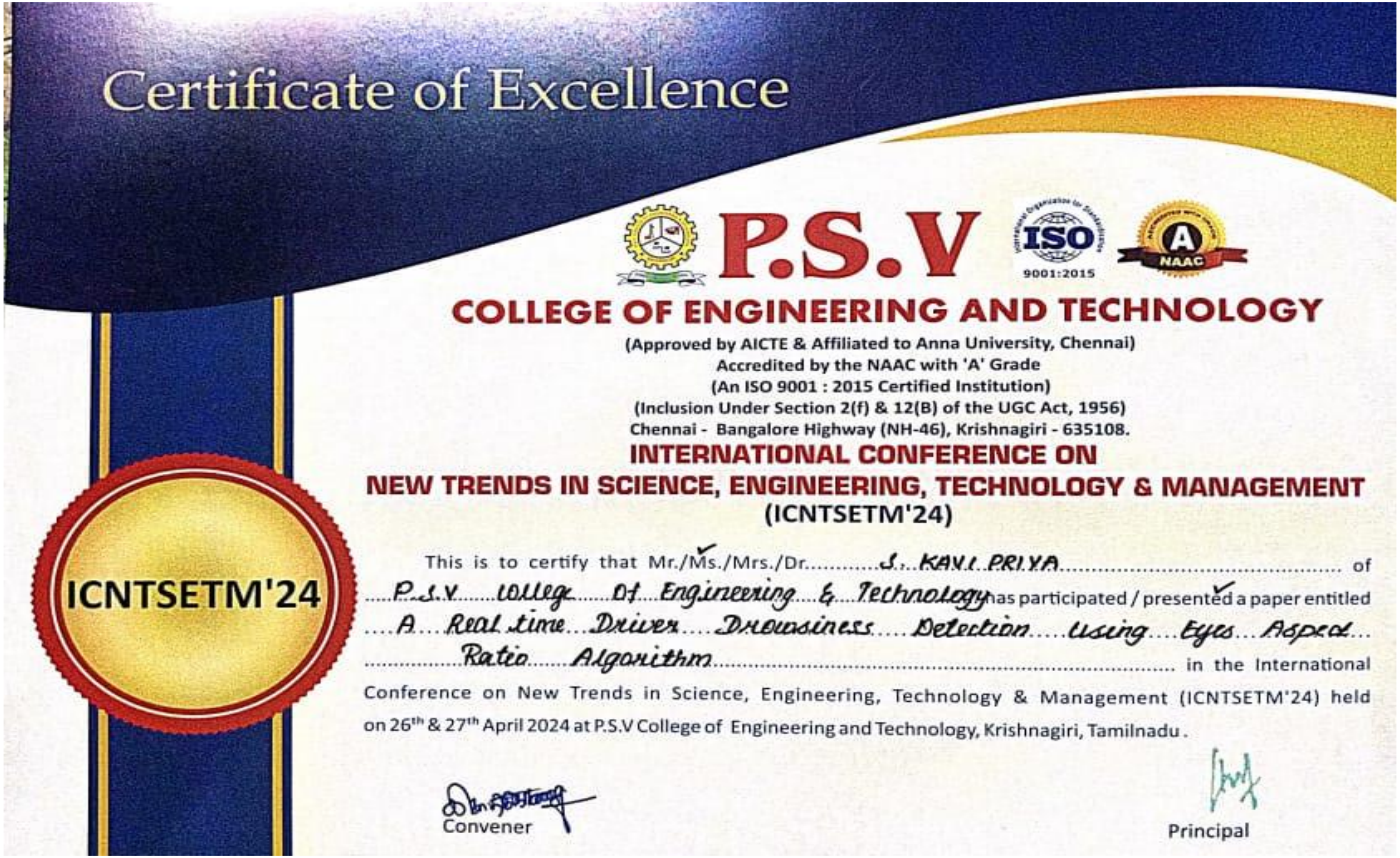
(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited by the NAAC with 'A' Grade
(An ISO 9001 : 2015 Certified Institution)
(Inclusion Under Section 2(f) & 12(B) of the UGC Act, 1956)
Chennai - Bangalore Highway (NH-46), Krishnagiri - 635108.

INTERNATIONAL CONFERENCE ON
NEW TRENDS IN SCIENCE, ENGINEERING, TECHNOLOGY & MANAGEMENT
(ICNTSETM'24)

This is to certify that Mr./[✓ Ms.]/Mrs./Dr. .... S. KIRUTHIKA .... of
.... P.S.V college of Engineering & Technology ... has participated / [✓ presented] a paper entitled
..... A Real time Driver Drowsiness Detection using Eyes Aspect
.... Ratio Algorithm .... in the International
Conference on New Trends in Science, Engineering, Technology & Management (ICNTSETM'24) held
on 26th & 27th April 2024 at P.S.V College of Engineering and Technology, Krishnagiri, Tamilnadu.

Convener

Principal



Certificate of Excellence

P.S.V

ISO 9001:2015

A NAAC

COLLEGE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited by the NAAC with 'A' Grade
(An ISO 9001 : 2015 Certified Institution)
(Inclusion Under Section 2(f) & 12(B) of the UGC Act, 1956)
Chennai - Bangalore Highway (NH-46), Krishnagiri - 635108.

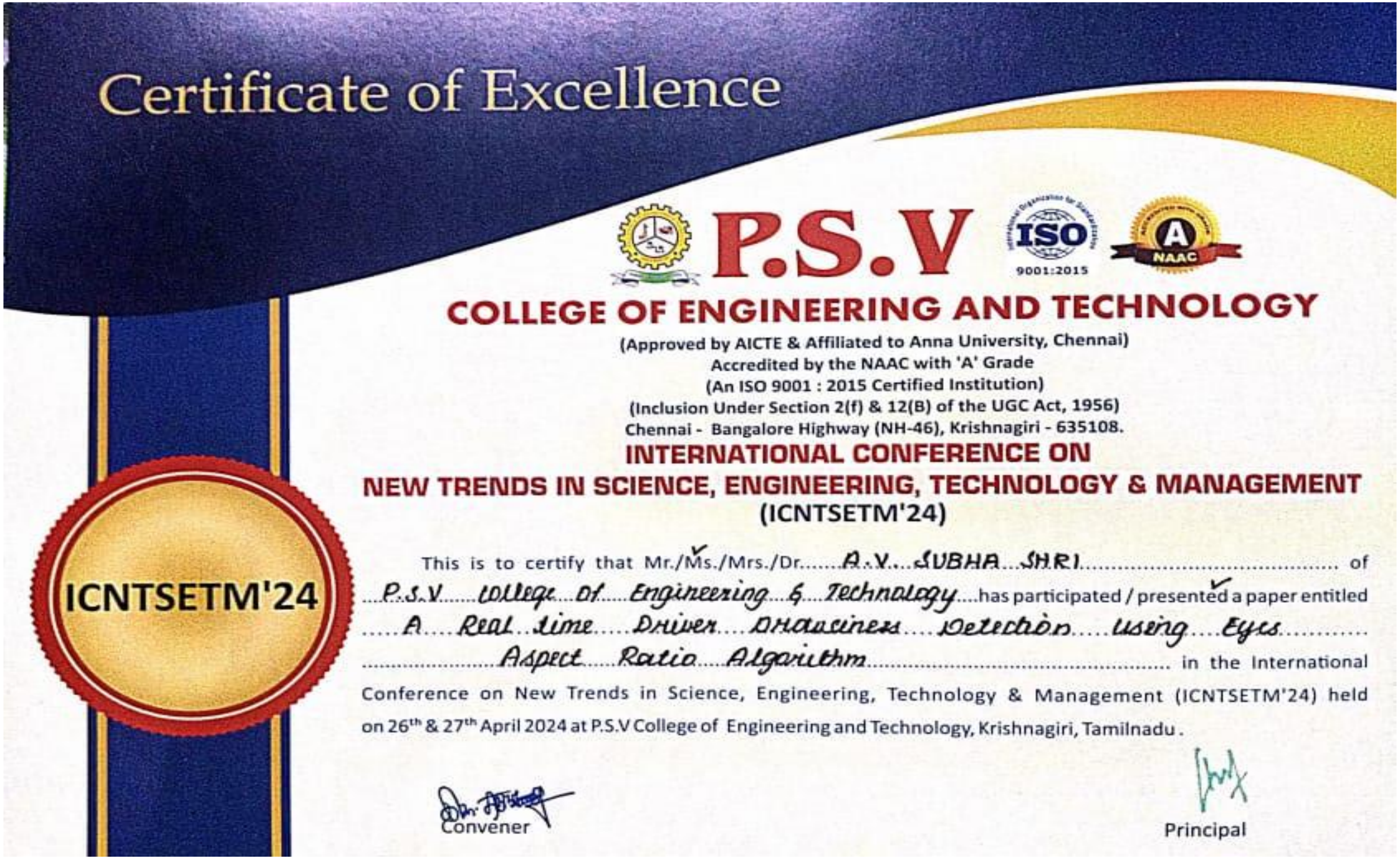
INTERNATIONAL CONFERENCE ON
NEW TRENDS IN SCIENCE, ENGINEERING, TECHNOLOGY & MANAGEMENT
(ICNTSETM'24)

ICNTSETM'24

This is to certify that Mr./Ms./Mrs./Dr. S. KAVI PRIYA of P.S.V college Of Engineering & Technology has participated / presented a paper entitled A Real time Driver Drowsiness Detection using Eyes Aspect Ratio Algorithm in the International Conference on New Trends in Science, Engineering, Technology & Management (ICNTSETM'24) held on 26th & 27th April 2024 at P.S.V College of Engineering and Technology, Krishnagiri, Tamilnadu.

Convener

Principal



Certificate of Excellence

ICNTSETM'24

P.S.V

COLLEGE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE & Affiliated to Anna University, Chennai)
Accredited by the NAAC with 'A' Grade
(An ISO 9001 : 2015 Certified Institution)
(Inclusion Under Section 2(f) & 12(B) of the UGC Act, 1956)
Chennai - Bangalore Highway (NH-46), Krishnagiri - 635108.

INTERNATIONAL CONFERENCE ON
NEW TRENDS IN SCIENCE, ENGINEERING, TECHNOLOGY & MANAGEMENT
(ICNTSETM'24)

This is to certify that Mr./[✓ Ms.]/Mrs./Dr. A.V. SUBHA SHRI of P.S.V College of Engineering & Technology has participated / [✓ presented] a paper entitled A Real time Driver Drowsiness Detection using Eyes Aspect Ratio Algorithm in the International Conference on New Trends in Science, Engineering, Technology & Management (ICNTSETM'24) held on 26th & 27th April 2024 at P.S.V College of Engineering and Technology, Krishnagiri, Tamilnadu.

Convener

Principal