

# FACE RECOGNITION ATTENDANCE SYSTEM USING PYTHON

*by Gayatri Kumari*

---

**Submission date:** 14-Apr-2022 07:44PM (UTC+0530)

**Submission ID:** 1810621364

**File name:** MAJOR\_PROJECT\_PT\_2\_THESIS\_1.docx (9.49M)

**Word count:** 15226

**Character count:** 79414



A project report on

## **FACE RECOGNITION ATTENDANCE SYSTEM USING PYTHON**

3

submitted in partial fulfillment of the requirements for the degree of

B. Tech

In

Electronics and Computer Science Engineering

By

<b>ANKIT KUSHWAHA</b>	<b>1830011</b>
<b>GAYATRI KUMARI</b>	<b>1830021</b>
<b>NIKET DUJARI</b>	<b>1830033</b>

under the guidance of

**Dr. AMIT KUMAR V. JHA**

3

School of Electronics Engineering

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

(Deemed to be University)

**BHUBANESWAR**

**APRIL 2022**

## CERTIFICATE

This is to certify that the **project (part-II)** report entitled "**FACE RECOGNITION ATTENDANCE SYSTEM USING PYTHON**" submitted by

**ANKIT KUSHWAHA** **1830011**

**GAYATRI KUMARI** **1830021**

**NIKET DUJARI** **1830033**

<sup>3</sup> in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology in Electronics and Computer Science Engineering** is a record of the work carried out under my guidance and supervision at School of Electronics Engineering, KIIT (Deemed to be University).

### Digital Signature

Signature of Supervisor

Dr. AMIT KUMAR V. JHA

School of Electronics Engineering

KIIT (Deemed to be University)

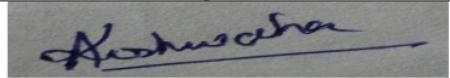
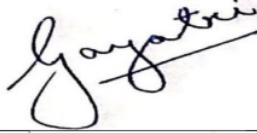
## ACKNOWLEDGEMENT

We are extremely grateful, having completing this project with so many people around us who were ready to help and solve all our doubts at any moment of time, and especially our Teacher Coordinator-**Prof. DR. AMIT KUMAR V. JHA**, who not only mentored us but also guided us wherever we were going wrong and paved our way by constantly checking up on us from time to time.

We would particularly like to thank him, on how patient he was throughout our project and gave us the required time to complete the project successfully. We were really in awe to see how calm he was, in listening and understanding our project and helping us by giving the best possible solutions to all our problems.

<sup>3</sup>  
We are also very thankful to **Prof. ARUN KUMAR RAY**, Dean (School of Electronics) and **Prof. AMLAN DATTA**, Associate Dean (School of Electronics) <sup>3</sup> for their support and suggestions during our course of the project work in the final year of our undergraduate course.

### STUDENT SIGNATURE

Roll Number	Name	Signature
1830011	ANKIT KUSHWAHA	
1830021	GAYATRI KUMARI	
1830033	NIKET DUJARI	

DATE: 15-04-2022

## **ABSTRACT**

We are all aware of the current pandemic situation and the pandemic has forced everyone to be contact less with each other. Attendance is almost mandatory everywhere be it educational institutions or workplaces. Since the situation is not at its normal pace ,hence taking attendance manually or via biometric can be risky . Thus facial recognition is the most feasible option available in the current situation for organizations to make attendance and visitor's entry contact less.

We were really inquisitive as to how the back-end and the entire process of contact less face recognition attendance system is done, and thus we decided to do this as a part of our project. We ensure you, we have given our best in creating the first part of our project.

Recognizing people by their faces in photos and video feeds is commonplace, from social media to phone cameras. A face recognition system is designed to match human faces to digital images. Finally, pixel values ranging from 0-255 are distinguished by a computer. Face recognition has been used in computer vision for a long time and has evolved over time. Many new techniques for identifying and distinguishing faces have been developed by researchers. There are numerous use cases, such as user authentication and verification.

This article discusses all aspects of attendance systems based on facial recognition. It discusses the difficulties encountered in face recognition, the face recognition library, and the development of an attendance marking system based on these techniques.

# TABLE OF CONTENTS

CERTIFICATE  
ACKNOWLEDGEMENT  
ABSTRACT  
LIST OF FIGURES  
LIST OF TABLES  
LIST OF ABBREVIATIONS /SYMBOLS

CHAPTER 1: INTRODUCTION	
1.1: BACKGROUND	1
1.2: BENEFITS	1
1.3: SOFTWARES/TOOLS USED	3
1.4: OBJECTIVE OF THE PROJECT	3
CHAPTER 2: INTRODUCTION-PYTHON BASICS	
2.1: ABOUT	4
2.2: WHY IS PYTHON SO POPULAR?	4
2.3: PYTHON FEATURES	4
2.4: WHY DO WE NEED PYTHON	5
2.5: PYTHON INSTALLATION	5
2.6: PYTHON FIRST PROGRAM	6
CHAPTER 3: VARIABLES AND DATA TYPES IN PYTHON	
3.1: LISTS	9
3.2: TUPLES	10
3.3: SET	10
3.4: DICTIONARY	11
3.5: TYPE CONVERSION	14
CHAPTER 4: OPERATORS IN PYTHON	
4.1: ARITHMETIC OPERATORS	17
4.2: PYTHON ASSIGNMENT OPERATORS	18
4.3: PYTHON COMPARISON OPERATORS	18
4.4: PYTHON LOGICAL OPERATORS	19
4.5: PYTHON IDENTITY OPERATORS	19
4.6: PYTHON MEMBERSHIP OPERATORS	20
4.7: PYTHON BITWISE OPERATORS	20

CHAPTER 5: FLOW CONTROL	
5.1: IF STATEMENT	21
5.2: IF ELSE STATEMENT	22
5.3: ELIF STATEMENT	23
5.4: NESTED IF	25
CHAPTER 6: FUNCTIONS	64
6.1: INTRODUCTION TO FUNCTIONS	31
6.2: APPLICATIONS OF FUNCTIONS	32
6.3: DEFAULT ARGUMENTS	32
6.4: KEYWORD ARGUMENTS	33
6.5: PARAMETER PASSING IN PYTHON	34
6.6: GLOBAL VARIABLES IN PYTHON	34
CHAPTER 7: STRINGS	
7.1: ACCESSING CHARACTERS IN PYTHON	36
7.2: OPERATORS IN STRINGS	36
7.3: BUILT IN FUNCTION IN STRINGS	37
7.4: ESCAPE SEQUENCE	38
CHAPTER 8: LIBRARIES IN PYTHON	
8.1: WORKING OF PYTHON LIBRARY	39
8.2: PYTHON STANDARD LIBRARY	39
8.3: USE OF LIBRARIES IN PYTHON	40
8.4: IMPORTING SPECIFIC ITEMS FROM A LIBRARY MODULE	41
CHAPTER 9: NUMPY	
9.1: INSTALLATION	43
9.2: OPERATIONS USING NUMPY	43
9.3: DATA TYPES IN NUMPY	45
9.4: WHY IS NUMPY FASTER IN PYTHON	45
CHAPTER 10: DLIB	
10.1: INSTALLATION	46
10.2: WHAT IS DLIB USED FOR?	46
10.3: HoG FACE DETECTOR IN DLIB	47
10.4: CNN FACE DETECTOR IN DLIB	48

<b>CHAPTER 11: OPEN CV</b>	
11.1: INSTALLATION	49
11.2: WHAT IS OPEN CV USED FOR?	49
11.3: OPEN CV FUNCTIONALITY	50
11.4: ADVANTAGES	50
11.5: DISADVANTAGES	50
<b>CHAPTER 12: FACE RECOGNITION ATTENDANCE SYSTEM</b>	
12.1: INTRODUCTION	51
12.2: DIFFERENT TYPES OF ATTENDANCE SYSTEMS	51
12.3: MANUAL ATTENDANCE MANAGEMENT SYSTEM	52
12.4: DIFFERENT FACE RECOGNITION TECHNIQUES	52
12.5: ADVANTAGES	54
12.6: DISADVANTAGES	54
12.7: HOW FACIAL RECOGNITION IS USED	54
12.8: LIMITATIONS AND CHALLENGES	55
12.9: FUTURE OF FACE RECOGNITION ATTENDANCE	55
<b>CHAPTER 13: IMPLEMENTATION OF THE CODE</b>	
13.1: LANGUAGES USED	56
13.2: SOFTWARES USED	56
13.3: STEPS TO BE FOLLOWED	56
13.4: IMPORT LIBRARIES	57
13.5: EXECUTION OF THE CODES	57
<b>CHAPTER 14: MODEL ANALYSIS</b>	
14.1: ENTITY SET	72
14.2: ATTRIBUTES	72
<b>PROJECT PLANNING AND MANAGEMENT</b>	
GANTT CHART	
CONCLUSION	
REFERENCES	

## **LIST OF FIGURES**

<b>Fig no.</b>	<b>Description</b>	<b>Page No.</b>
3.1	LISTS IN PYTHON	9
3.2	TUPLES IN PYTHON	10
4.1	ARITHMETIC OPERATORS	17
4.2	ASSIGNMENT OPERATORS	18
4.3	COMPARISON OPERATORS	18
12.1	FEATURE BASED APPROACH	53
13.1	SNAPSHOT OF CODE (1)	57
13.2	SNAPSHOT OF CODE (2)	58
13.3	SNAPSHOT OF CODE (3)	58
13.4	SNAPSHOT OF CODE (4)	59
13.5	SNAPSHOT OF CODE (5)	59
13.6	SNAPSHOT OF CODE (6)	60
13.7	SNAPSHOT OF CODE (7)	60
13.8	SNAPSHOT OF CODE (8)	61
13.9	SNAPSHOT OF CODE (9)	62
13.10	SNAPSHOT OF CODE (10)	62
13.11	SNAPSHOT OF CODE (11)	63
13.12	SNAPSHOT OF CODE (12)	64

13.13	SNAPSHOT OF CODE (13)	64
13.14	OUTPUT AFTER CODE EXECUTION	65
13.15	USER ENTERS THE REQUIRE FIELD	66
13.16	TAKING IMAGES	66
13.17	IMAGES TAKEN SUCCESSFULLY	67
13.18	SNAPSHOT OF 100 IMAGES TAKEN	67
13.19	PROFILE SAVED SUCCESSFULLY	68
13.20	STUDENT'S DETAILS SAVED IN DATABASE	69
13.21	SYSTEM TAKING ATTENDANCE	69
13.22	IDENTIFY YOUR FACE WITH NAME	70
13.33	STUDENT'S DATA REGISTERED IN SHEET	71
13.24	ATTENDANCE SHEET	71
14.1	ENTITY SET	72
14.2	FLOWCHART DEPICTING WORKFLOW OF THE SYSTEM	74
15.1	GANTT CHART	76

## LIST OF TABLES

Table no.	Description	Page No.
5.1	TYPES OF IF STATEMENTS	21
7.1	BUILT-IN FUNCTIONS IN STRINGS	38
7.2	ESCAPE SEQUENCE IN STRINGS	38
9.1	DATA TYPES IN NUMPY	45
15.1	PROJECT PLANNING AND MANAGEMENT	75

## LIST OF ABBREVIATIONS/SYMBOLS

ABBREVIATION	DESCRIPTION
+	Addition operator
-	Subtraction operator
*	Multiplication operator
/	Division operator
%	Modulus operator
**	Exponent operator
=	Assignment operator
!=	Not equals to
==	Equal to
>	Greater than
<	Less than
&	Bitwise AND
	Bitwise OR
^	Bitwise XOR
<<	Left Shift
>>	Right Shift

# **CHAPTER 1:**

## **INTRODUCTION**

This chapter focuses with basic introduction of the entire project which includes the background, benefits and the software used to demonstrate our project.

### **1.1 BACKGROUND**

We are all privy to the modern-day pandemic scenario and the pandemic has pressured all people to be contact less with each other. Attendance is sort of obligatory everywhere be it instructional establishments or workplaces. Since the scenario isn't always at its everyday pace, hence taking attendance manually or thru biometric may be risky. Thus facial recognition is the maximum feasible alternative to be had in the modern scenario for agencies to make worker's attendance and tourist access contact less.

Touch-less structures are going to emerge as the brand new norm as corporations get their workplaces COVID equipped.

Businesses that were using fingerprint biometric or any touch-based totally systems for time, attendance, and traveller control are going to should recollect switching to contact less structures like face recognition devices.

### **1.2 BENEFITS**

Facial recognition is one of the few biometric technologies that is gaining traction. Facial recognition technology can help with identification, tracking, permissions, indexing, and maintenance of applications, among other things.

Face recognition technology has advanced, allowing for easier access to multimedia information. Furthermore, using face recognition to regulate network access makes it practically not possible for cyber criminals to obtain a user's password while simultaneously improving human-computer connection. One of the reasons why facial recognition attendance systems are becoming more popular is because of this. Some of the benefits of the technology is discussed below:

- **Automatic time tracking system :** Automated attendance management systems can completely automate the monitoring of entry and exit times that would otherwise be done by hand and with other biometric systems. Because the system's powerful algorithms can autonomously find and identify faces, no human intervention or physical validation is required. With facial recognition, it is simple to track employee time.
- **Cost-effective:** By automating employee time tracking, a facial recognition attendance system can save businesses money. True in, for example, Small and medium-sized organisations will find it more economical because it can be used on smart phones. An attendance system, regardless of its size of the company, can:

- i. An improvement in staff productivity of 10%
- ii. Cut administrative costs by 5% to 10%.
- iii. Reduce supervisory time by 15% while supporting supervisors with attendance management.
- iv. Because the data from the facial recognition-based employee attendance system is genuine and precise, the cost savings are even larger.

- **Time saving:** COVID19 is wreaking havoc over the globe, and it's probably time to pay attention to the social alienation. In today's environment, maintaining a good distance from everyone else has become a requirement. This can be a problem if you use a manual attendance system. A face recognition-based attendance management system would not only enable you to log a person's attendance, but it will also allow you to work remotely while still being able to see who is entering and exiting. This stresses how much secure, time-saving, and speedier the entire approach is for documenting attendance.
- **Increased security:** A face recognition-based attendance system will not only calculate attendance but will also record the entrances and exits of visitors. When a specific man's identity and timing of entry and exit must be recorded, this system will come in useful because it can simply show you when he or she came in and where he or she went to a very accurate level. All of this translates to a significantly higher level of employment security for you.
- **Easy to manage:** Because the keeping records & track of everyday activities will be much easier with an artificially intelligent attendance system than with a manual approach. Everything will be completed by the system. Many software packages can show how so many hours or minutes a person spent at his or her desk during the day. On a vast scale, all of this is conceivable. Imagine recording the activities of a large crowd of 200 people in real time and in an organized manner! Face recognition is an example of AI's power.
- **Smart Integration:** It is relatively simple to integrate any other HRMS or payroll system with a facial recognition attendance system. The entry, exit, and date formats can be changed to be consistent with other systems in an organisation because these systems are flexible and readily customised. It considerably simplifies the data organisation procedure. Furthermore, time zone settings can be readily altered based on location, allowing software to be utilised globally without the need for further configuration. A GPS-based attendance system can be used by a corporation with geographically spread offices to track attendance in all of its locations.
- **Compatibility with mobile devices :** Front-facing cameras are standard on nearly all smartphones, tablets, and laptops. This means that a facial recognition attendance system does not require any additional hardware to be implemented. In comparison to other biometric systems, such as fingerprint scanners, this is more affordable and convenient. There is no need for training or orientation for work-from-home employees because all employees are already accustomed to using the front-facing camera on their mobile device. These systems have simple user interfaces that are simple to use for anyone.

### **1.3 SOFTWARES/TOOLS USED**

Following are the softwares/tools used for building the entire project:

- **Visual Studio Code(VS Code)**-Code editor used for editing
- **Tkinter**- For whole GUI
- **OpenCV**- For taking images and face recognition  
(cv2.face.LBPHFaceRecognizer\_create())
- **Others**- CSV,NumPy,Pandas,date time,dlib etc.

### **1.4 OBJECTIVE OF THE PROJECT**

To execute Student participation framework dependent on the face acknowledgment of webcams picture in the study hall and store the data in data set.

The target of this undertaking is to foster face acknowledgment based mechanized understudy participation framework. Anticipated that achievements in order should satisfy the targets are:

1. To recognize the face section from the video outline.
2. To concentrate the valuable elements from the face distinguished.
3. To group the elements to perceive the face distinguished.
4. To record the participation of the distinguished understudy.

## CHAPTER 2:

### INTRODUCTION-PYTHON BASICS

This chapter focuses on the basic fundamentals of the Python Language such as its features, need, and installation steps.

#### **2.1 ABOUT:**

The Python programming language is often used for building websites, automating processes<sup>73</sup> and analyzing data. General-purpose programming languages, such as Python, can be used to design a wide range of applications, making them unspecialized for any particular kind of problem. The language has become one of the most widely used programming languages today due to its functionality and beginner-friendliness. Unlike other programming languages, Python is a dynamic, high-level, and interpreted programming language. To improve applications, it supports Object Oriented Programming Software. It's straightforward and straightforward to read, with high-quality data structures. Python is a simple-to-understand but powerful and flexible programming language, which makes it ideal for application development. Python's syntax, dynamic typing, and translation environment make it an excellent language for rapid application development and writing.

#### **2.2 WHY IS PYTHON SO POPULAR?**

*Python's popularity is due to a variety of factors. Here's a closer look at what makes it so useful to programmers.*

It features a straightforward syntax similar to standard English, making it easy to read and understand. This accelerates the development of projects as well as the development of existing ones. It is flexible. Python can be used for a variety of purposes, including web development and machine learning. It is easy to use, making it popular among new editors. It is free to use and distribute, even commercially, because it is an open source. The Python module and the library archive — piles of code generated by third-party users to extend Python capabilities — are large and growing. Python has a strong community that adds to the module library and libraries and serves as a resource for other programmers. Due to the large support network, finding a stumbling block solution is very easy; someone has probably had a similar problem before.

#### **2.3 PYTHON FEATURES:**

<sup>44</sup> Python provides many useful features for the editor. These features make it very popular and are the most widely used language. Listed below are a few key features of Python.

- Extensible
- Object-Oriented Language
- Embeddable

- Learn Standard Library
- Integrated
- Dynamic Memory Allocation
- Interpreted Language
- GUI Programming Support
- Open Source Language

## **2.4 WHY DO WE NEED PYTHON?**

23

Python is a programming language that is extensively used for online & application design, task automation, data analysis, and visualisation. Python is reasonably straightforward to learn, thus many non-programmers, such as accountants and researchers, have adopted it for a variety of tasks, including money management.

### **Data analysis and machine Learning :**

Python has now become an advanced analytics standard, permitting data analysts and other experts to use the language to execute complex statistical computations, develop data visualisations, build machine learning techniques, manage data, and conduct other data-related tasks.

### **Web development:**

Python has become the industry standard for data scientists, allowing them to do complex statistical computations, build data visualisations, create machine learning algorithms, handle and analyse data, and execute other data-related tasks can.

### **Automation or Scripting:**

If you come across a repetitive task, you can automate it with Python to make it more efficient. The technique of programming code to implement these automated processes is known as scripting.. The coding industry can use automation to check many files for errors, convert files, perform simple math operations, and remove duplicate data.

## **2.5 PYTHON INSTALLATION:**

### **Downloading installing Python:**

1. Go to [www.python.org/downloads/](http://www.python.org/downloads/)
2. Check your system requirement and download Python according to it.

### **Installing python on windows:**

1. Click on Python for Windows, then download the latest release.
2. Scroll to the bottom and select either the 64-bit or 32-bit installer as per your system requirements.

### **Installing python on Linux**

1. Go to the Ubuntu Software Center folder and double-click it.
2. From the All Software drop-down list box, select Developer Tools.
3. Select Python 3.3.4 and double-click it.
4. Select Install.
5. Delete the Ubuntu Software Center folder from your computer.

### **2.6 PYTHON FIRST PROGRAM:**

Unlike other programming languages, Python allows you to run code on multiple lines. We can do this using one statement in Python.

```
1 print("Hello, World!")  
  
Hello, World!  
[Finished in 0.2s]
```

## CHAPTER 3:

### VARIABLES AND DATA TYPES IN PYTHON

This chapter focuses on the different kinds of variables in Python Language and its uses with examples.

#### Variables in Python:

- 2 ● A variable is a term that is used to refer to a memory region. Values are stored in Python variables, commonly known as identifiers.
- You don't have to indicate the type of a variable in Python. Because Python is an inference language, it is capable of determining variable types.
- Letters or numbers can be used as variable names, but they must begin with a character or underscores.

#### Example:

Python Variables	
x = 45	Type = Integer 45 x
name = "DataFlair"	Type = String "DataFlair" name
nums = [ 1, 8.5, 9 ]	Type = Lists [1, 8.5, 9] nums

#### Identifier in Python:

Variables are examples of identifiers. Identifiers are used to identify literals used in programs.

#### Rules:

- 2 1. The first character of a variable must be a character or an underscore (\_).
- 2 2. All letters except the first letter can be lowercase (az), uppercase (AZ), underscore, or digit (09) alphabets.
- 3 3. The identifier name cannot contain spaces or special characters (!, @, #, %, ^, &, \*).
- 4 4. Identifier names must not be similar to keywords defined in that language.
- 5 5. Examples of valid identifiers: a123, \_n, n\_9, etc.
- 6 6. Invalid Identifier Example: 1A, N% 4, N 9, etc.

#### 3 Data Types in Python:

- Variables can hold values, and each value has a data type. Python is a dynamically typed language. Therefore, you do not need to define

the variable type when you declare the variable. The interpreter implicitly binds the value to that type.

- Variable a contains the integer value 5, and its type is undefined. The Python interpreter automatically interprets the variable a as an integer.
- Python allows you to see the types of variables used in your program. Python provides a type () function that returns the type of the passed variable.

```
a = 5
```

### Python Data Types:

**1) Numbers:** Number is a type of data that stores numerical values. Numeric data types in Python include integers, floating points, and complex numbers.

a) Python supports 3 types of numeric data.

- i) **Integer** - Integer value can be of any length such as 92, 4, 76, -44, 63 etc.
- ii) **Float** - Float data type is used to store floating-point numbers like 3.2, 5.427, 38.26, etc.
- iii) **Complex number** - A complex number contains a real and an imaginary part. For example,  $x + iy$  is a complex number where  $x$  denotes the real part and  $y$  denotes the imaginary part.

**2) Sequence Type:**

i) **String:** A string is defined as a collection of quoted characters. To define a string in Python, you can use ', ", or triple quotations.

ii) **Lists:** are analogous to C arrays in Python.

The list, on the other hand, might contain a variety of facts. Commas (,) separate the items in the list, which are enclosed in square brackets [].

iii) **Tuple:** Tuples are comparable to lists in many ways.

Tuples, like lists, are made up of components of various data kinds.

Commas (,) are used to divide multiple items, which are then enclosed in parenthesis ().

**3) Boolean:** There are two built-in Boolean values: True and False. These values are used to determine whether or not the provided statement is true. This is denoted by the bool class. A non-zero value or "T" can be used to represent true, while 0 or "F" can be used to represent false.

**4) Set:** A Python set is a collection of data types that is not ordered. It is mutable (changeable after construction) and has its own elements. The arrangement of the elements is not determined by the set. You can get the elements' modified arrangement back.

**5) Dictionary:** A dictionary is a collection of element key/value pairs that are not in any particular sequence. It's similar to an ordered sequence or hash table, with each key storing a unique value. Any primitive data type can be used as the key, and any Python object can be used as the value.

### **3.1 LISTS:**

Lists are the same as dynamically sized arrays declared in other languages (C ++ Vector and Java ArrayList).

The list does not necessarily have to be the same, making it one of the most powerful tools in Python.

A single list can contain data types such as integers, strings, and objects. The list can be modified, so you can modify it even after it is created.

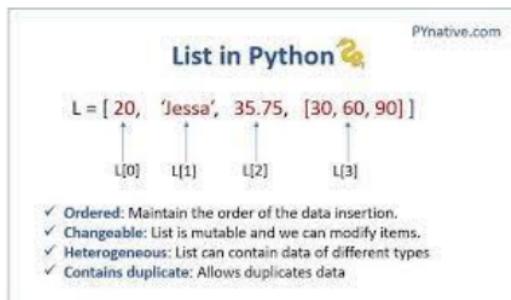
The list can be defined as a collection of different types of values or elements. The entries in the list are separated by commas (,) and enclosed in square brackets [].

#### **Characteristics of Lists:**

The following traits can be found on the list:

- The list is arranged in alphabetical order.
- An index can be used to retrieve list elements. The lists are changeable in nature.
- A list can contain a variety of different items.

A list can be define as below:



**Figure 3.1** Lists in Python

### **3.2 TUPLES:**

A succession of unchanging Python objects is stored in a Python tuple. Tuples are similar to lists in that the values of the items in the list can be changed. but the tuples are immutable and you cannot change the values of the items stored in the tuple.

#### **Creating a tuple:**

2

A tuple can be described as a collection of comma-separated (,) values enclosed in small () square brackets. Parentheses are optional, but we recommend that you use them.

Tuples can be defined as:

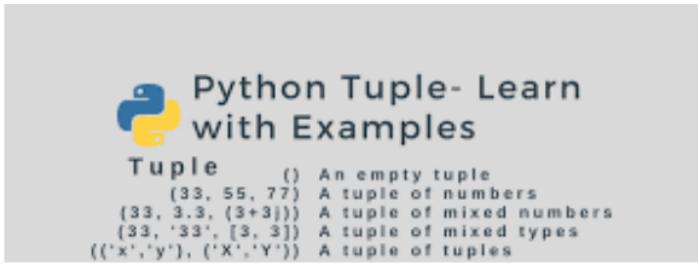


Figure 3.2 Tuples in Python

### **3.3 SET:**

6

A Python set is a collection of elements that are not in any particular sequence. Each element inside the set must be unique and immutable, and duplicate elements are removed. It is possible to alter the set. That is, after it has been built, you can edit it.

The items of the set, unlike other Python collections, are not indexed, which means you can't access them directly by index.

You can, however, publish them all at once or loop through the set to generate a list of items.

#### **Creating a set:**

Curly braces can be used to enclose comma-separated immutable elements in sets. You may also use Python's set () function to build a set with the provided sequence.

#### **Example 1: Using curly braces**

```
# Set In python using Curly Braces

Days = {"Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"}

print(Days)
print(type(Days))
print("looping through the set elements ... ")
for i in Days:
    print(i)
```

### Example 2: Using set() method

```
# Set In python using Set Method

Days = set(["Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"])
print(Days)
print(type(Days))
print("looping through the set elements ... ")
for i in Days:
    print(i)
```

### 3.4 DICTIONARY:

Data in the form of key/value pairs is stored in a Python dictionary. A dictionary is a Python data type that may be used to emulate the actual data location where a certain value exists for a specific key. This is an example of a varying data structure. The element's key and value are used to define the dictionary.

- A single element must make up the key.
- Any sort of value can be used. Lists, tuples, integers, and other data types

#### Creating the dictionary:

Multiple key/~~val~~<sup>ue</sup> pairs surrounded in curly brackets can be used to build the dictionary, with each key separated from the value by a colon (:). The following is the syntax for creating a dictionary.

#### Syntax:

```
dictionary = {"Name": "Tom", "Age": 23}
```

In the dictionary above, the key name and age are strings, immutable objects. Let's look at an example of creating a dictionary and printing its contents.

#### **Example:**

```
app.py > ...
1 dict = {'Africa':200,'australia':300,'England':400}
2 print("Country Name",dict)
3 dict.update([('China':500)])
4 print("updated country name",dict)
5
6
```

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

arvind@arvind:~/Documents/PythonProject\$ python3 app.py
Country Name {'Africa': 200, 'australia': 300, 'England': 400}
updated country name {'Africa': 200, 'australia': 300, 'England': 400, 'China': 500}
arvind@arvind:~/Documents/PythonProject\$



### Accessing the dictionary values:

We have described how to use indexes to access list and tuple data.

However, the key is unique in the dictionary, so you can use the key to access the value in the dictionary.

You can access the dictionary values as follows:

#### Code:

```
emp = {"Name": "YADNYESH", "Age": 32, "salary":47000}
②int(type(emp))
print("printing Employee data .... ")
print("Name : %s" %emp ["Name"])
print("Age : %d" %emp ["Age"])
print("Salary : %d" %emp ["salary"])
print("Company : %s" %emp ["Company"])
```

#### Output:

```
<class 'dict'>
printing Employee data ....
Name : YADNYESH
Age : 32
Salary : 47000
>
```

### Adding dictionary values:

A dictionary is a variable data type whose value can be changed by using a unique key. Along with the key Dict[key] = value, the value can be modified.

Existing values can also be updated using the update() method.

The value will be updated if the key value already exists in the dictionary. Otherwise, the dictionary will be updated with the new key. Let's have a look at how to modify the dictionary value with an example.

#### Programme:

```
1. # Initialise aDictionary
2. ④ct = {}
3. print("Empty Dictionary: ")
4. print(dict)
5.
6. # Adding dictionary elements one by one
7. Dict[0] = 'Joe'
8. Dict[2] = 'Ross'
9. D⑥[3] = 'Michael'
10. print("\nDictionary after adding 3 elements: ")
```

```

11. print(Dict)
12.
13. # Adding set of values with a single Key
14. # The emp_age doesn't exist inside dictionary
15. Dict['emp_age'] = 31, 24, 38
16. print("\nDictionary after adding 3 elements: ")
17. print(Dict)
18.
19. # Updating the existing Value of a Key
20. Dict[3] = 'Python'
21. int("\nUpdated key value: ")
22. print(Dict)

```

#### Output:

```

Empty Dictionary:
{}

Dictionary after adding 3 elements:
{0: 'Joe', 1: 'Ross', 2: 'Michael'}
Dictionary after adding 3 elements:
{0: 'Joe', 1: 'Ross', 2: 'Michael', 'emp_age': (31, 24, 38)}
Updated key value: {0: 'Joe', 1: 'Ross', 2: 'Python', 'emp_age': (31, 24, 38)}

```

#### 3.5 TYPE CONVERSION:

Typecasting is a method of casting a variable data type to a specific data type so that the user can perform the operation. This article describes different types casting techniques.

There can be two types of Type Casting in Python –

- Implicit Type Casting
- Explicit Type Casting

#### Implicit Type Casting:

Python will automatically change one data type to another using this function. The user is not required to participate in this procedure.

Program:

```

# Python program to demonstrate
# implicit type Casting

# Automatic conversion of a to int
a = 9
print(type(a))

# Automatic conversion of b to float
b = 7.0
print(type(b))

```

```
# Automatic conversion of c to float
# as it is a float addition
c = a + b
print(c)
print(type(c))
```

**Output:**

```
<class 'int'>
<class 'float'>
10.0
<class 'float'>
```

**Explicit Type Casting**

In order to complete the required operation, Python requires user participation to change the varying data type to a specific data type. This data type function is primarily intended for typecasting.

Int() accepts a float or string as an input and returns an object of type int.  
float() takes an int or a string as a parameter and generates a float type object.  
str() generates a string type object when given a float or int as an argument.

**Program:**

```
#program to demonstrate explicit type conversion
#initializing the value of a
a=10.6
print("The type of 'a' before typecasting is ",type(a))
print(int(a))
print("The type of 'a' after typecasting is",type(a))
#initializing the value of b
b=8.3
print("The type of 'b' before typecasting is ",type(b))
print(int(b))
print("The type of 'b' after typecasting  is",type(b))
#initializing the value of c
c=7
print("The type of 'c' before typecasting is ",type(c))
print(float(c))
print("The type of 'c' after typecasting is",type(c))
```

**Output:**

Shell Clear

```
The type of 'a' before typecasting is <class 'float'>
10
The type of 'a' after typecasting is <class 'float'>
The type of 'b' before typecasting is <class 'float'>
8
The type of 'b' after typecasting is <class 'float'>
The type of 'c' before typecasting is <class 'int'>
7.0
The type of 'c' after typecasting is <class 'int'>
> |
```

## CHAPTER 4: OPERATORS IN PYTHON

2  
An operator is a symbol that is responsible for performing a specific operation between two operands. Python has a number of operators, which are listed below: An operator is a symbol that is responsible for performing a specific operation between two operands. Python has a number of operators, which are listed below:

- o Assignment Operators
- o Comparison Operators
- o Identity Operators
- o Logical Operators
- o Bitwise Operators
- o Arithmetic Operators
- o Membership Operators

### 4.1. Arithmetic operators:

Arithmetic operators are used with numbers to calculate common mathematical operations

## Arithmetic Operators

a = 5  
b = 2

Operator	Meaning	Example	Result
+	Addition Operator. Adds two Values.	a + b	7
-	Subtraction Operator. Subtracts one value from another	a - b	3
*	Multiplication Operator. Multiples values on either side of the operator	a * b	10
/	Division Operator. Divides left operand by the right operand.	a / b	2.5
%	Modulus Operator. Gives remainder of division	a % b	1
**	Exponent Operator. Calculates exponential power value. a ** b gives the value of a to the power of b	a ** b	25
//	Integer division, also called floor division. Performs division and gives only integer quotient.	a // b	2

Figure 4.1 Arithmetic operators

58

#### **4.2. Python Assignment Operators:**

Assignment operators are used to assign values to variables:

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x  = 3	x = x   3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

**Figure 4.2** Assignment Operators

65

#### **4.3. Python Comparison Operator:**

Comparison operators are used to compare two values:

Python Comparison Operators		
Operator	Name	Example
==	Equal	a == b
!=	Not equal	a != b
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b

54

**Figure 4.3** Comparison Operators

#### **4.4. Python Logical Operators:**

Logical operators are used to combine conditional statements:

Operator	Meaning	Example	Result
and	Logical and	(5<2) and (5>3)	False
or	Logical or	(5<2) or (5>3)	True
not	Logical not	not (5<2)	True

OPERATION	RESULT	NOTES
X or Y	If X is False, then Y, else X	Y is executed only if X is False Else if X is true, X is result.
X and Y	If X is false, then X else Y	Y is executed only if X is true, else if X is false , X is result.
not X	if X is true, then false, else true	not has lower priority than non - boolean operators. Eg. not a==b => not (a==b)

#### 4.5. Python Identity Operators:

29

The Identity operator is used to compare objects when they are actually the same object in the same location, not when they are the same.

##### Python Identity Operators :



Identity Operators are used to check the address reference of two variable is same or not.

Operator	Description
is	It is evaluated to be true if the reference present at both sides points to the same object. a = 10 , b = 10 , a is b returns true.
is not	It is evaluated to be true if the reference present at both side do not point to the same object. a = 10 , b = 20 a is not b returns true

**29**  
**4.6. Python Membership Operators:**

Membership operators are used to test whether a sequence is represented by an object.

Operator	Meaning	Example
In	True if value/variable found in the sequence	5 in x
Not in	True if value/variable is not found in sequence	5 not in x

**4.7. Python Bitwise Operators:**

**Table 4.1** Types of Bitwise Operators

**Types of Bitwise Operators**

Operator	Name	Example	Result
&	Bitwise AND	6 & 3	2
	Bitwise OR	10   10	10
^	Bitwise XOR	2^2	0
~	Bitwise 1's complement	~9	-10
<<	Left-Shift	10<<2	40
>>	Right-Shift	10>>2	2

## CHAPTER 5: FLOW CONTROL

In reality, there are situations in which we have to make decisions, and based on those decisions we decide what to do next. In programming, a situation similar to this emerges when you must make some judgments and then execute the next block of code based on those decisions. Programming language decision statements determine the direction of program execution flow.

Statement	Description
If Statement	It is used to determine whether or not a certain condition exists. A piece of code (if-block) is executed if the condition evaluates to true.
If-else Statement	It is identical to the if statement, but it also includes a code block for the condition under test that is incorrect. The else statement is executed if the condition stated in the if statement is untrue.
Nested if Statement	We can utilise an if else statement within an outer if statement with nested if statements.

**Table 5.1** Types of if statements

### Indentation in Python:

In Python, you cannot use parenthesis in block-level code to simplify programming and accomplish simplicity. Indentation is used to declare blocks in Python.

- Two statements that are indented at the same level are considered to be part of the same section.
- An indent statement is usually given four spaces. In Python, this is a standard amount of indentation.
- Because it declares code blocks, indentation is the most often utilised feature of the Python language.
- The indentation level for all statements in the block should be the same. Let's take a look at how indentation is implemented in Python choices and other places.

### **5.1 If Statement:**

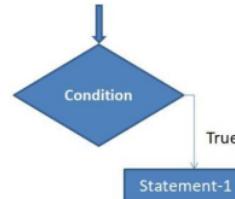
If a condition is true, the if statement executes an if-block block of code. Any acceptable formula that responds to either true or false can be used as the if statement's condition.

### Syntax and Flowchart:

Syntax-  
**if**(condition):  
    statement

Where:

- if** : Keywords
- Condition** : It may be any valid python arithmetic, Logical or relational expression
- Statement** : Valid Python statements that executes when condition in **if** statement is **True**



### Example:

```
a = 33  
b = 200  
if b > a:  
    print("b is greater than a")
```

### Output

```
"b is greater than a".
```

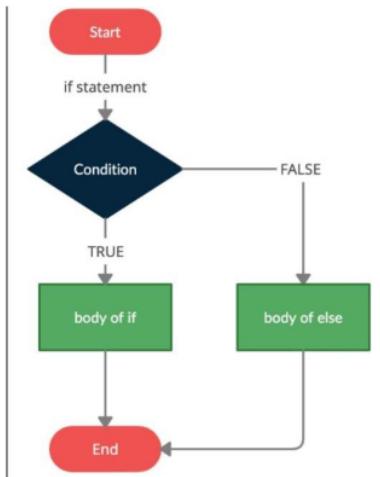
### 5.2 If-else Statement:

If the condition is false, the if-otherwise statement gives an else block that is run in conjunction with the if statement. The if block will be executed if the condition is true. The else block will be invoked if the first block is not executed.

### Syntax:

```
if test expression:  
    Body of if  
else:  
    Body of else
```

**Flowchart:**



**Example:**

A screenshot of a Python development environment. The code editor window shows a file named "example.py" with the following content:

```
example.py
1
2 num = 30
3 if num > 10:
4     print("True")
5 else:
6     print("False")
7
```

Below the code editor is a terminal window titled "Shell" with the following output:

```
Shell>>> %Run example.py
True
>>>
```

**5.3 2 elif statement:**

The **elif statement** can be used to verify various conditions and then execute a specific block of the statement based on the true conditions. You can use as many elif statements as you need in your programme. Using elif is, however, optional. The elif statement functions similarly to a C ifelseif ladder statement. It has to be followed by an if condition.

45

**Syntax:**

if test expression:

    Body of if

elif test expression:

    Body of elif

else:

    Body of else

**Flowchart:**

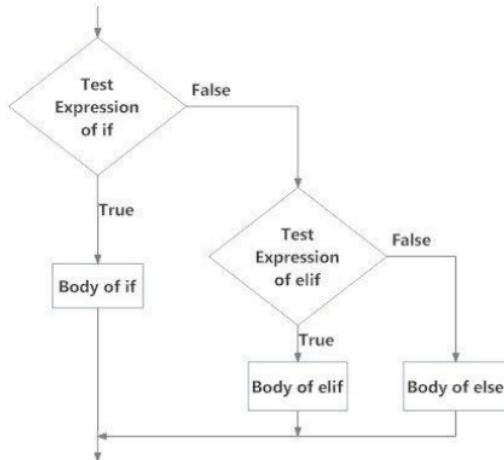


Fig: Operation of if...elif...else statement

**Example:**

20

"In this program,  
we check if the number is positive or  
negative or zero and  
display an appropriate message"

num = 3.4

# Try these two variations as well:

# num = 0  
# num = -4.5

```
if num > 0:  
    print("Positive number")  
elif num == 0:  
    print("Zero")  
else:
```

```
print("Negative number")
```

#### OUTPUT:

Positive number is printed when variable num is positive.

If num equals zero, nothing is printed.

Negative number is printed if num is negative.

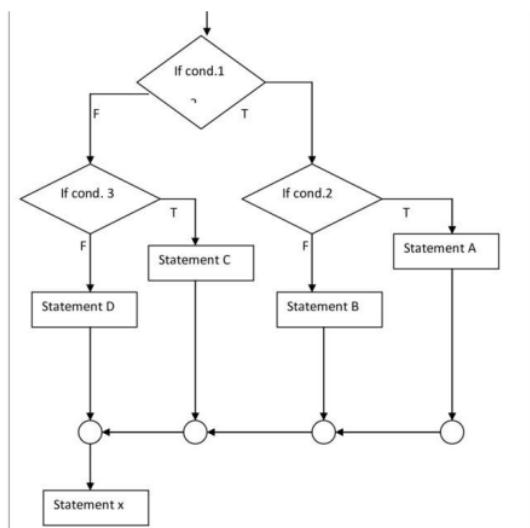
#### **5.4 Nested If:**

57 A nested if is an if statement that is the target of another if statement. A nested if statement means an if statement within another if statement. Yes, Python allows if statements to be nested within if statements. That is, you can put an if statement inside another if statement.

#### Syntax:

```
if (condition1):
    # Executes when condition1 is true
    if (condition2):
        # Executes when condition2 is true
        # if Block is end here
    # if Block is end here
```

#### Flowchart:



Program:

```
Python112.py x
1  #
2  # Example file for working with conditional statement
3  #
4  def main():
5      x, y = 8, 4
6
7      if (x < y):
8          st = "x is less than y"
9      else:
10         st = "x is greater than y"
11         print(st)
12
13  if __name__ == "__main__":
14      main()
15
```

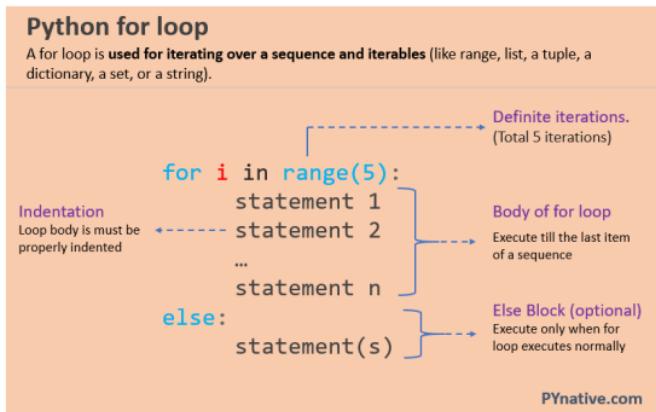
Run Python11.2  
C:\Users\DK\Desktop\Python code\Python Test  
x is greater than y

Use "else condition", if there is any other outcomes you want to print out in case your "if condition" does not gives the expected result

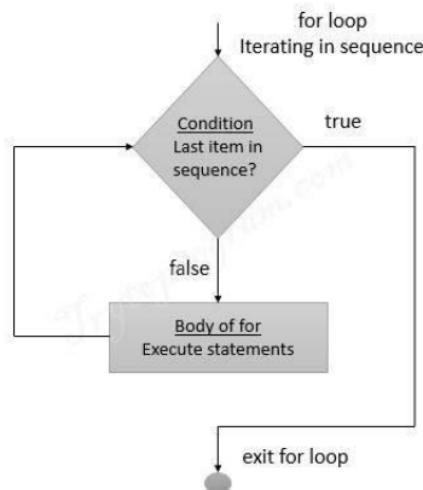
## CHAPTER 6: LOOPS

Python for loops are used to repeatedly execute a statement or a section of a programme. It's frequently used to iterate across data structures like lists, tuples, and dictionaries.

The syntax of the Python for loop is shown below.



### Flowchart:



39

### 6.1. For loop Using Sequence:

#### **Example-1: Iterating string using for loop**

Program:

```
In [4]: my_list = "Hello World!"  
for i in my_list:  
    print(i)  
  
H  
e  
l  
l  
o  
  
W  
o  
r  
l  
d  
!
```

**Example- 2: Program to print the table of the given number.**

Program:

```
list = [1,2,3,4,5,6,7,8,9,10]  
n = 5  
for i in list:  
    c = n*i  
    print(c)
```

Output:

```
5  
10  
15  
20  
25  
30  
35  
40  
45  
50
```

## **6.2. For loop Using range() function:**

To create a sequence, use the range () function. Passing range (10) will result in a value ranging from 0 to 9.

The range () function's syntax is illustrated below.

### **Syntax:**



```
range(start,stop,step size)
```

### **Program:**

```
52 Generate numbers between 0 to 7
for i in range(7):
    print(i)
```

### **Output:**

```
0
1
2
3
4
5
6
```

- The beginning of the iteration is represented by the start.
- The stop indicates that the loop will continue until it reaches stop-1. Iterations 1 to 4 will be generated using range(1,6). It's a choice.
- The step size is used to omit the iteration's specific numbers. It's entirely up to you whether or not you want to use it. The step size is set at 1 by default. It's a choice.

### ● 6.3. Nested for loop:

Any amount of for loops can be nested within a for loop in Python. For each iteration of outer loop, the inner loop is called n times. Here's how to write it:

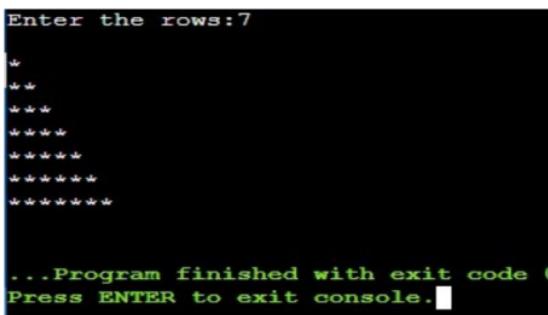
#### Syntax:

```
16  
# outer for loop  
for element in sequence  
    # inner for loop  
    for element in sequence:  
        body of inner for loop  
    body of outer for loop
```

#### Example:

```
1.# User input for number of rows  
2. rows = int(input("Enter the rows:"))  
3. 16Outer loop will print number of rows  
4. for i in range(0,rows+1):  
5. # Inner loop will print number of Astrisk  
6. for j in range(i):  
7. print("*",end = "")  
8. print()
```

#### Output:



A terminal window showing the execution of a Python script. The user enters "7" when prompted "Enter the rows:". The program then prints a triangle of asterisks, with 1 row on line 1, 2 rows on line 2, and so on up to 7 rows on line 7. Finally, it prints "Program finished with exit code 0" and "Press ENTER to exit console.".

```
Enter the rows:7  
*  
**  
***  
****  
*****  
*****  
*****  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

## CHAPTER 6: FUNCTIONS

14

A function is a piece of code that can be reused to execute a single, related task. Functions boost the functionality of your application and allow you to reuse a big amount of code..This chapter focuses mainly about the functions used in python language and also its various type.

### 6.1 INTRODUCTION TO FUNCTIONS

50

In computer programming, a function is a specified chunk of code that performs a certain task. Typically, this requires taking some data, altering it, and then providing a result.. We'll go through how to build python functions, how to define functions with an arbitrary figure of positional parameters, and how to describe functions with an arbitrary figure of keyword arguments in this context.

Syntax: def func(parameters):  
    """string"""  
    statement(s)

As seen above, a function definition is made up of the following components.

5

I.The keyword def indicates the beginning of a function header.

II.The function's name, which could be used to identify it from others. In Python, the same rules apply to identifying function because they do to naming identifiers.

III.We use parameters to pass values to a function (arguments). They aren't necessary.

IV.A comma marks the end of the function header (:).

V.Documentation strings which can be used to describe the function's functionality.

6

VI.The body of the function consists of one and more valid Python statements. All statements must have the same indentation level (usually 4 spaces).

69

VII.To return a value from a function, use the return statement.

Example: def project():  
    print ("This is my major project")

## **6.2 APPLICATIONS OF FUNCTIONS**

Most programming languages require the use of functions. Functions are reusable chunks of code that can be accessed by using the name of the function. In a Python program, functions can be called from anywhere, including from within other functions.

Using functions has a number of advantages:

- You can use functions to run the same code several times.
- Using functions, long programmes are split down into smaller components.
- Functions can be used and shared by other programmers.

A function name is assigned to each function. When a function is called in a programme, the name of the function is utilized. When you call a function, you're telling it to do something. The programme can provide input to functions. Input arguments, or simply arguments, are the arguments submitted to a function. Arguments are the pieces of code that are supplied as input to a function. Output is something that functions can do. The output of a function is said to be returned to the programme. A function's output can be allocated to a variable and then used in a programme. Let's have a look at the numerous benefits of using functions in Python.

## **6.3 DEFAULT ARGUMENTS**

Function parameters in Python can have default values. If the function is invoked without the need for an argument, the parameters are set to its value equal. In Python, function parameter syntax and default values are represented differently. The default value for the function parameter will be utilised if no argument value is provided during the function call. The default value is set using the assignment(=) operator with the form keyword name=value.

Let's take a look at it from the standpoint of a function student. The function student has 3 parameters, two of which are supplied with default values. As a result, the student function only accepts one necessary argument (first name) and two optional arguments.

Example: def student(first name, last name ='John', class ='Sixth'):

```
print(first name, last name, 'studies in', standard, 'class')
```

When calling functions, keep the following pointers in mind:

- 1)The order of the inputs is critical when giving keyword parameters.
- 2)There should be only one value for each parameter.

3)The keyword name that is passed in must match the actual term name.

4)The sequence in which non-keyword parameters are passed to a function is critical.

A function's default value can be set for any number of arguments. If we have a default argument, we must also have default values for all of the arguments to the right of it.

Non-default arguments, in other words, cannot be followed by default arguments.

## 6.4 KEYWORD ARGUMENTS

Keyword arguments (also known as named arguments/kwarg<sup>\*\*</sup>) are values that are identified by specific parameter names when supplied into a function. A parameter and the assignments operator, =, come before a keyword argument.<sup>17</sup>

In the way that they bind a value to a keyword, keyword arguments are similar to dictionaries.

When we call a function with some arguments, the arguments are allocated these values based on their position. In Python, keyword arguments can also be used to call functions. When invoking functions in this way, the sequence (position) of the arguments might be changed. Following that, all subsequent calls to the above function are valid and produce the same result.

\*\*kwargs is a special syntax for passing a keywords, variable-length argument list in Python function definitions.

Because of double star, it's called kwargs. This is because the double star allows us to pass keyword arguments through (and any number of them).<sup>1</sup>

You give a variable a name when you send it as a keyword argument to a function. The kwargs serve as a dictionary, mapping each term to the value we pass along with it.

**Example:** def function(\*\*kwargs):  
    for key, value in kwargs.items():  
        print ("%s == %s" %(key, value))

```
# Driver code  
function(first ='Project', mid ='on', last='Python')
```

**Output:** last == Python  
mid == on  
first == Project

## **6.5 PARAMETER PASSING IN PYTHON**

A function or process typically requires some understanding of the area in which it is used. The link between the area from which the function was called and the function body, i.e. the function body, is made up of special variables called parameters. These parameters can be used to bring any type of object from "outside" into a function. The syntax for defining variables and the semantics for sending arguments to the parameters of a function or procedure are determined by the programming language.

The terms parameter and argument are frequently used interchangeably, although there is a distinction to be made. Arguments, or the values provided to a procedure, are used in procedure calls, whereas parameters are found inside functions or procedures.

Python employs a technique known as "Call-by-Object," sometimes known as "Call by Object Reference" or "Call by Sharing." Passing immutable arguments to a function, such as integers, strings, or tuples, acts like Call-by-Value. If we pass mutable parameters, the situation changes. In the All parameters (arguments) in the Python programming language are supplied by reference. It means that if you change a parameter's value inside a function, the change is updated in the calling function as well.

## **6.6 GLOBAL VARIABLES IN PYTHON**

36

In Python, a global variable is a variable defined outside of the function or in the global scope. A global variable can all be accessible from both in or out of the function.

Example:

```
def f():
    print("Internal", s)
    # Global scope
    s = "This is my project"
    f()
    print("External", s)
```

Output: Internal This is my project  
External This is my project

1

If we wish to conduct assignments or alter the global variable, we must utilise the global keyword in the function. Printing and accessing do not require global. Because the assignment to s inside of f() causes Python to "think" that we want a local variable, the first sentence returns an error. If a variable isn't declared as a global variable and is updated or created within a function, it's considered local. The term "global" must be used to tell Python that we wish to use the global variable, as shown in the example below:

```
1 # This function modifies the global variable 's'
def f():
    global s
    s += ' Major Project'
    print(s)
    s = "Project On Python"
    print(s)

# Global Scope
s = "Project is done"
f()
print(s)
```

Output: Project is done Major Project  
Project On Python  
Project On Python

# CHAPTER 7:

## STRINGS

In Python, strings are byte arrays that represent unicode characters., like many other  
59 widely used programming languages. In Python, however, there is no such thing as a  
A single character is a one-length string of the character data type. To get to the  
elements in the string, use square brackets.

5  
In Python, single quotes, double quotes, and even triple quotes can be used to generate strings.

### 7.1 ACCESSING CHARACTERS IN PYTHON

In Python, the Indexing function can be used to target specific character within a String. Negative address references, such as -1 for the last character, -2 for the second last character, and so on, allow characters from the back of the String to be accessed.

An IndexError will occur if you access an index that is out of range. Only integers, floats, and other kinds that will produce a TypeError are allowed to be used as indexes.

### 7.2 OPERATIONS IN STRINGS

7  
As we all know, a string is a collection of characters. Concatenation, repetition, membership, and slicing are all possible operations on the string data type in Python. These operations are detailed in detail in the subsections below, along with relevant examples.

- I. **Concatenation:** To concatenate is to bind two things together. The concatenation operator plus, symbolised by the symbol +, in Python allows us to connect two strings.

Example: >>> str1 = 'Major' #string 1  
>>> str2 = 'Project!' #string 2  
>>> str1 + str2 #Concatenated strings  
'Major Project!'  
#str1 and str2 remain same  
>>> str1  
#after this operation.  
'Major'  
>>> str2  
'Project!'

**Repetition:** The repetition operator in Python, symbolised by the symbol \*, allows us to repeat a text.

```
Example: >>>s1= 'Project'  
>>>s1*2  
'ProjectProject'  
>>>s1*4  
'ProjectProjectProjectProject'
```

**Membership:** 'in' and 'not in' are two Python membership operators. The 'in' operator returns a pair of strings. True is returned if the 1st string is a substring of the 2nd string; otherwise, False is returned.

```
Example: >>> s1 = 'Major Project!'  
>>> 'M' in s1  
True  
>>> 'Maj' in s1  
True  
>>> 'My' in s1  
False
```

**Slicing:** We utilise a function called slicing in Python to retrieve a portion of a string or substring.<sup>7</sup> This can be accomplished by using An index range is specified by strings 179. The slice method str1[n:m] returns the portion of the string str1 starting at <sup>7</sup>dex n (inclusive) and continuing at m, given a string str1 (exclusive). In other words, str1[n:m] returns all characters starting with str1[n] <sup>7</sup>d ending with str1[m-1]. The difference of two indices m and n, i.e., the number of characters in the substring, will always be equal to the number of characters in the substring (m-n).

### 7.3 BUILT IN FUNCTION IN STRINGS

len()	Returns length of string
title()	Returns first letter of string with uppercase and rest with lowercase
lower()	Used to convert all uppercase letters to lowercase
upper()	Used to Convert all lowercase letters to uppercase
count(s,start,end)	Returns no of times substring s occurs in <sup>7</sup> at string
endswith()	Returns true if the string ends with that particular substring
startswith()	Returns true if the string starts with that particular string
isalnum()	Returns true if either the character is string or numeric

isupper()	Returns true if all characters are in uppercase
islower()	Returns true if all characters are in lowercase

**Table 7.1** Built in functions in Strings

#### 7.4 ESCAPE SEQUENCE

The escape sequence is shown by the backslash(/) symbol. A special character can be followed by a backslash, which is interpreted differently. Single quotes must be escaped within the string. We can use the same method as with double quotations.

\newline	5 nores new line
\\	backslash
'\'	Single quotes
\"	Double Quotes
\a	ASCII Bell
\b	ASCII Backspace(BS)
\f	ASCII formatted
\n	ASCII linefeed
\r	ASCII Carrige Return (CR)
\t	ASCII Horizontal Tab
\v	ASCII Vertical Tab
\ooo	Character with octal value
\xHH	Character with hex value

**Table 7.2** Escape Sequence in Strings

## CHAPTER 8:

# LIBRARIES IN PYTHON

A library is a collection of volumes or a place or location where a large number of books are stored for future use. A library is a collection of precompiled programmes that can be used in a programme for certain, well-defined actions later on. Guides, configuration data, message formats, methods, and variables, among other things, can be found in a library.

1 A Python library is a collection of related modules. It includes code packages that can be utilised in a number of different programmes. For programmers, it automates and facilitates Python programming. We won't have to write the same code for different programmes because we won't have to. Machine learning, data science, data visualisation, and other fields rely heavily on Python libraries.

### 8.1 WORKING OF PYTHON LIBRARY

1 A Python library is basically a collection of computer code or modules of codes that we can utilise in a programme for specific actions, as described previously. We use libraries to avoid having to rewrite code that is already present in our programme. However, here's how it functions. The library files have a DLL extension in Microsoft Windows (Dynamic Load Libraries). When we link a library to a programme and start it, the linker looks for that library automatically. It takes the library's functions and applies them to the programme. In our software, we use a library's methods in this manner. We'll look at how we use libraries in Python applications in more detail later.

### 8.2 PYTHON STANDARD LIBRARY

Python's exact syntax, linguistics, and symbol are all included in the Python Standard Library. It comes with built-in modules that provide you access to essential system functions like I/O and other core modules. The C programming language is used to create the majority of the Python libraries. Over 200 core modules make up Python's standard library. Python is a high-level programming language thanks to all of these factors. The Python Standard Library is really useful. Programmers won't be able to use Python's features until they have it. Apart from that, Python has a number of libraries that make a programmer's life easier. Let's look at a few of the most popular libraries:

**TensorFlow:** Google's TensorFlow library was created in partnership with the Brain Team. It's a high-level calculation library that's open-source. It's also found in deep learning and machine learning algorithms. There are a lot of tensor operations in it. This Python module is also used by researchers to solve complicated calculations in mathematics and physics.

**Matplotlib:** Matplotlib is a library that allows you to plot numerical data. It's for this reason that it's employed in data analysis. It's an open-source toolkit that creates high-resolution graphics such as pie charts, scatter plot, and graphs.

**Numpy:** Numpy is an acronym for "Numerical Python." It is the most widely utilised library. It's a very well machine-learning library that really can handle multi-dimensional data and large matrices. For rapid calculations, it features built-in mathematical functions.. Numpy is used internally by libraries like TensorFlow to conduct a variety of tensor operations. One of the most important components of this library is the Array Interface.

**SciPy:** "SciPy" is an acronym for "Scientific Python." It's a high-level scientific computation package that's open-source. This library is based on a Numpy extension. It uses Numpy to perform complex calculations. The numerical data code is stored in SciPy, whereas Numpy supports sorting and indexing of array data. It is also commonly used by engineers and application developers.

**Scrapy:** Scrapy is an accessible library that allows you to scrape data from websites. It supports both high-level screen scraping and incredibly fast web crawling. It's also useful for automated data testing and data mining.

**Scikit-learn:** It's a well-known Python framework for working with huge datasets. Scikit-learn is an open-source machine learning library. It supports a variety of supervised and unsupervised algorithms, including regression analysis, classification, clustering, and others. This package makes use of the Numpy and SciPy libraries.

**PyGame:** This package provides a simple interface to the platform-independent graphics, audio, and input libraries of the Standard Directmedia Library (SDL). It's used to make video games utilising the Python programming language and computer animation and audio packages.

**PyTorch:** It is the nation's biggest machine learning library for tensor computations. It includes a lot of APIs for doing tensor computations with GPU acceleration. It also aids in the resolution of neural network application challenges.

**PyBrain:** The name "PyBrain" stands for Python Based Reinforcement Learning, Artificial Intelligence, and Neural Networks library. It's an open-source library designed for people who are new to machine learning. It includes machine learning techniques that are quick and simple to use. It is quite adaptable and simple to comprehend, which makes it particularly useful for developers who are new to research topics.

Python has a plethora of libraries. For our objectives, we can use an appropriate library. As a result, Python libraries are extremely important and beneficial to developers.

### **8.3 USE OF LIBRARIES IN PYTHON**

We want to keep the modularity of the code while we create large-scale Python apps. For ease of maintenance, we separate the code into portions and we can use that code and when we need it. In Python, modules play this role. We define regularly used functions as modules, which we can then import into any programme, rather than repeating the same code in numerous projects, which makes the code more difficult. Although we are not required to develop that code, we may still benefit from its

advantages by importing its module. A library stores a number of interconnected modules. And we import a module from its library anytime we require it.

1

Let's have a look at the following example:

```
# Importing math library  
  
import math  
A = 36  
print(math.sqrt(A))
```

Output: 6.0

Without writing the original code to calculate the square root of an integer, we loaded the math library and utilised one of its methods, sqrt (square root). This is how a library helps programmers. We only needed the sqrt method of the math library in this case, therefore we imported the entire library. Alternatively, we can use a library module to import certain things.

#### **8.4 IMPORTING SPECIFIC ITEMS FROM A LIBRARY MODULE**

1

We imported a whole library to use one of its methods, just like in the previous code. However, we could have just used the math library's "sqrt" function. We can import certain items from a library using Python.

Consider the following code as an example:

```
# Importing specific items
```

```
from math import sqrt, sin  
A = 81  
B = 3.14  
print(sqrt(A))  
print(sin(B))
```

Output: 9.0

0.0015926529164868282

## CHAPTER 9:

### 19 NUMPY

NumPy (numerical Python) is a collection of multidimensional array objects and routines for manipulating them. NumPy is a Python library that allows you to perform mathematical and logical operations on arrays.

6  
Python scripting language NumPy. Its full name is 'Numerical Python.' It's a library that includes multidimensional array objects as well as array processing functions.

12  
Numeric, the precursor of NumPy, was designed by Jim Hugunin. There was also the creation of Numarray, a new package with some additional capabilities. Travis Oliphant designed the NumPy package in 2005 by integrating the Numarray and Numeric packages' features. A vast number of people have contributed to this open-source project.

#### 9.1 INSTALLATION

NumPy can be installed on Mac and Linux using the pip command:  
install numpy with pip

56  
There is no analogous package manager for Windows to the those found on Linux or Mac. One can manually install it on windows.

#### 12 9.2 OPERATIONS USING NUMPY

NumPy allows a developer to perform the following tasks:

- I. Both mathematical and logical array operations are available.
- II. Algorithms and Fourier transforms for sha19 change.
- III. Linear algebra operations are the same as algebraic operations.

NumPy has functions for linear algebra and generating random numbers.

NumPy is a Python-based alternative to MatLab.

NumPy is widely combined with Python library and Matplotlib to create graphs. MatLab, a prominent technical computing platform, is commonly replaced with this combination. Python, on the other hand, is now often recognised as a more modern and efficient programming language than MatLab.

NumPy has the added benefit of being open-source.

The most significant object in NumPy is the ndarray type of N-dimensional array. It refers to a group of goods that are all of the same type.

12

Every item in an ndarray is the same size as the memory block. Each entry in ndarray is a data-type object object (called dtype).

An array scalar object replicates any item extracted (by slicing) from an ndarray object. The following diagram illustrates the link between ndarray, data-type object (dtype), and array scalar type.

Different array construction routines discussed later in the course can be used to create an instance of the ndarray class. The following is how to make a basic ndarray in NumPy using an array function:

`numpy.array`

It can make an ndarray out of any object with an array interface or any method that returns an array.

`numpy.array(object, dtype = None, copy = True, order = None, subok = False, ndmin = 0)`

The ndarray object is made up of a one-dimensional continuous expanse of memory storage and an indexing method that links each item to a memory address. The elements are kept in either row-major order (C style) or column-major order in the memory block (FORTRAN or MatLab style).

Following parameters are used in the above:

Sr.No.	Parameter & Description
1	<b>object</b> Any object exposing the array interface method returns an array or any (nested) sequence.
2	<b>dtype</b> The desired data type of array, optional
3	<b>copy</b> Optional. By default (true), the object is copied
4	<b>order</b> C (row-major) or F (column-major) or A (any) (default)
5	<b>subok</b> By default, returned array forced to be a base class array. If true, sub-classes passed through
6	<b>ndmin</b> Specifies minimum dimensions of the resultant array

**Figure 9.1** Parameters used in NumPy

The ndarray object is made up of a one-dimensional continuous expanse of memory storage and an indexing method that links each item to a memory address. The elements are kept in either row-major order (C style) or column-major order in the memory block (FORTRAN or MatLab style).

### **9.3 DATA TYPES IN NUMPY**

34  
Listed below are the data types in NumPy.

bool_	Boolean (True or false) stored as byte
int_	Default integer type
int p	Used for indexing
int c	Usually int32 or int64
int 8	Byte(-128 to 127)
int 16	Integer(-32768 to 32767)
float_	Shorthand for float64
float64	double precision float
complex_	Shorthand for complex 128
complex64	Complex number, represented by two 32-bit floats
complex128	Complex number, represented by two 64-bit floats

**Table 9.1** Data Types in NumPy

### **9.4 WHY IS NUMPY FASTER IN PYTHON?**

Because of the following reasons, NumPy Arrays are faster than Python Lists:

- 28
- An array is a data structure of identical data types that are stored in contiguous memory locations. In Python, a list is a collection of different data types that are stored in non-contiguous memory areas.
  - The NumPy package divides a task into many fragments, which are subsequently processed in parallel.
  - NumPy is a Python library that integrates C, C++, and Fortran code. In comparison to Python, these programming languages have an extremely short execution time.

## CHAPTER 10

### DLIB

Dlib is a general-purpose, cross-platform software library written in the C++ computer language. Contract-based software engineering and component-based software engineering are heavily influencing its architecture. As a result, it is primarily and initially a set of self-contained software components. The Boost Software License governs the distribution of this free software.

Since its start in 2002, Dlib has grown to include a diverse set of utilities. As of 2016, it includes computer elements for networking, messaging, GUI, data formats, linear algebra, machine learning, image processing, a mining, and a range of other activities, as well as a variety of other activities. Dlib was published in the Journal of Machine Learning Research in 2009, and most of the work in recent years has gone into establishing a broad range of statistical machine learning tools. Since then, it's been used in a variety of fields.

#### 10.1 INSTALLATION

Inorder to install dlib,one must first install cmake library,

```
pip install cmake
```

After that one can install dlib library using pip install,

```
pip install dlib
```

#### 10.2 WHAT IS DLIB USED FOR?

Dlib is a pre-trained benchmark facial detector that uses 68 coordinates (x, y) to estimate the positioning of face points on a person's face, as shown in the image below. The iBUG300-W dataset and a pre-trained algorithm were used to find these sites.

Dlib is a modern C++ toolkit with machine learning algorithms and tools for creating complex C++ software that addresses real-world problems. In both industry and academia, it is used in a variety of domains, including robotics, microcontrollers, cell phones, and enormous high-performance computing systems.

CNN and HoG are the two types of algorithm that can be analyzed using dlib.

### 10.3 HoG FACE DETECTOR IN DLIB

13

HOG, or Histogram of Oriented Gradients, is a feature extraction technique for extracting features from image data. It is commonly utilised in object detection tasks in computer vision.

Let's take a look at some of the key features of HOG that set it apart from other feature descriptors:

61

- The HOG descriptor is concerned with an object's structure or form. How is this different from the edge characteristics we extract for photos, you might wonder? We just decide whether or not a pixel is an edge in case of edge features. HOG can also be used to provide the edge direction. This is done by extracting the gradient and orientation of the edges (or magnitude and direction).
- In 'localised' parts, these orientations are likewise defined. This implies the entire image is separated into several pieces, each with its own gradients and orientation. In the following sections, we'll go through this in further depth.
- Finally, in each of these sections, the HOG would build a separate Histogram. The statistics are made up of the gradient and orientation of the pixel values, hence the term 'Histogram of Oriented Gradients.'

67

#### Pros:

- On the CPU, the fastest technique is
- For frontal and somewhat non-frontal faces, this is a great option.
- In comparison to the other three, this model is the lightest.
- When there is a modest amount of occlusion, it works.

#### Cons:

- The main disadvantage is that it cannot detect little faces because it is trained to detect faces with a minimum face size of 8080 pixels. As a result, you must ensure that the face size in your application is greater than that. However, you can train your own face detector to recognise tiny faces.
- The bounding box frequently excludes a portion of the forehead and, on rare occasions, a portion of the chin.
- When there is a lot of occlusion, it doesn't operate very well.
- Side faces and extreme non-frontal faces, such as looking down or up, do not work.

11

## **10.4 CNN FACE DETECTOR IN DLIB**

8

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that can take an input image and assign importance (trainable weights and biases) to various parts of it, as well as distinguish between them. The amount of pre-processing required by a ConvNet is much less than that required by other classification methods. ConvNets can learn these filtration with adequate training, whereas simple techniques require hand-engineering of filters.

55

A ConvNet's architecture is inspired by the Visual Cortex's organization and is similar to the related to this particular of Nerve cells in the Human Brain. Individual neurons can only respond to stimuli in the Receptive Field, a tiny portion of the visual field. To span a distance, a number of comparable fields can be piled on top of one another.

**Pros:**

- It's suitable for a variety of face shapes.
- Occlusion-resistant
- On the GPU, it runs really quickly.
- The training method is really simple.

**Cons:**

- CPU performance is really slow.
- Because it is trained for a minimum face size of 80 x 80, it does not recognise little faces. As a result, you must ensure that the face size in your application is greater than that. However, you can train your own face detector to recognise tiny faces.
- The bounding box is a fraction of the size of the HoG detector.

## CHAPTER 11

### OPEN CV

26

The OpenCV-Python library is a collection of Python bindings for computer vision. Guido van Rossum designed Python, a general purpose programming language that rapidly gained prominence due to its ease of use and readability of code.

#### **11.1 INSTALLATION**

27

- Download and install Visual Studio and CMake.
- Download and install necessary Python packages to their default locations  
Python & Numpy

35

Check that Python and Numpy are working properly.

Get the OpenCV source code. It can come from Sourceforge (for the official release version) or Github (for the development version) (for latest source).

Opencv and make a new folder in it after extracting it to a folder.

27

CMake-gui (Start > All Programs > CMake-gui) is now open.

Fill in the following fields (as shown in the image below):

Locate the opencv folder by selecting Browse Source... from the File menu.

35

Select Browse Build... and navigate to the build folder we established before.

Select Configure.

#### **11.2 WHAT IS OPENCV USED FOR?**

OpenCV is a fantastic image analysis and computer vision application. It's an open-source library that may be used for tasks like feature selection, object tracking, and landmark detection, among others. Python, Java, and C++ are just a few of the languages that can be used.

Some of the applications of OpenCV are as follows:

- Image stitching from a street view
- Inspection and surveillance by robots
- Car navigation and control by robots and self-driving cars
- Image analysis in medicine
- Search and retrieval of video/images
- Motion pictures - 3D structure
- Installations of interactive art

### **11.3 OPENCV FUNCTIONALITY**

- Image/video I/O, processing, and display are all things that need to be taken into consideration
- Detection of objects or features
- Computer vision based on geometry, monocular or stereo
- Photographic computation
- Clustering and machine learning are two terms that come to mind while thinking about machine learning (ml, flann)
- Acceleration using the CUDA architecture (gpu)

### **11.4 ADVANTAGES**

- OpenCV is free.
- OpenCV is a fast library because it is written in C/C++.
- RAM utilisation is low (about 60–70 megabytes).
- Because OpenCV can run on any device that can run C, it is extremely portable.

### **11.5 DISADVANTAGES**

- When compared to MATLAB, OpenCV is less user-friendly.
- A flann library is included with OpenCV. When you try to utilise the OpenCV library in conjunction with the PCL library, you'll run into complications.

## CHAPTER 12:

### FACE RECOGNITION ATTENDANCE SYSTEM

This chapter mainly focuses on the theoretical part related to attendance using face-recognition, i.e., the different types of attendance systems and face recognition techniques.

To ensure that there is discipline among students and to ensure that students get utmost knowledge, the attendance system was introduced in school, colleges and universities. There are mainly two methods to mark the attendance of a student. The first is where the person-in-charge calls out the roll number of the student and the second method is to take the student signature on a paper against his/her roll number. A large number of applications can be made around this idea, which will not only help save time, but also eliminate the chance of proxy attendance due to its effective identification. The main objective behind this project is to build an Automated Attendance System using Python Language and with the help of OpenCV/Python libraries and recognizer algorithm. The proposed application can be used in any institution or field, where attendance system is present and plays a vital role in the working of that institution. It can be said that this project is an effective solution for all universities and colleges to track and manage the regular attendance of its students.

#### 12.1 INTRODUCTION

The fast-growing technology is imparting<sup>10</sup> a magnificent amount of knowledge-oriented technical ideas and innovations. Deep Learning is one of the most interesting technology in which the machine trains itself after we provide it with some datasets as an input and then, the machine applies different learning algorithms and then provides the corresponding output.<sup>41</sup> In today's world, attendance of a student is considered to be very important for the student as well as the management faculty of an institution. With the advancement in the field of deep learning technology, the machine can now detect the attendance of the student by just his/her face and also maintains a record of the data collected by it.

#### 12.2 DIFFERENT TYPES OF ATTENDANCE SYSTEMS

<sup>49</sup>

Broadly, the attendance of a student can be recorded and maintained in 2 ways, which are:

- Manual Attendance System (MAS) □
- Automated Attendance System (AAS).

### **12.3 MANUAL ATTENDANCE MANAGEMENT SYSTEM**

In a manual attendance management system, a faculty member who is responsible for a specific subject calls out the student's name/roll number and manually stamps his or her attendance if he or she is present. Manual attendance can be a time-consuming or stressful process, and in some situations, the teacher may misspell or skip someone's name, or a student may answer on place of his absent classmates many times, which is referred to as proxy attendance. As a result, the traditional method of taking physical attendance in the classroom has a number of drawbacks. We choose Automatic Attendance System (AAS) to help us overcome these drawbacks.

### **12.4 AUTOMATED ATTENDANCE MANAGEMENT SYSTEM**

10

Automated Attendance System (AAS) is a process in which the presence or the absence of a particular student in the classroom is automatically detected by using face recognition technology and Deep Learning. In addition to this, it is also possible to detect whether the student has 10 slept during the lecture or whether he/she is awake during the lectures. This system can also be implemented during an exam to ensure that the student is present. The presence of a student in a class can be determined by capturing their face on a high-definition camera or video device, so that it is very easy for the machine to detect the presence of all the students in the classroom.

### **12.5 DIFFERENT FACE RECOGNITION TECHNIQUES**

37

There are mainly 2 types of Face-Recognition techniques, which are-

- Feature-based approach
- Brightness-based approach

The feature-based technique (also known as a local face recognition system) is used to point 31 essential characteristics of the face such as the ears, eyes, mouth, and nose, while the brightness-based approach (also known as a global face recognition system) is used to recognise all of the image's parts.



Example of Feature-based approach

### **12.6 ADVANTAGES :**

1. Automated time tracking system
2. Cost effective
3. Increased security
4. Time saving
5. Easy to manage
6. Reduces the chances for proxy attendance
7. Secure
8. Boon to the current pandemic situation
9. More accurate as compared to the manual attendance or biometric system
10. Smart integration

### **12.7 DISADVANTAGES:**

1. Data privacy breach
2. If in wrong hands,it can cause disaster
3. Low reliability
4. Lack of regulations in the AI in Face recognition systems
5. <sup>47</sup> Ita Privacy Concern.
6. Poor Image Quality Limits Facial Recognition's Effectiveness.
7. Small Image Sizes Make Facial Recognition More Difficult.
8. Massive data storage

### **12.8 HOW FACIAL RECOGNITION IS USED**

The technology is used for a wide range of applications. These are some examples:

- Phone unlocking
- Airports and border crossings
- Enforcement of the law
- Locating missing people
- Lowering retail crime
- Enhancing retail experiences
- Banking
- Advertising and marketing
- Healthcare
- Keeping track of student or employee attendance

## **12.9 LIMITATIONS AND CHALLENGES**

Limitations: There are capacity drawbacks to the use of facial reputation, which include threats to privateness, violations of rights and private freedoms, ability records robbery and other crimes. There's also the hazard of errors because of flaws in the technology.

Challenges: Pose changes are quite sensitive to facial recognition systems. When the character's top movement and viewing attitude change, the character's facial position changes. As the rotating viewpoint increases, it becomes a task to perceive the genuine face.

## **12.10 FUTURE OF FACE RECOGNITION ATTENDANCE SYSTEM**

The outcome of the phase of facial predominance is remarkable. Forecasters anticipate that this time will progress at a breakneck pace, resulting in major agreements in the not-too-distant future. The major sections that have a purpose to be exceedingly advanced are security and reconnaissance. Private enterprises, public structures, & resource are just a several of the domains that are currently enthusiastically adopting it. It is believed that it will be quickly followed by the use of retailers and banking frameworks to keep up with deception in charge/FICO assessment card purchases and expenses, particularly those made online. The breakout clauses of a severely deficient secret phrase device would be filled by this innovation. Robots that employ facial recognition technology may be subject to regulation in the future.

## CHAPTER 13

### IMPLEMENTATION OF THE CODE

This chapter focuses on implementing the code on how the project is built. For a better understanding of the user, we have attached the snapshots of the codes and the outputs are also attached for reference.

#### **13.1 LANGUAGES USED**

PYTHON

#### **13.2 SOFTWARES USED**

- **Visual Studio Code(VS Code)**-Code editor used for editing
- **Tkinter**- For whole GUI
- **OpenCV**-For taking images and face recognition  
(cv2.face.LBPHFaceRecognizer\_create())
- **Others**- CSV,Numpy,Pandas,datetime etc.

#### **13.3 STEPS TO BE FOLLOWED**

The following packages are needed for this project in Python. You can install them by running the following command:-

1. <sup>42</sup> pip install tk-tools
2. pip install opencv-contrib-python
3. pip install datetime
4. pip install pytest-shutil
5. pip install python-csv
6. pip install numpy
7. pip install pillow
8. pip install pandas
9. pip install times
10. pip install face\_recognition
11. pip install dlib

### 13.4 IMPORT LIBRARIES

```
import cv2
import numpy as np
import face_recognition
```

### 13.5 EXECUTION OF THE CODES

Attached below are the snippets of the code, which were executed in the system. We have created a folder named main.py - this folder was used to write the codes. The reader can enter any name for the folder as per their choice and convenience.

Import all necessary libraries like that,

The screenshot shows a Windows desktop environment. In the center is a code editor window titled "main.py". The code itself is a Python script for a Face Recognition Based Attendance Monitoring System. It imports various libraries including tkinter, cv2, os, numpy, PIL, pandas, datetime, and time. It defines several functions: "assure\_path\_exists" to create directory paths, "tick" to update a digital clock, "contact" to show a message box with contact information, and "check\_haarcascadefile" to verify the presence of a Haar cascade XML file. The code ends with a call to "window.destroy()". At the bottom of the screen, the taskbar is visible with icons for File Explorer, Edge browser, Mail, and others. A search bar at the bottom left contains the placeholder "Type here to search". The system tray shows battery level at 66%, a 31°C temperature, and other standard icons.

```
main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help
#####
# IMPORTING #####
#####
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox as mess
import tkinter.simpledialog as tsd
import cv2,os
import csv
import numpy as np
from PIL import Image
import pandas as pd
import datetime
import time

#####
# FUNCTIONS #####
#####

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

#####

def tick():
    time_string = time.strftime('%H:%M:%S')
    clock.config(text=time_string)
    clock.after(200,tick)

#####

def contact():
    mess._show(title='Contact us', message="Please contact us on : shubhamkumar8180323@gmail.com")

#####

def check_haarcascadefile():
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
    if exists:
        pass
    else:
        mess._show(title='Some file missing', message='Please contact us for help')
        window.destroy()

#####
# Main Program #####
#####

window = tk.Tk()
clock = tk.Label(window, font=(('times new roman'), 18), bg='white')
clock.pack(fill=tk.X, expand=1)
assure_path_exists("Attendance")
attendance = pd.read_csv("Attendance.csv")
mess._show(title='Attendance System', message='Attendance System')
window.mainloop()
```

Figure 13.1- Snapshot of the code(1)

Fig 13.1 The first step is creating a face cascade, which stores the data it collects to detect faces. After that, it loads the data into memory and ready to use.

Face Recognition Algorithms –

- Haar Cascade
- LBPH (Local Binary Pattern Histogram)

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help

def save_pass():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        master.destroy()
        new_pas = tsd.askstring("Old Password not found", 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered successfully!!!')
            return
    op = (old.get())
    newp = (new.get())
    nnewp = (nnew.get())
    if (op == key):
        if (newp == nnewp):
            txf = open("TrainingImageLabel\psd.txt", "w")
            txf.write(newp)
        else:
            mess._show(title='Error', message='Confirm new password again!!!')
        return
    else:
        mess._show(title='Wrong Password', message='Please enter correct old password.')
    return
mess._show(title='Password Changed', message='Password changed successfully!!!')
master.destroy()

#####
def change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")

    lb14 = tk.Label(master,text=' Enter Old Password',bg='white',font=('comic', 12, ' bold '))
    lb14.place(x=10, y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('comic', 12, ' bold '),show='*')
    old.place(x=180, y=10)
    lb15 = tk.Label(master, text=' Enter New Password', bg='white', font=('comic', 12, ' bold '))
    lb15.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('comic', 12, ' bold '),show='*')
    new.place(x=180, y=45)
    lb16 = tk.Label(master, text='Confirm New Password', bg='white', font=('comic', 12, ' bold '))
    lb16.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('comic', 12, ' bold '),show='*')
    nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel", command=master.destroy,fg="black" ,bg="red" ,height=1,width=25 , activebackground = "white",font=('comic', 10, ' bold '))
    cancel.place(x=10, y=120)
    save=tk.Button(master, text="Save", command=save_pass, fg="black", bg="#00fcca", height = 1,width=25, activebackground="white", font=('comic', 10, ' bold '))
    save.place(x=10, y=120)
    master.mainloop()
#####

def pw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring("Old Password not found", 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")

```

Figure 13.2- Snapshot of the code(2)

Fig 13.2 & 13.3 we create a password for security purpose to safe a data and also you change your password as per your convenience.

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help

def change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")

    lb14 = tk.Label(master,text=' Enter Old Password',bg='white',font=('comic', 12, ' bold '))
    lb14.place(x=10, y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('comic', 12, ' bold '),show='*')
    old.place(x=180, y=10)
    lb15 = tk.Label(master, text=' Enter New Password', bg='white', font=('comic', 12, ' bold '))
    lb15.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('comic', 12, ' bold '),show='*')
    new.place(x=180, y=45)
    lb16 = tk.Label(master, text='Confirm New Password', bg='white', font=('comic', 12, ' bold '))
    lb16.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('comic', 12, ' bold '),show='*')
    nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel", command=master.destroy,fg="black" ,bg="red" ,height=1,width=25 , activebackground = "white",font=('comic', 10, ' bold '))
    cancel.place(x=10, y=120)
    save=tk.Button(master, text="Save", command=save_pass, fg="black", bg="#00fcca", height = 1,width=25, activebackground="white", font=('comic', 10, ' bold '))
    save.place(x=10, y=120)
    master.mainloop()
#####

def pw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring("Old Password not found", 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")

```

Figure 13.3- Snapshot of the code(3)

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help
def pw():
    assure_path_exists("TrainingImageLabel")
    extst1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was registered successfully!!')
            return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
        TrainImages()
    elif (password == None):
        pass
    else:
        mess._show(title='Wrong Password', message='You have entered wrong password')
    #####
def clear():
    txt.delete(0, 'end')
    res = "1)Take Images >> 2)Save Profile"
    message1.configure(text=res)

def clear2():
    txt2.delete(0, 'end')
    res = "1)Take Images >> 2)Save Profile"
    message1.configure(text=res)

#####
def TakeImages():
    check_haarcascadefile()
    columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
    assure_path_exists("StudentDetails/")
    #####
    def clear():
        txt.delete(0, 'end')
        res = "1)Take Images >> 2)Save Profile"
        message1.configure(text=res)

    def clear2():
        txt2.delete(0, 'end')
        res = "1)Take Images >> 2)Save Profile"
        message1.configure(text=res)

    #####
    def TakeImages():
        check_haarcascadefile()
        columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
        assure_path_exists("StudentDetails/")
        serial = 0
        exists = os.path.isfile("StudentDetails\StudentDetails.csv")
        if exists:
            with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
                reader1 = csv.reader(csvFile1)
                for l in reader1:
                    serial = serial + 1
                serial = (serial // 2)
            csvFile1.close()
        else:
            with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
                writer = csv.writer(csvFile1)
                writer.writerow(columns)
                serial = 1
            csvFile1.close()
        Id = (txt.get())
        name = (txt2.get())
        if ((name.isalpha()) or (' ' in name)):
            cam = cv2.VideoCapture(0)
            harcascadePath = "haarcascade_frontalface_default.xml"
            detector = cv2.CascadeClassifier(harcascadePath)
            sampleNum = 0
            while (True):
                ret, img = cam.read()
                gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                faces = detector.detectMultiScale(gray, 1.3, 5)
                for (x, y, w, h) in faces:
                    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                    # incrementing sample number
                    sampleNum = sampleNum + 1
                    # saving the captured face in the dataset folder TrainingImage
                    cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' + str(sampleNum) + ".jpg",
                               gray[y:y + h, x:x + w])
                    # display the frame
                    cv2.imshow('Taking Images', img)
                # wait for 100 milliseconds
                if cv2.waitKey(100) & 0xFF == ord('q'):
                    break
            # break if the sample number is morethan 100

```

Figure 13.4- Snapshot of the code(4)

face\_recognition library uploads images in BGR format, to print an image you must convert it to RGB using OpenCV.

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help
columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
assure_path_exists("StudentDetails/")
assure_path_exists("TrainingImage/")
serial = 0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for l in reader1:
            serial = serial + 1
        serial = (serial // 2)
    csvFile1.close()
else:
    with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
        writer = csv.writer(csvFile1)
        writer.writerow(columns)
        serial = 1
    csvFile1.close()
Id = (txt.get())
name = (txt2.get())
if ((name.isalpha()) or (' ' in name)):
    cam = cv2.VideoCapture(0)
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    sampleNum = 0
    while (True):
        ret, img = cam.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = detector.detectMultiScale(gray, 1.3, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
            # incrementing sample number
            sampleNum = sampleNum + 1
            # saving the captured face in the dataset folder TrainingImage
            cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' + str(sampleNum) + ".jpg",
                       gray[y:y + h, x:x + w])
            # display the frame
            cv2.imshow('Taking Images', img)
        # wait for 100 milliseconds
        if cv2.waitKey(100) & 0xFF == ord('q'):
            break
    # break if the sample number is morethan 100

```

Figure 13.5- Snapshot of the code(5)

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help
    break
    # break if the sample number is morethan 100
    elif sampleNum > 100:
        break
    cam.release()
    cv2.destroyAllWindows()
    res = "Images Taken for ID : " + Id
    row = [serial, '', Id, '', name]
    with open('StudentDetails\StudentDetails.csv', 'a') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerow(row)
    csvFile.close()
    message1.configure(text=res)
else:
    if (name.isalpha() == False):
        res = "Enter Correct name"
        message.configure(text=res)

#####
def TrainImage():
    check_haarcascadefile()
    assure_path_exists("TrainingImageLabel/")
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    haarcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(haarcascadePath)
    faces, ID = getImagesAndLabels("TrainingImage")
    try:
        recognizer.train(faces, np.array(ID))
    except:
        mess._show(title='No Registrations', message='Please Register someone first!!!')
        return
    recognizer.save("TrainingImageLabel\Trainer.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
    message.configure(text="Total Registrations till now : " + str(ID[0]))
#####

def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]

```

Figure 13.6- Snapshot of the code(6)

In this section, we convert the train image into specific codes and save the codes with the given name of the person for that image.

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help
# get the path of all the files in the folder
imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
# create empty face list
faces = []
# create empty ID list
Ids = []
# now looping through all the image paths and loading the Ids and the images
for imagePath in imagePaths:
    # loading the image and converting it to gray scale
    pilImage = Image.open(imagePath).convert('L')
    # Now we are converting the PIL Image into numpy array
    imageNp = np.array(pilImage, 'uint8')
    # getting the Id from the image
    ID = int(os.path.split(imagePath)[-1].split('.')[1])
    # extract the face from the training image sample
    faces.append(imageNp)
    Ids.append(ID)
Ids.append(ID)
return faces, Ids

#####

def TrackImages():
    check_haarcascadefile()
    assure_path_exists("Attendance/")
    assure_path_exists("StudentDetails/")
    for k in tv.get_children():
        tv.delete(k)
    msg = ''
    i = 0
    j = 0
    recognizer = cv2.face.LBPHFaceRecognizer_create() # cv2.createLBPHFaceRecognizer()
    exists3 = os.path.isfile("TrainingImageLabel\Trainer.yml")
    if exists3:
        recognizer.read("TrainingImageLabel\Trainer.yml")
    else:
        mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!!')
        return
    haarcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(haarcascadePath);

    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX

```

Figure 13.7- Snapshot of the code(7)

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py [3.10.0]
File Edit Format Run Options Window Help
can = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists1:
    df = pd.read_csv("StudentDetails\StudentDetails.csv")
else:
    mess._show(title='Details Missing', message='Students details are missing, please check!')
    can.release()
    cv2.destroyAllWindows()
    window.destroy()
while True:
    ret, im = cam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
        serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
        if (conf < 50):
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
            timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
            ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
            ID = str(ID)
            ID = ID[1:-1]
            bb = str(aa)
            bb = bb[2:-2]
            attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]
        else:
            Id = 'Unknown'
            bb = str(Id)
        cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
    cv2.imshow('Taking Attendance', im)
    if (cv2.waitKey(1) == ord('q')):
        break
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
if exists:
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile:

```

Figure 13.8- Snapshot of the code(8)

Creating a function that will create an Attendance.csv file to keep attendees on time. Note: here you need to create an Attendance.csv file manually and provide a path to the task. With open (“filename.csv”, ‘r +’) creates a file and mode ‘r +’ is used to open the read and write file.

First we check that the name of the person attending the event already exists.csv we will not record attendance again.

If the name of the attendee is not available at attendance.csv we will record the name of the attendee during the work call.

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help
if exists:
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerow(attendance)
        csvFile.close()
else:
    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerow(col_names)
        writer.writerow(attendance)
    csvFile.close()
with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile:
    reader1 = csv.reader(csvFile)
    for lines in reader1:
        i = i + 1
        if (i > 1):
            if (i % 2 != 0):
                iidd = str(lines[0]) + ' '
                tv.insert('', 0, text=iidd, values=[str(lines[2]), str(lines[4]), str(lines[6])])
csvFile.close()
cam.release()
cv2.destroyAllWindows()

#####
# USED STUFFS #####
global key
key = ''
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
day,month,year=date.split("-")
mont={'01':'January',
      '02':'February',
      '03':'March',
      '04':'April',
      '05':'May',
      '06':'June',
      '07':'July',
      '08':'August',
      '09':'September',
      '10':'October',
      '11':'November',
      '12':'December'
      }

#####
# GUI FRONT-END #####
window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
window.title("Attendance System")
window.configure(background="#2d420a")

frame1 = tk.Frame(window, bg="#c79cff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#c79cff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance Monitoring System", fg="white", bg="#2d420a", width=55, height=1, font=('comic', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+" | ", fg="#ff6e5", bg="#2d420a", width=55, height=1, font=('comic', 22, ' bold '))
datef.pack(fill="both", expand=1)

clock = tk.Label(frame3, fg="#ff6e5", bg="#2d420a", width=55, height=1, font=('comic', 22, ' bold '))
clock.pack(fill="both", expand=1)

head2 = tk.Label(frame2, text="For New Registrations", fg="black", bg="#00fcca", font=('comic', 17, ' bold '))
head2.grid(row=0, column=0)

head1 = tk.Label(frame1, text="For Already Registered", fg="black", bg="#00fcca", font=('comic', 17, ' bold '))
head1.place(x=0, y=0)

```

Figure 13.9- Snapshot of the code(9)

```

#####
# GUI FRONT-END #####
window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
window.title("Attendance System")
window.configure(background="#2d420a")

frame1 = tk.Frame(window, bg="#c79cff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#c79cff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance Monitoring System", fg="white", bg="#2d420a", width=55, height=1, font=('comic', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+" | ", fg="#ff6e5", bg="#2d420a", width=55, height=1, font=('comic', 22, ' bold '))
datef.pack(fill="both", expand=1)

clock = tk.Label(frame3, fg="#ff6e5", bg="#2d420a", width=55, height=1, font=('comic', 22, ' bold '))
clock.pack(fill="both", expand=1)

head2 = tk.Label(frame2, text="For New Registrations", fg="black", bg="#00fcca", font=('comic', 17, ' bold '))
head2.grid(row=0, column=0)

head1 = tk.Label(frame1, text="For Already Registered", fg="black", bg="#00fcca", font=('comic', 17, ' bold '))
head1.place(x=0, y=0)

```

Figure 13.10- Snapshot of the code(10)

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help

head2 = tk.Label(frame2, text="For New Registrations", fg="black", bg="#00fcfa", font=("comic", 17, ' bold '))
head2.grid(row=0, column=0)

head1 = tk.Label(frame1, text="For Already Registered", fg="black", bg="#00fcfa", font=("comic", 17, ' bold '))
head1.place(x=0, y=0)

lbl = tk.Label(frame2, text="Enter ID", width=20, height=1, fg="black", bg="#c79cff", font=("comic", 17, ' bold '))
lbl.place(x=80, y=55)

txt = tk.Entry(frame2, width=32, fg="black", font=("comic", 15, ' bold '))
txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name", width=20, fg="black", bg="#c79cff", font=("comic", 17, ' bold '))
lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2, width=32, fg="black", font=("comic", 15, ' bold '))
txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images >> 2)Save Profile", fg="black", bg="#c79cff", width=39, height=1, activebackground ="#3ffc00", font=("comic", 17, ' bold '))
message1.place(x=7, y=230)

message2 = tk.Label(frame2, text="", fg="black", bg="#c79cff", width=39, height=1, activebackground ="#3ffc00", font=("comic", 16, ' bold '))
message2.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance", width=20, fg="black", bg="#c79cff", height=1, font=("comic", 17, ' bold '))
lbl3.place(x=100, y=115)

res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile:
        readerl = csv.reader(csvFile)
        for l in readerl:
            res = res + 1
    res = (res // 2) - 1
    csvFile.close()
else:
    res = 0
message.configure(text='Total Registrations till now : '+str(res))

#####
##### MENUBAR #####
#####


```

Figure 13.11- Snapshot of the code(11)

Resize image only 1/4 of the focus area. The output frame will be of the correct size. Changing the size improves the Frame per second.

`face_recognition.face_locations()` is referred to as a resized image (`imgS`).  
`face_recognition.distance()` returns the range of the image range and all images in our train guide.  
A small face distance indicator will be the same face.  
After finding the same name we call the mark Attendance function.  
Draw the connecting box using the `cv2.rectangle()`.  
We put the same name in the output frame using `cv2.putText()`.

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help

#####
##### MENUBAR #####
menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('comic', 29, ' bold'),menu=filemenu)

#####
##### TREEVIEW ATTENDANCE TABLE #####
tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=52)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ="ID")
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')

#####
##### SCROLLBAR #####
scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)

#####
##### BUTTONS #####
clearButton = tk.Button(frame2, text="Clear", command=clear ,fg="black" ,bg="#ff7221" ,width=11 ,activebackground = "white" ,font=('comic', 11, ' bold '))
clearButton.place(x=335 ,y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2 ,fg="black" ,bg="#ff7221" ,width=11 ,activebackground = "white" ,font=('comic', 11, ' bold '))
clearButton2.place(x=335 ,y=172)
takeImg = tk.Button(frame2, text="Take Images", command=TakeImages ,fg="white" ,bg="#6d00fc" ,width=34 ,height=1 ,activebackground = "white" ,font=('comic', 15 ))
takeImg.place(x=30 ,y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white" ,bg="#6d00fc" ,width=34 ,height=1 ,activebackground = "white" ,font=('comic', 15 ))
trainImg.place(x=30 ,y=380)
trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages ,fg="black" ,bg="#3ffc00" ,width=35 ,height=1 ,activebackground = "white" ,font=trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy ,fg="black" ,bg="#eb4600" ,width=35 ,height=1 ,activebackground = "white" ,font='comic '
quitWindow.place(x=30 ,y=450)

ln:60 Col:0

```

Figure 13.12- Snapshot of the code(12)

```

main.py - C:\Users\Lenovo\Dropbox\PC\Downloads\FACE RECOGNITION BASED ATTENDANCE MONITORING SYSTEM\main.py (3.10.0)
File Edit Format Run Options Window Help

#####
##### TREEVIEW ATTENDANCE TABLE #####
tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=52)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ="ID")
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')

#####
##### SCROLLBAR #####
scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)

#####
##### BUTTONS #####
clearButton = tk.Button(frame2, text="Clear", command=clear ,fg="black" ,bg="#ff7221" ,width=11 ,activebackground = "white" ,font=('comic', 11, ' bold '))
clearButton.place(x=335 ,y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2 ,fg="black" ,bg="#ff7221" ,width=11 ,activebackground = "white" ,font=('comic', 11, ' bold '))
clearButton2.place(x=335 ,y=172)
takeImg = tk.Button(frame2, text="Take Images", command=TakeImages ,fg="white" ,bg="#6d00fc" ,width=34 ,height=1 ,activebackground = "white" ,font=('comic', 15 ))
takeImg.place(x=30 ,y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white" ,bg="#6d00fc" ,width=34 ,height=1 ,activebackground = "white" ,font=('comic', 15 ))
trainImg.place(x=30 ,y=380)
trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages ,fg="black" ,bg="#3ffc00" ,width=35 ,height=1 ,activebackground = "white" ,font=trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy ,fg="black" ,bg="#eb4600" ,width=35 ,height=1 ,activebackground = "white" ,font='comic '
quitWindow.place(x=30 ,y=450)

#####
##### END #####
window.configure(menu=menubar)
window.mainloop()

ln:60 Col:0

```

Figure 13.13- Snapshot of the code(13)

### **13.6 OUTPUT**

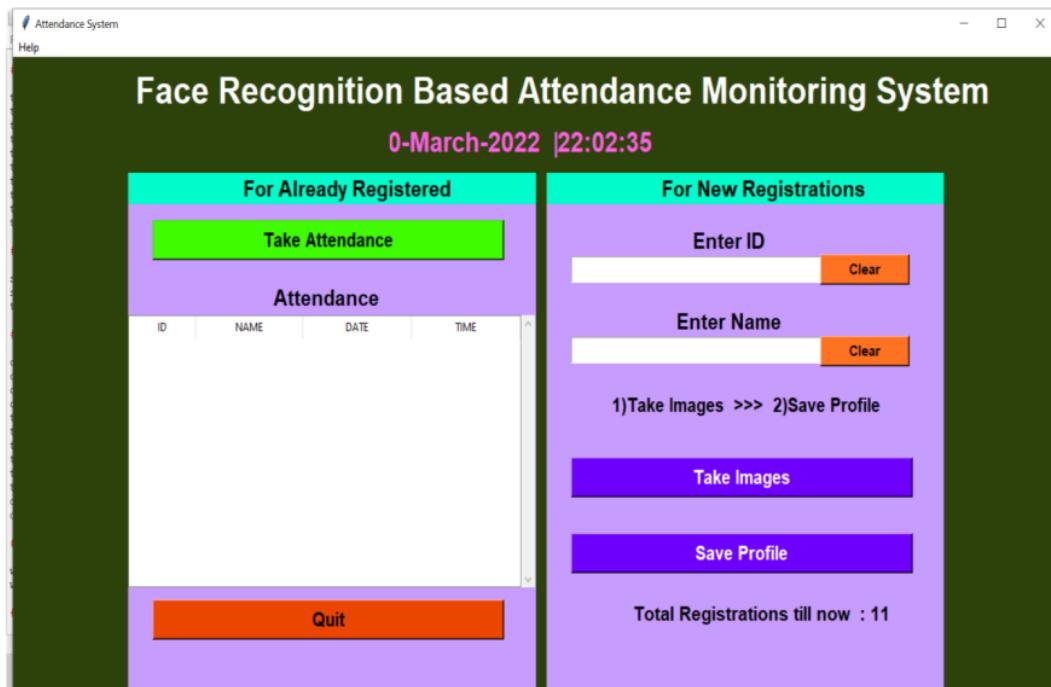


Figure 13.14- Output shown as soon as the above code is executed

Following the above-mentioned output, take the following steps:

1. Run the code through the command, which will immediately open a new window.
2. Separate the two portions The first is for new registrations, and the second is for existing registrations.
3. Next, we typed in Enter ID and Enter Name.
4. Next, click Take Images, which launches your web camera (by default permission allowed)
5. When users register for new pupils, 100 photos of that student will be taken and saved to a single folder.
6. Then press the Save Profile button. When photos are taken for any student, the user must click the save profile option, which will prompt them to provide their admin password.

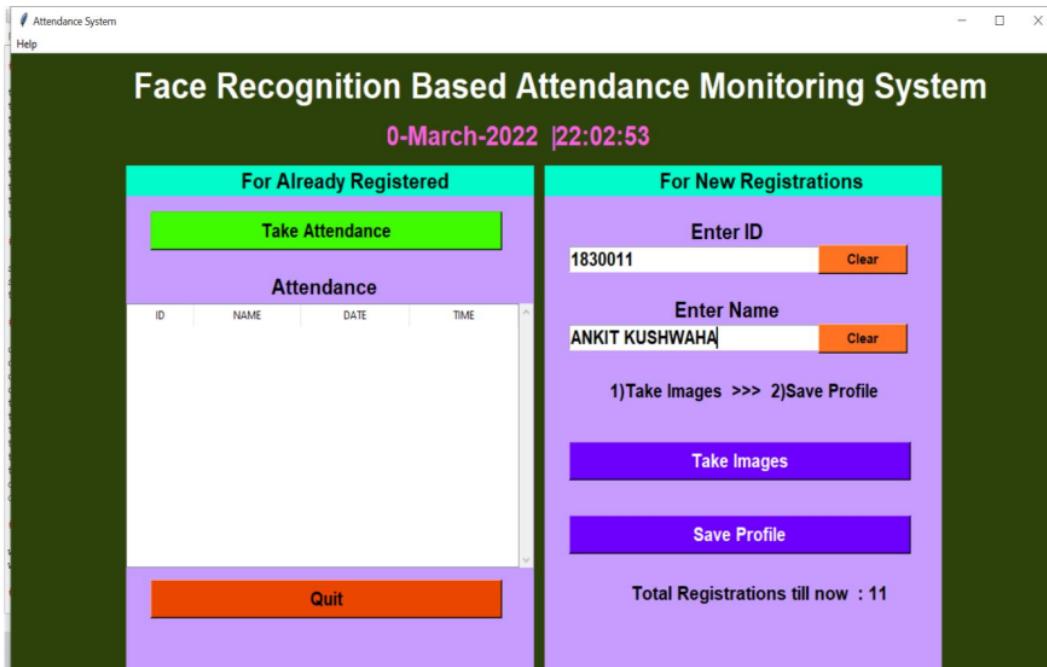


Figure 13.15- User enters the required field

Fig 13.15 User Enter ID and User Name then click the button take Images.

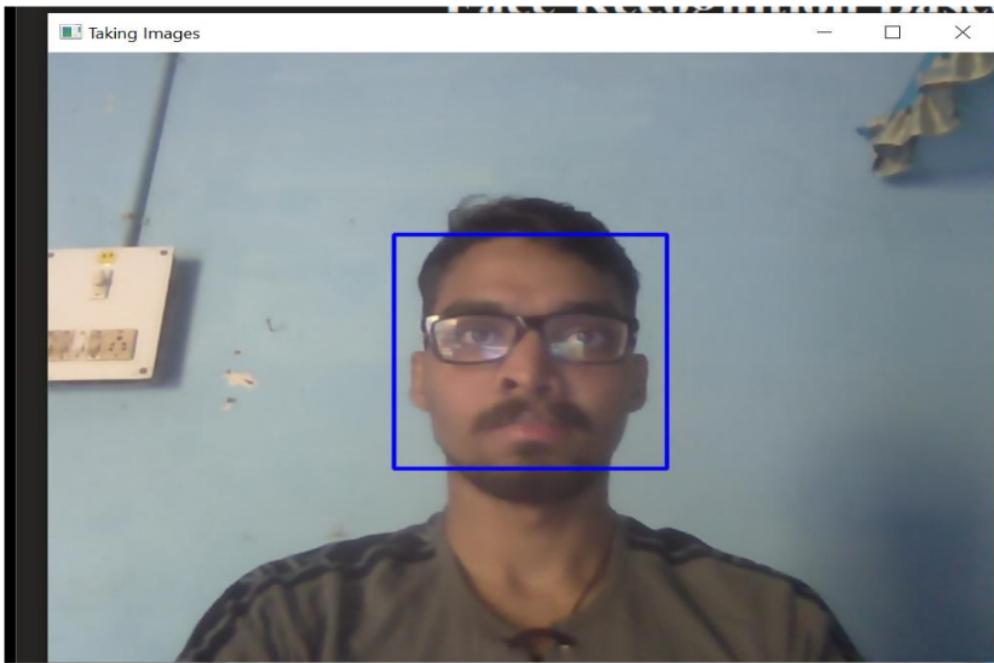


Figure 13.16. Taking Images

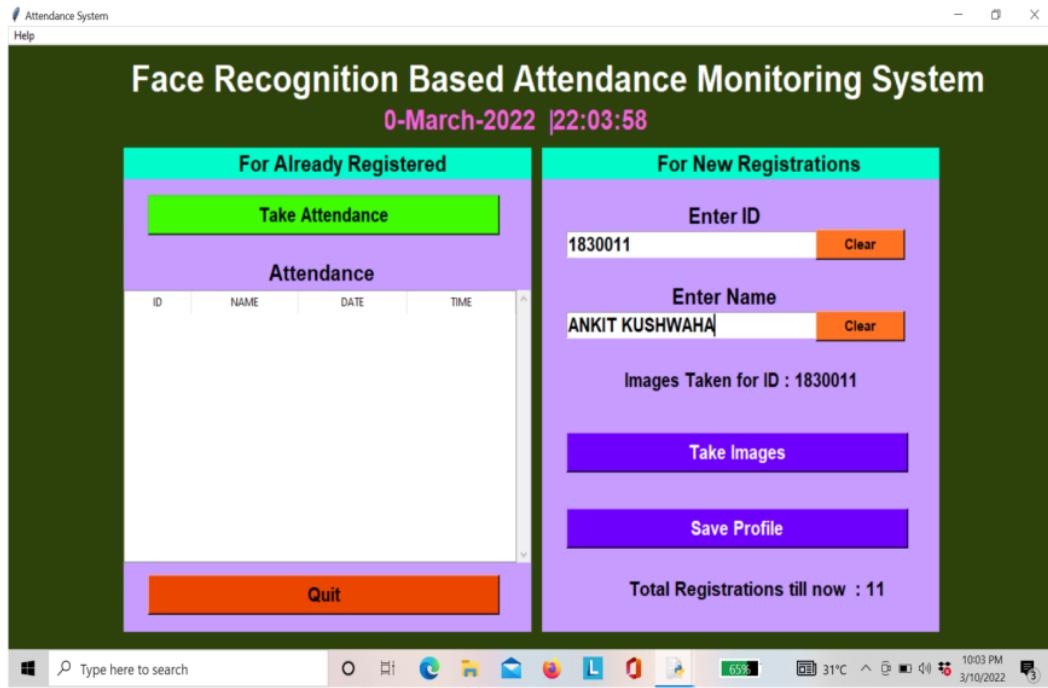


Figure 13.17- Images Taken successful

Fig 13.18. show your images taken by webcam are successful and save for your ID and its generate 100 images to store database.

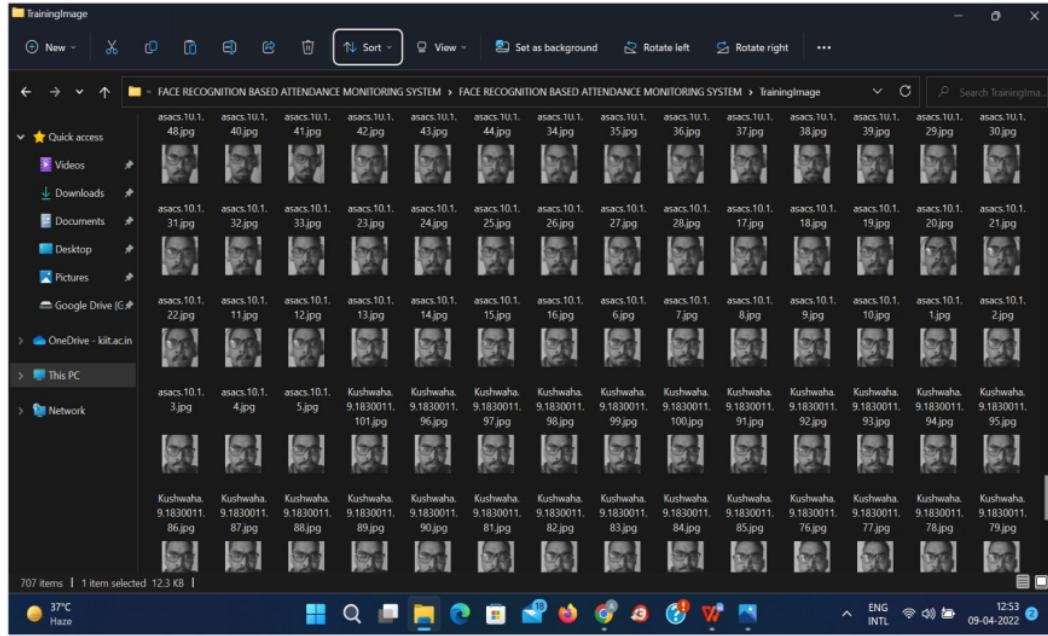


Figure 13.18- 100 Images

Fig.13.19 shows that after takes images then go to click save profile then its ask your password to save your profile successfully after entering right password.

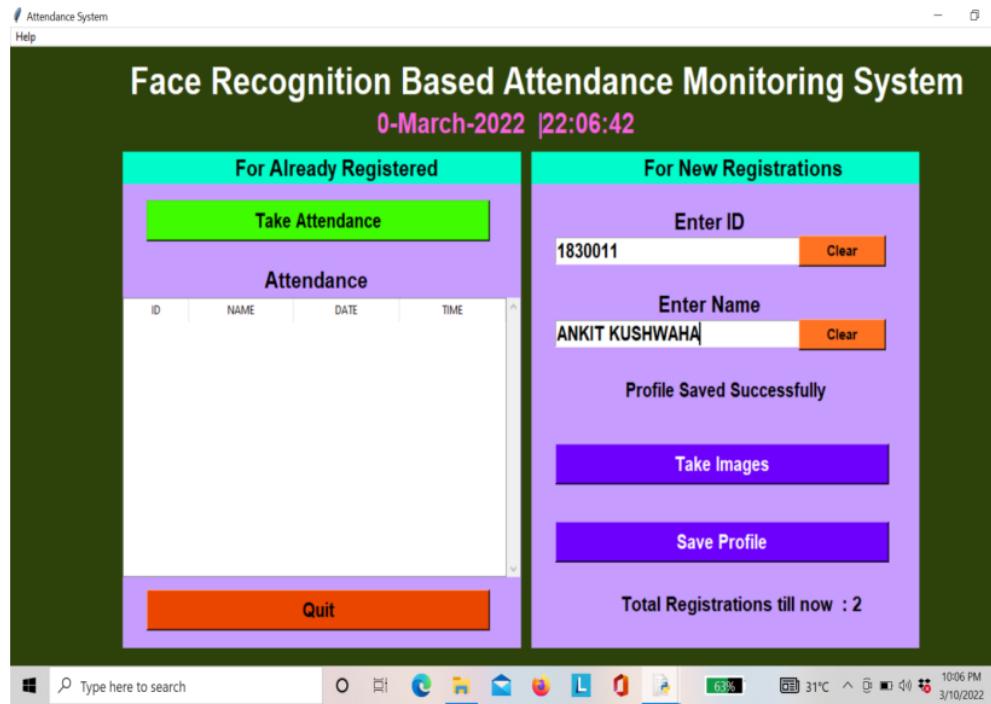


Figure 13.19- Profile Saved Successfully

The screenshot shows a WPS Office spreadsheet window titled 'StudentDetails.csv'. The data is organized into columns A through M. Row 8 contains '3' in column A and 'Ankit' in column E. Row 10 contains '4' in column A and 'ankit' in column E. Row 12 contains '5' in column A and 'ankit' in column E. Row 14 contains '6' in column A and '1830011' in column D. Row 16 contains '7' in column A and 'amkit' in column E. Row 18 contains '8' in column A and '1830011' in column D. Row 19 contains 'Kushwaha' in column E. Row 22 contains '10' in column A and 'asacs' in column E. Row 24 contains '11' in column A and 'Ankit Kushwaha' in column E. Row 26 contains '12' in column A and '1830011' in column D. Row 28 contains '13' in column A and 'asdhi' in column E. The bottom status bar shows the date as 09-04-2022, time as 12:54, and battery level at 37°C Haze.

Figure 13.20- Student Details saved in Database

Fig.13.21 Shows that after successfully registered then user click the take attendance tab then system again take your images to match the already stored in database images. We import a picture and convert it to encodings for testing, then match the encodings with the stored encodings during training, based on maximal similarity. You acquire the name associated with train encodings when you identify the encoding that matches the test image.



Figure 13.21- System Taking Attendance

Now after this Creating a method that will save attendance as a.csv file.

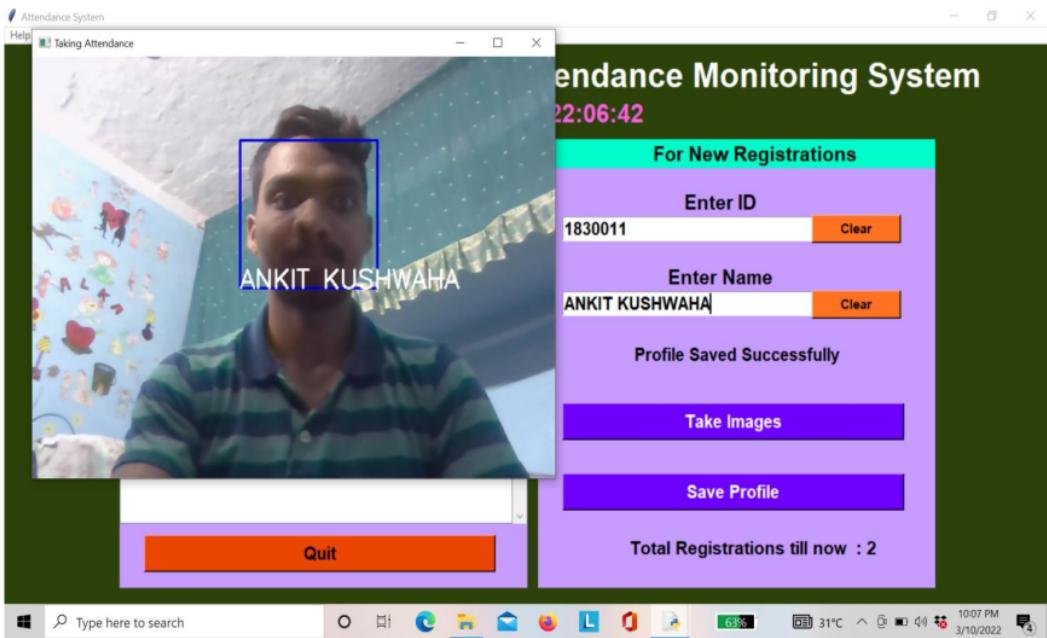


Figure 13.22- Identify your Face with Name

Fig.13.22. After saving your profile, then going for attendance, the system checks or matches your face to store in the database to identify your face, and after identifying your face, then it shows your name automatically.

If unknown. If your face is not found in the database, then it shows unknown, which means you didn't register your name with your id and your profile isn't saved.

At this point, we convert the train image into encodings and save the encodings alongside the person's given name for that image.

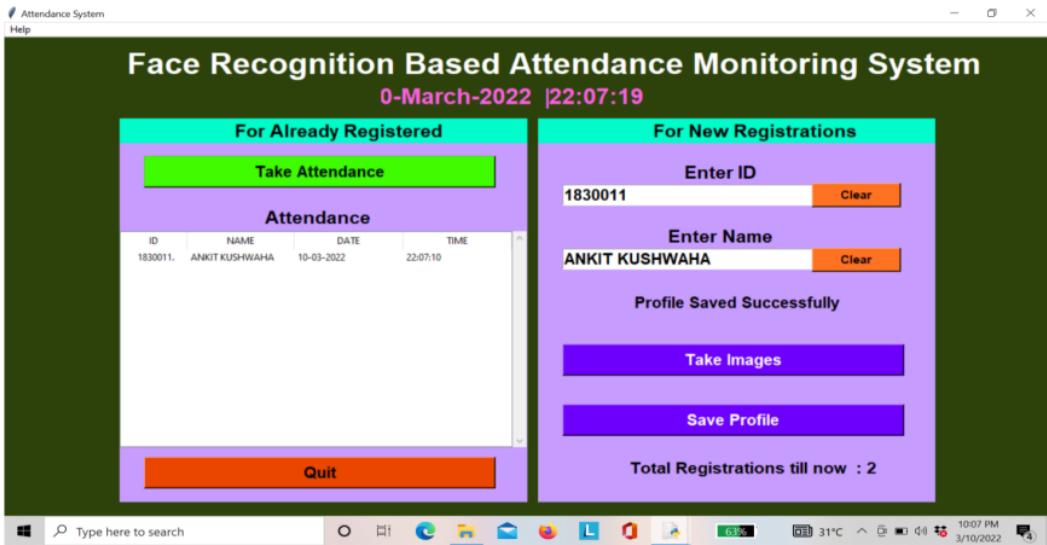


Figure 13.23- Registered In sheet

Fig.13.24. After taking attendance, it shows your id, name, date, and time, also in the database in the form of an excel file.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Id	Name		Date		Time												
2																		
3	1830011	Ankit Kushwaha		2/2/2022		12:16:50												
4																		
5	11	Ankit		2/2/2022		12:33:36												
6																		
7																		
8																		
9																		
10																		
11																		
12																		
13																		
14																		
15																		
16																		
17																		
18																		
19																		
20																		

Figure 13.24- Attendance sheet

Fig.13.24 shows your attendance report with name , id , date and time saved in excel form.

In this Implementation we explained how to use the face recognition library to develop a face recognition system and created an attendance system. For the face recognition attendance system, you can use Tkinter or PyQt to create a GUI.

## CHAPTER 14

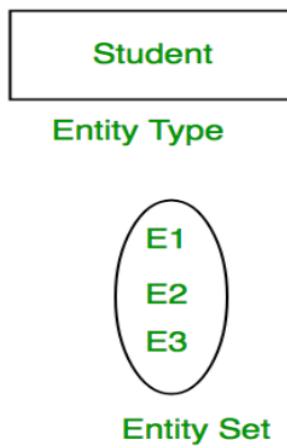
### MODEL ANALYSIS

4  
E-R model is used to model the logical value of system to represent the overall logical structure of the database.

It uses 3 concepts:  
Entities set  
Relationship set  
Attributes

#### 14.1 ENTITY SET

- Any object that has physical existence is called an entity.  
Example: student,faculty etc.
- Set of entities of same type is called **entity set**.  
Example: Set of students



**Figure 14.1** Entity set

#### 14.2 ATTRIBUTES

- These are the properties or characteristics of entities.  
Example: roll no,name,CGPA etc. of student

9

### **KEY ATTRIBUTE:**

The key attribute is the attribute that uniquely identifies each entity in the entity set. Each student's Roll No will, for example, be unique. In the ER diagram, the important feature is shown as an oval with underlying lines.

9

### **COMPOSITE ATTRIBUTE:**

A composite attribute is an attribute that is made up of several other attributes. The Address element of the student Entity type, for example, includes Streets, Town, District, and Country. The composite attribute is represented as an oval made up of ovals in the ER diagram.

### **MULTIVALUED ATTRIBUTE:**

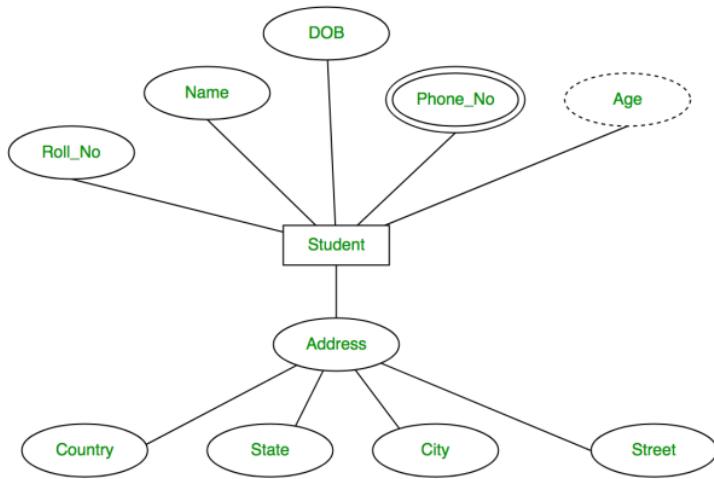
4

For a given entity, an attribute has several values. Phone No, for example (can be more than one for a given student). The double oval in the ER diagram represents a multivalued attribute.

### **DERIVED ATTRIBUTE:**

4

The term derived attribute refers to an attribute that can be derived from other entity type attributes. For instance, consider the age of a person (can be derived from date of birth). The dashed oval in the ER diagram represents a derived attribute.



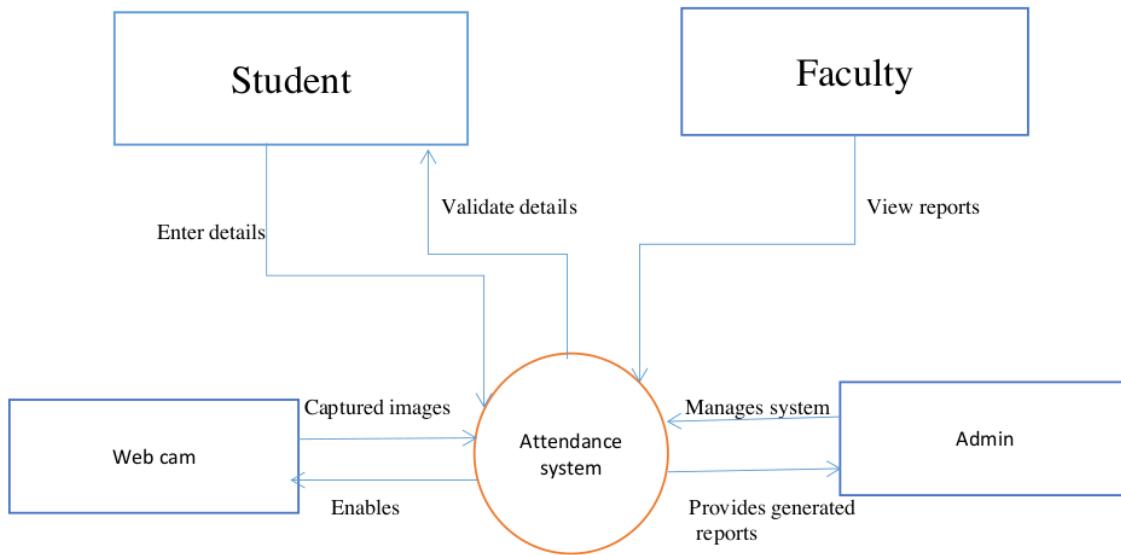


Figure 14.2 - Flowchart depicting the workflow of the attendance system

The above flowchart depicts the entire workflow of attendance system.

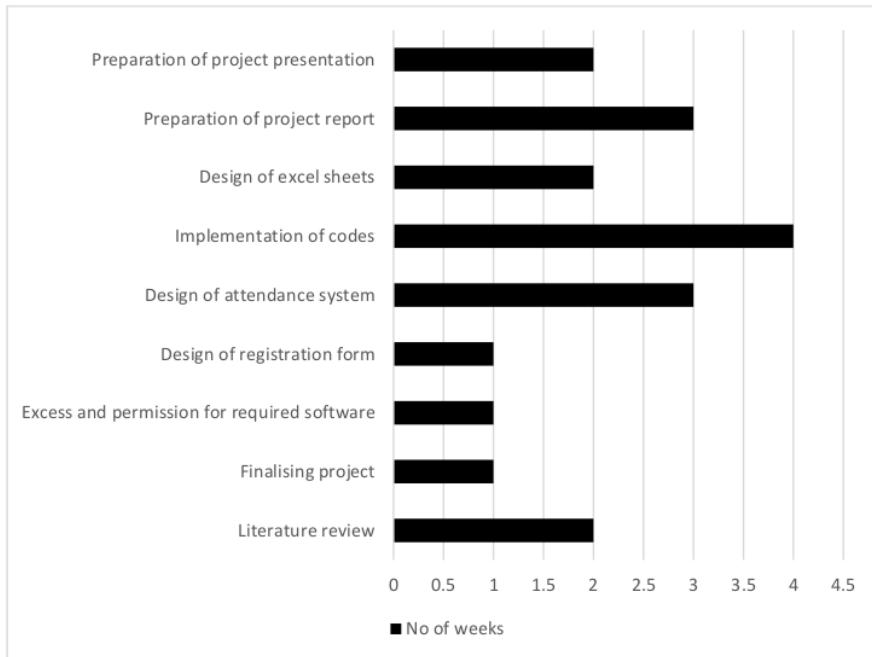
- Web cam is used to capture images.
- Admin manages the system and the attendance system simultaneously provides generated reports.
- Student enters the details and the system validates the details.
- At last faculty views the reports.

**33  
PROJECT PLANNING AND MANAGEMENT**

<b>Activity</b>	<b>Starting week</b>	<b>Number of weeks</b>
Literature review	3 <sup>rd</sup> week of July	2
Finalising project	1 <sup>st</sup> week of August	1
Excess and permission for required software	2 <sup>nd</sup> week of August	1
Design of registration form	2 <sup>nd</sup> week of September	1
Design of attendance system	1 <sup>st</sup> week of October	3
Implementation of the codes	2 <sup>nd</sup> week of January	4
Design of excel sheet to store the database of system	3 <sup>rd</sup> week of February	2
Preparation of project report	2 <sup>nd</sup> week of March	3
Preparation of project presentation	1 <sup>st</sup> week of April	2

## **GANTT CHART**

The gantt chart is shown below:



## CONCLUSION

72  
The major goal of this project is to create a reliable class attendance system based on facial recognition. A student's attendance will be tracked by the system using his or her facial Id. The face will be detected and recognised using the computer's webcam. It will mark the student's attendance and update it in the system after it recognises his or her face. Keeping in mind the day-to-day demands and wants of the modern school system, we chose the Automated attendance system utilising facial recognition theme. Because of technological breakthroughs, we were able to think outside the box and come up with some ideas that will not only change the future, but will also be extremely beneficial. Education is a very important thing a person can obtain, since it is one of the foundations for a good lifestyle and will undoubtedly raise the living community's standards. The participation and involvement in schools, colleges, and universities is something that our current educational system needs. Rather than attending class and studying, many prefer to skip class to keep themselves occupied with gadgets or games. Low attendance causes pupils to be unable to gain the knowledge that they are expected to gain, which is critical for them and can lead to a better and brighter future.

## REFERENCES

- [1] Zou, Y., & Zhou, W. (2019). Automatic seam detection and tracking system for robots based on laser vision. *Mechatronics*, 63, 102261.
- [2]. Raj, A. A., Shoheb, M., Arvind, K., & Chethan, K. S. (2020, June). Face Recognition Based Smart Attendance System. In 2020 International Conference on Intelligent Engineering and Management (ICIEM) (pp. 354-357). IEEE.
- [3]. Saabith, A. S., Fareez, M. M. M., & Vinothraj, T. (2019). Python current trend applications-an overview. *International Journal of Advance Engineering and Research Development*, 6(10).
- [4] Taori, P., & Dasararaju, H. K. (2019). Introduction to Python. In *Essentials of Business Analytics* (pp. 917-944). Springer, Cham.
- [5] Chollet, F. (2021). Deep learning with Python. Simon and Schuster.
- [6] Reiz, A. N., de la Hoz, M. A., & García, M. S. (2019). Big data analysis and machine learning in intensive care units. *Medicina Intensiva (English Edition)*, 43(7), 416-426.
- [7] Gollapudi, S. (2019). OpenCV with Python. In *Learn Computer Vision Using OpenCV* (pp. 31-50). Apress, Berkeley, CA.
- [8] Zadka, M. (2019). Installing Python. In *DevOps in Python* (pp. 1-6). Apress, Berkeley, CA.
- [9] Rana, Y. (2019). Python: Simple though an Important Programming language. *International Research Journal of Engineering and Technology (IRJET)*, 6(2), 1856-1858.
- [10] Pejović, P. (2019). Application of python programming language in measurements. *Facta Universitatis, Series: Electronics and Energetics*, 32(1), 1-23.
- [11] Milliken, C. P. (2020). Python Basics. In *Python Projects for Beginners* (pp. 21-46). Apress, Berkeley, CA.
- [12] Cui, S., Zhao, G., Dai, Z., Wang, L., Huang, R., & Huang, J. (2021). PYInfer: Deep Learning Semantic Type Inference for Python Variables. *arXiv preprint arXiv:2106.14316*.
- [13] Syben, C., Michen, M., Stimpel, B., Seitz, S., Ploner, S., & Maier, A. K. (2019). PYRO-NN: Python reconstruction operators in neural networks. *Medical physics*, 46(11), 5110-5115.
- [14] Chityala, R., & Pudipeddi, S. (2020). *Image processing and acquisition using Python*. Chapman and Hall/CRC.

- [15] Python, R. (2019). Python. Python Releases for Windows, 24.
- [16] Moruzzi, G. (2020). Python basics and the interactive mode. In Essential Python for the Physicist (pp. 1-39). Springer, Cham.
- [17] Yarlagadda, R. T., Surya, L., & Patel, D. M. (2021). Python for Beginners. Python for beginners, REDSHINE Publications (2021), <http://redshine.co.in/product/9798741059081>.
- [18] Trung, T. N. PYTHON PROGRAMMING AND SCIENTIFIC COMPUTATION.
- [19] Hill, C. (2020). Learning scientific programming with Python. Cambridge University Press.
- [20] Pine, D. J. (2019). Introduction to Python for science and engineering. CRC Press.
- [21] Verma, O. P. (2019). Python Membership Operators.
- [22] Novak, N. (2019). Python if statements.
- [23] Verma, O. P. (2019). Lecture-05: Python Operators.
- [24] Yudin, A. (2021). Getting Started with Python. In Basic Python for Data Management, Finance, and Marketing (pp. 1-47). Apress, Berkeley, CA.
- [25] Taori, P., & Dasararaju, H. K. (2019). Introduction to Python. In Essentials of Business Analytics (pp. 917-944). Springer, Cham.
- [26] Moruzzi, G. (2020). Python basics and the interactive mode. In Essential Python for the Physicist (pp. 1-39). Springer, Cham.
- [27] McKinney, W. (2015). Pandas, python data analysis library. URL <http://pandas.pydata.org>.
- [28] Lemenkova, P. (2019). Processing oceanographic data by Python libraries NumPy, SciPy and Pandas. Aquatic Research, 2(2), 73-91.
- [29] Chen, D. Y. (2017). Pandas for everyone: Python data analysis. Addison-Wesley Professional.
- [30] Nelli, F. (2018). The pandas Library—An Introduction. In Python Data Analytics (pp. 87-139). Apress, Berkeley, CA.
- [31] Lee, W. M. (2019). Python machine learning. John Wiley & Sons.
- [32] Chromiński, K., Benko, Ł., Hernández-Figueroa, Z. J., González-Domínguez, J. D., Rodríguez-del-Pino, J. C., & Přichystal, J. (2021). Python Fundamentals.

- [33] Hunt, J. (2019). Functions in Python. In A Beginners Guide to Python 3 Programming (pp. 117-131). Springer, Cham.
- [34] Rajagopalan, G. (2021). Getting Familiar with Python. In A Python Data Analyst's Toolkit (pp. 1-43). Apress, Berkeley, CA.
- [35] Sundnes, J. (2020). Introduction to Scientific Programming with Python (p. 148). Springer Nature.
- [36] Baudart, G., Hirzel, M., Kate, K., Mandel, L., & Shinnar, A. (2019, June). Machine learning in Python with no strings attached. In Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages (pp. 1-9).
- [37] Singh, P., & Manure, A. (2019). Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python. Apress.
- [38] Stančin, I., & Jović, A. (2019, May). An overview and comparison of free Python libraries for data mining and big data analysis. In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 977-982). IEEE.
- [39] Oliphant, T. E. (2006). A guide to NumPy (Vol. 1, p. 85). USA: Trelgol Publishing.
- [40] Idris, I. (2015). NumPy: Beginner's Guide. Packt Publishing Ltd.
- [41] King, D. E. (2009). Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10, 1755-1758.
- [42] Beel, J., Gipp, B., Langer, S., Genzmehr, M., Wilde, E., Nürnberg, A., & Pitman, J. (2011, June). Introducing Mr. DLib, a machine-readable digital library. In Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries (pp. 463-464).
- [43] Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 478-482). IEEE.
- [44] Suwarno, S., & Kevin, K. (2020). Analysis of face recognition algorithm: Dlib and opencv. Journal Of Informatics And Telecommunication Engineering, 4(1), 173-184.
- [45] Zhang, D., Li, J., & Shan, Z. (2020, November). Implementation of Dlib Deep Learning Face Recognition Technology. In 2020 International Conference on Robots & Intelligent System (ICRIS) (pp. 88-91). IEEE.

- [46] Wang, J., & Li, Z. (2018, July). Research on face recognition based on CNN. In IOP Conference Series: Earth and Environmental Science (Vol. 170, No. 3, p. 032110). IOP Publishing
- [47] Brownlee, J. (2019). Deep learning for computer vision: image classification, object detection, and face recognition in python. Machine Learning Mastery.
- [48] Hossain, R., Rahman, M., & Tania, O. A. (2019). Image Capturing and Automatic Face Recognition.
- [49] Qiao, S., & Ma, J. (2018). A face recognition system based on convolution neural network. In 2018 Chinese Automation Congress (CAC) (pp. 1923-1927). IEEE.
- [50] Huang, S., & Luo, H. (2020, July). Attendance System Based on Dynamic Face Recognition. In 2020 International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 368-371). IEEE.
- [51] Khan, M., Chakraborty, S., Astya, R., & Khepra, S. (2019, October). Face Detection and Recognition Using OpenCV. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 116-119). IEEE.
- [52] Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 478-482). IEEE.
- [53] Puthea, K., Hartanto, R., & Hidayat, R. (2020, May). The Attendance Marking System based on Eigenface Recognition using OpenCV and Python. In Journal of Physics: Conference Series (Vol. 1551, No. 1, p. 012012). IOP Publishing.
- [54] Lukas, S., Mitra, A. R., Desanti, R. I., & Krisnadi, D. (2016, October). Student attendance system in classroom using face recognition technique. In 2016 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1032-1035). IEEE.
- [55] Bai, K., Sreemae, K., Sairam, K., Praveen Kumar, B., & Saketh, K. (2021). A survey on real-time automated attendance system. In Proceedings of International Conference on Advances in Computer Engineering and Communication Systems (pp. 473-480). Springer, Singapore.
- [56] Devadiga, A., Sharma, A. K., & Sen, A. (2020). Face Recognition Based Attendance System (Doctoral dissertation, CMR Institute of Technology, Bangalore).
- [57] Bhat, A., Rustagi, S., Purwaha, S. R., & Singhal, S. (2020, July). Deep-learning based group-photo Attendance System using One Shot Learning. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 546-551). IEEE.
- [58] Chan, C. C. K., & Chen, C. C. (2020, August). Continuous Real-time Automated Attendance System using Robust C2D-CNN. In 2020 3rd IEEE

International Conference on Knowledge Innovation and Invention (ICKII) (pp. 96-99). IEEE.

- [59] Sawhney, S., Kacker, K., Jain, S., Singh, S. N., & Garg, R. (2019, January). Real-time smart attendance system using face recognition techniques. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 522-525). IEEE.
- [60] Malhis, L. M., & Samara, S. (2011). AAS (automated attendance system).
- [61] Hirpara, J., Shah, M., Nanda, D., & Kanani, P. (2021, October). Automated Attendance System, Mask Detection and Social Distancing Violation Tracker for Post Covid Scenarios. In 2021 2nd Global Conference for Advancement in Technology (GCAT) (pp. 1-8). IEEE.
- [62] Al-Amoudi, I., Samad, R., Abdullah, N. R. H., Mustafa, M., & Pebrianti, D. (2022). Automatic Attendance System Using Face Recognition with Deep Learning Algorithm. In *Proceedings of the 12th National Technical Seminar on Unmanned System Technology 2020* (pp. 573-588). Springer, Singapore..
- [63] Li, Y., Guo, K., Lu, Y., & Liu, L. (2021). Cropping and attention based approach for masked face recognition. *Applied Intelligence*, 51(5), 3012-3025.
- [64] Kabir, M. A., & Ghazali, R. (2021). Facial Recognition for Smart Attendance Management System Using Local Binary Patterns Histograms. *Applied Information Technology And Computer Science*, 2(2), 1696-1708.
- [65] Aritonang, M., Hutahaean, I. D., Sipayung, H., & Tambunan, I. H. (2020, September). Implementation of Fingerprint Recognition Using Convolutional Neural Network and RFID Authentication Protocol on Attendance Machine. In *Proceedings of the 2020 10th International Conference on Biomedical Engineering and Technology* (pp. 151-156).
- [66] Sun, X., Zheng, S., & Fu, H. (2020). ROI-attention vectorized CNN model for static facial expression recognition. *IEEE Access*, 8, 7183-7194.
- [67] Minu, M. S., Arun, K., Tiwari, A., & Rampuria, P. (2020). Face recognition system based on haar cascade classifier. *International Journal of Advanced Science and Technology*, 29(5), 3799.
- [68] Tang, J., Su, Q., Su, B., Fong, S., Cao, W., & Gong, X. (2020). Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition. *Computer Methods and Programs in Biomedicine*, 197, 105622.
- [69] Yusof, Y. W. M., Nasir, M. M., Othman, K. A., Suliman, S. I., Shahbudin, S., & Mohamad, R. (2018). Real-time internet based attendance using face recognition system. *International Journal of Engineering & Technology*, 7(3.15), 174-178.

# FACE RECOGNITION ATTENDANCE SYSTEM USING PYTHON

## ORIGINALITY REPORT



## PRIMARY SOURCES

1	<a href="http://origin.geeksforgeeks.org">origin.geeksforgeeks.org</a>	2%
2	<a href="http://www.cours-gratuit.com">www.cours-gratuit.com</a>	2%
3	<a href="http://www.coursehero.com">www.coursehero.com</a>	1%
4	<a href="http://www.geeksforgeeks.org">www.geeksforgeeks.org</a>	1%
5	<a href="http://www.whitehatadda.in">www.whitehatadda.in</a>	1%
6	Submitted to Botswana Accountancy College Student Paper	1%
7	<a href="http://ncert.nic.in">ncert.nic.in</a>	1%
8	Submitted to University of Wales Institute, Cardiff Student Paper	1%
9	Submitted to Info Myanmar College Student Paper	<1%

10	jespublication.com Internet Source	<1 %
11	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1 %
12	docplayer.net Internet Source	<1 %
13	Submitted to Akdeniz University Student Paper	<1 %
14	dokumen.pub Internet Source	<1 %
15	Submitted to University of Huddersfield Student Paper	<1 %
16	pynative.com Internet Source	<1 %
17	Submitted to University of Teesside Student Paper	<1 %
18	www.learnopencv.com Internet Source	<1 %
19	Submitted to Oxford Brookes University Student Paper	<1 %
20	www.iare.ac.in Internet Source	<1 %
21	royallib.com Internet Source	<1 %

22	www.javatpoint.com Internet Source	<1 %
23	Submitted to NCC Education Student Paper	<1 %
24	manualzz.com Internet Source	<1 %
25	pdf4pro.com Internet Source	<1 %
26	Submitted to Liverpool John Moores University Student Paper	<1 %
27	4pps.co Internet Source	<1 %
28	Submitted to University of Adelaide Student Paper	<1 %
29	dbmskhscs12.blogspot.com Internet Source	<1 %
30	"Advance Concepts of Image Processing and Pattern Recognition", Springer Science and Business Media LLC, 2022 Publication	<1 %
31	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
Submitted to University of Florida		

32	Submitted to Indian Institute of Technology, Bombay Student Paper	<1 %
33	Submitted to Indian Institute of Technology, Bombay Student Paper	<1 %
34	pdfcoffee.com Internet Source	<1 %
35	iot.teamprojects.de Internet Source	<1 %
36	Submitted to Australian College of Business and Technology Student Paper	<1 %
37	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	<1 %
38	psasir.upm.edu.my Internet Source	<1 %
39	Submitted to Monash University Student Paper	<1 %
40	ebin.pub Internet Source	<1 %
41	Submitted to Ibri College of Technology Student Paper	<1 %
42	Submitted to Middlesex University Student Paper	<1 %

43	Submitted to Noroff University College Student Paper	<1 %
44	Submitted to University of Oklahoma Health Science Center Student Paper	<1 %
45	veewom.com Internet Source	<1 %
46	www.slideshare.net Internet Source	<1 %
47	Submitted to University of Hertfordshire Student Paper	<1 %
48	medium.com Internet Source	<1 %
49	Rohit Rastogi, Abhinav Tyagi, Himanshu Upadhyay, Devendra Singh. "Algorithmic Analysis of Automatic Attendance System Using Facial Recognition", International Journal of Decision Support System Technology, 2022 Publication	<1 %
50	Submitted to The NorthCap University, Gurugram Student Paper	<1 %
51	Submitted to Yeshwant Rao Chavan College of Engineering Student Paper	<1 %

52	modular.math.washington.edu Internet Source	<1 %
53	www.wikizer.com Internet Source	<1 %
54	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
55	Submitted to Coventry University Student Paper	<1 %
56	Submitted to Gitam University Student Paper	<1 %
57	Submitted to Guru Jambheshwar University of Science & Technology Student Paper	<1 %
58	Submitted to Asian Institute of Chartered Bankers Student Paper	<1 %
59	www.w3schools.com Internet Source	<1 %
60	Submitted to South Birmingham College Student Paper	<1 %
61	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
62	collegeburden.com Internet Source	<1 %

63	content.techgig.com	<1 %
Internet Source		
64	www.abe.pl	<1 %
Internet Source		
65	Submitted to Open University of Mauritius	<1 %
Student Paper		
66	Submitted to Palm Beach Currumbin State High School	<1 %
Student Paper		
67	Submitted to SASTRA University	<1 %
Student Paper		
68	www.emeraldinsight.com	<1 %
Internet Source		
69	www.interfaceware.com	<1 %
Internet Source		
70	"Excel 2007", Springer Science and Business Media LLC, 2007	<1 %
Publication		
71	dotnetcode4programmer.blogspot.com	<1 %
Internet Source		
72	publisher.uthm.edu.my	<1 %
Internet Source		
73	www.econstor.eu	<1 %
Internet Source		

74

[www.seanmcp.com](http://www.seanmcp.com)

Internet Source

<1 %

75

[www.upgrad.com](http://www.upgrad.com)

Internet Source

<1 %

---

Exclude quotes      On

Exclude bibliography      On

Exclude matches      < 10 words