1. INTRODUCTION

In the fast-paced technological landscape of the 21st century, traditional attendance tracking methods have become outdated and inefficient. The Face Recognition Attendance System represents a cutting-edge solution that leverages the power of facial recognition technology to streamline and enhance the attendance tracking process. This project aims to revolutionize the way organizations manage attendance by providing a secure, accurate, and automated system.

1.1 Description of Project

The technology aims in imparting a tremendous knowledge oriented technical innovation these days. Generally, in the classroom the attendance was taken by the teachers manually at the beginning and ending of the class. The problem with this approach is that it requires some time to take attendance and the manual process will have chances to make mistakes in most of the cases. So, we are introducing the concept of Face Recognition Based Attendance system, the main objective the proposed system is to allot attendance to the students using face recognition-based algorithms to achieve fail proof attendance system.

This project aims to automate the traditional attendance system where the attendance is marked manually. It also enables an organization to maintain its records attendance digitally. Digitalization of the system would also help in better visualization of the data using graphs to display the no. of employees present today, total work hours of each employee. Its added features serve as an efficient upgrade and replacement over the traditional attendance system.

Face detection is used for many applications for the identification of human faces in digital images or video. It is defined as specific case of object-class detection; where it is used to find the locations and sizes of all objects in an image that belong to a given class.

Face Recognition is a form of biometric software that maps an individual's facial features mathematically and stores the data as a face print. The software consists of Deep Learning algorithms to compare a live capture or digital image to the stored face print in order to verify an individual's identity.

1.2 Object of project

The proposed system will reduce the paperwork where attendance will no longer involve any manual recording. The new system will also reduce the total time needed to do attendance recording. The new system will acquire individual attendance by means of facial recognition to secure data accuracy of the attendance. The following are objectives of the project:

- To develop a portable Smart Attendance System which is handy and self-powered.
- To ensure the speed of the attendance recording process is faster than the previous system.
- To detect unique faces with the help of computer's camera
- Able to recognize the face of an individual accurately based on the face database.
- Provide a user-friendly interface for admins to access the attendance.
- Allow new students or to store their faces in the database by using a GUI

1.3 Motivation

The motivation behind the Face Recognition Based Attendance System project is to:

1. Enhanced Security: Facial recognition systems contribute to improved security by providing a biometric authentication method that can be difficult to forge or manipulate.

2. Streamlined Identification: These systems facilitate quick and seamless identification, reducing the need for manual verification processes and enhancing efficiency.

3. User Experience: Integration of facial recognition in devices, such as smartphones, offers users a convenient and secure way to unlock devices or access personalized features.

4. Research Advancements: Development of facial recognition technology contributes to advancements in computer vision, artificial intelligence, and machine learning, driving progress in these fields.

5. Law Enforcement: Facial recognition is employed by law enforcement agencies for identifying and tracking individuals, aiding in crime prevention and investigation.

6. Access Control: Businesses and organizations use facial recognition for access control, ensuring authorized personnel can enter restricted areas.

7. Public Safety: Deployment in public spaces can enhance safety by identifying potential threats or individuals of interest.

8. Biometric Verification: As a biometric modality, facial recognition offers a unique and non-intrusive way to verify and authenticate individuals.

9. Efficiency in Transactions: Some industries leverage facial recognition to expedite processes, such as payments or check-ins, providing a faster and more convenient experience for users.

10. Health and Wellness Monitoring: Facial recognition can be applied in healthcare for monitoring patient conditions or tracking wellness metrics.

Each of these motivations aligns with specific use cases and benefits associated with the implementation of facial recognition systems.

1.4 Modules

Every project is made up of several modules using modules in a project helps in reusability of code and makes the code easier to read. Face Recognition System For Smart Attendance is no different. It consist of 4 modules, these are divided by the functionality they contain different functions of the project to run, these modules are :-

1.4.1 Homepage

The homepage module contains all the code which is written for the graphical user interface. This module has different sections for the different functionality.

1.4.2 Take picture

This is the smallest module of this project. The only functionality it offers is that it turns the camera on to click the picture of a student and save it to the Training Image folder.

1.4.3 Recognize face

Recognize face module is the module with main functionality of reorganization the face of the student who is standing/sitting in front of the camera and calling the function in other module to mark the attendance for the recognized student

1.4.4 Data manager

The modulus is responsible for accessing the data, saving it, deleting it. There are 2 types of data in this project. 1st is relational data, and 2nd is image data. There are two separate folder that store auto generated Excel file for these 2 types of data. Data manager module is the responsible of managing both of these Excel file folders.

2. LITERATURE SURVEY

2.1 Technologies Learned

This project is written completely in python so first thing to learn was to get familiar with the syntax of the python programming language and learn basic of data types, conditions statement, functions, modules etc.

Second thing to learn was various python libraries and modules.

Down below is mentioned modules whose working knowledge was acquired before starting the development of this project :-

- ✓ opencv
- ✓ tkinter
- ✓ datetime
- ✓ csv
- ✓ times
- ✓ os

The knowledge is acquired in the need to know basic, so it may or may not be complete and accurate.

2.2 Basic Concepts Used

Face Recognition Based Attendance System, we use some basic concepts that make the system work smoothly:

i. Facial Recognition Technology

Easy Explanation: Recognizing people by their faces.

How it's Used: Captures and analyzes unique features of a face to tell who it is.

ii. Biometric Data

Easy Explanation: Special information unique to each person.

How it's Used: Collects and stores unique facial features for each student.

iii. Image Processing

Easy Explanation: Working with pictures to make them better for the computer to understand.

How it's Used: Cleans up and improves facial images for accurate recognition.

iv. Machine Learning Algorithms

Easy Explanation: Computers that learn and get better at tasks over time.

How it's Used: Trains the system to recognize faces more accurately as it sees more examples.

v. Neural Networks

Easy Explanation: Computer models inspired by how our brain works.

How it's Used: Helps the system understand and remember facial features.

vi. User Interface (UI)

Easy Explanation: The way we interact with a computer or system.

How it's Used: Creates an easy and friendly way for teachers and administrators to use the attendance system.

vii. Real-time Processing

Easy Explanation: Doing things immediately as they happen.

How it's Used: Records attendance instantly, so it's always up-to-date.

3. REQUIREMENT AND ANALYSIS

3.1 Hardware Specifications

CPU – Intel core **i3** 7th gen

Ram – 1 GB of free ram

Storage - Depends on number of students

Webcam - 720p

3.2 Software Specification

Operating System – Windows 10 or above

IDE - Visual Studio Code

Programming Language – python 3.9

3.3 Technology Introduction

3.3.1 python 3.9

python is high level, object oriented, interpreted programming language which can be used for plenty of tasks including machine learning, web development, app development. Python provides multiple modules which are used in this project to get different functionality .

3.3.2 tkinter

This is the standard GUI module of python. It is used to code graphical user interface in this project .

3.3.3 opencv

Opencv module or cv2 is used to use camera from within the python code, to take picture and save them.

3.3.4 os

os module of python is used to access the directories directly from the code. It is used for file handling in this project.

3.3.5 datetime

In this project datetime module is responsible to find out what time it is by checking the time of the computer system and it also helps to take attendance with the timestamp.

3.3.6 face recognition

This is responsible for detecting the face of the student and matching it from the Training Image to find which student is standing/sitting in front of camera.

3.3.7 Visual Studio Code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.
Visual Studio Code is a free coding editor that helps you start coding quickly. Use it to code in any programming language, without switching editors. Visual Studio Code has support for many languages, including Python, Java, C++, JavaScript, and more.

3.3.8 Windows 11

The windows 11 is the latest operating system developed by Microsoft which is the heart of this project. The project in completely built and tested on this operating system.

4. DESIGN

4.1 Design Strategy

4.1.1 Block Diagram

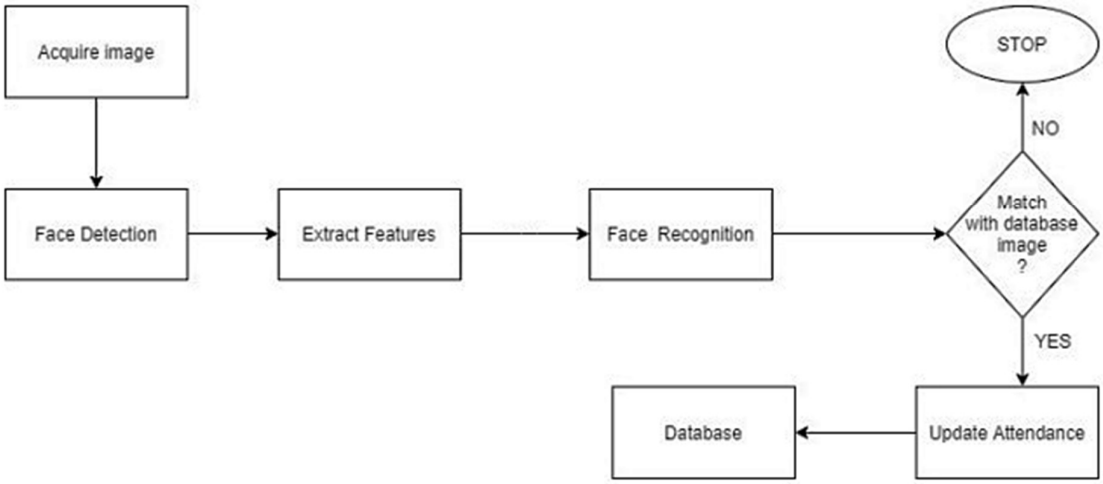


Figure - 4.1.1 Block Diagram

4.1.2 ER Diagram

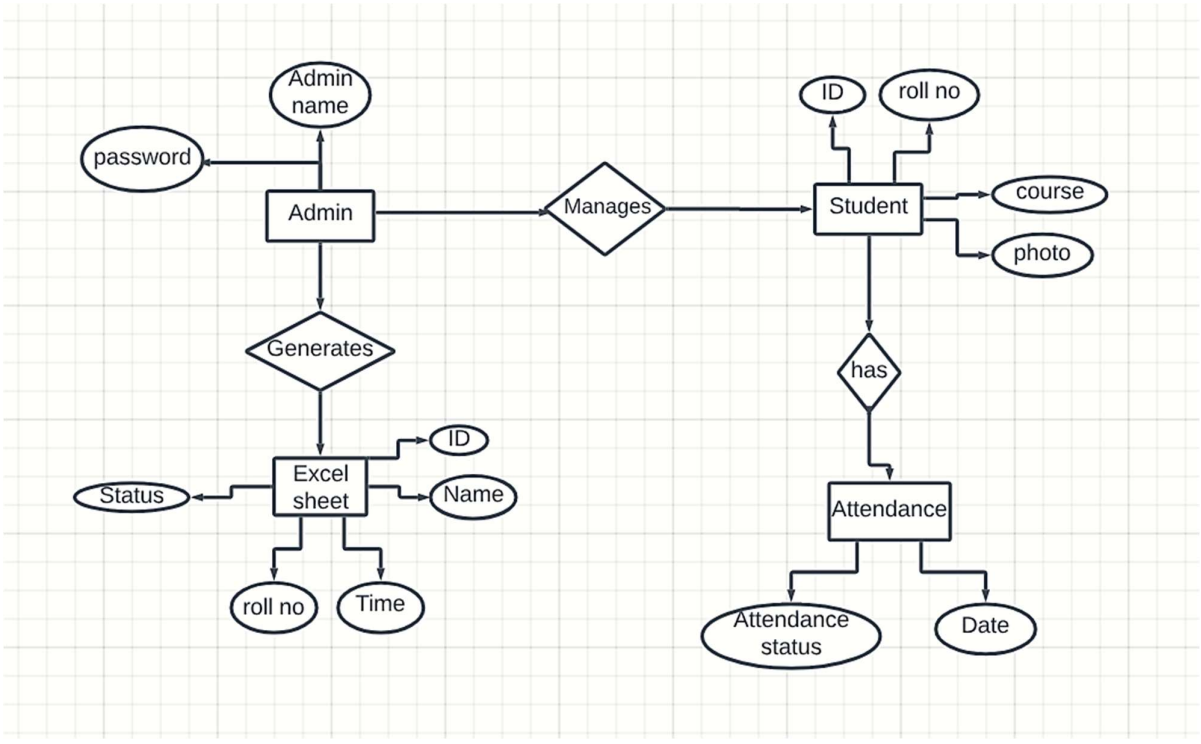


Figure – 4.1.2 ER Diagram

4.1.3 Data Flow Diagram

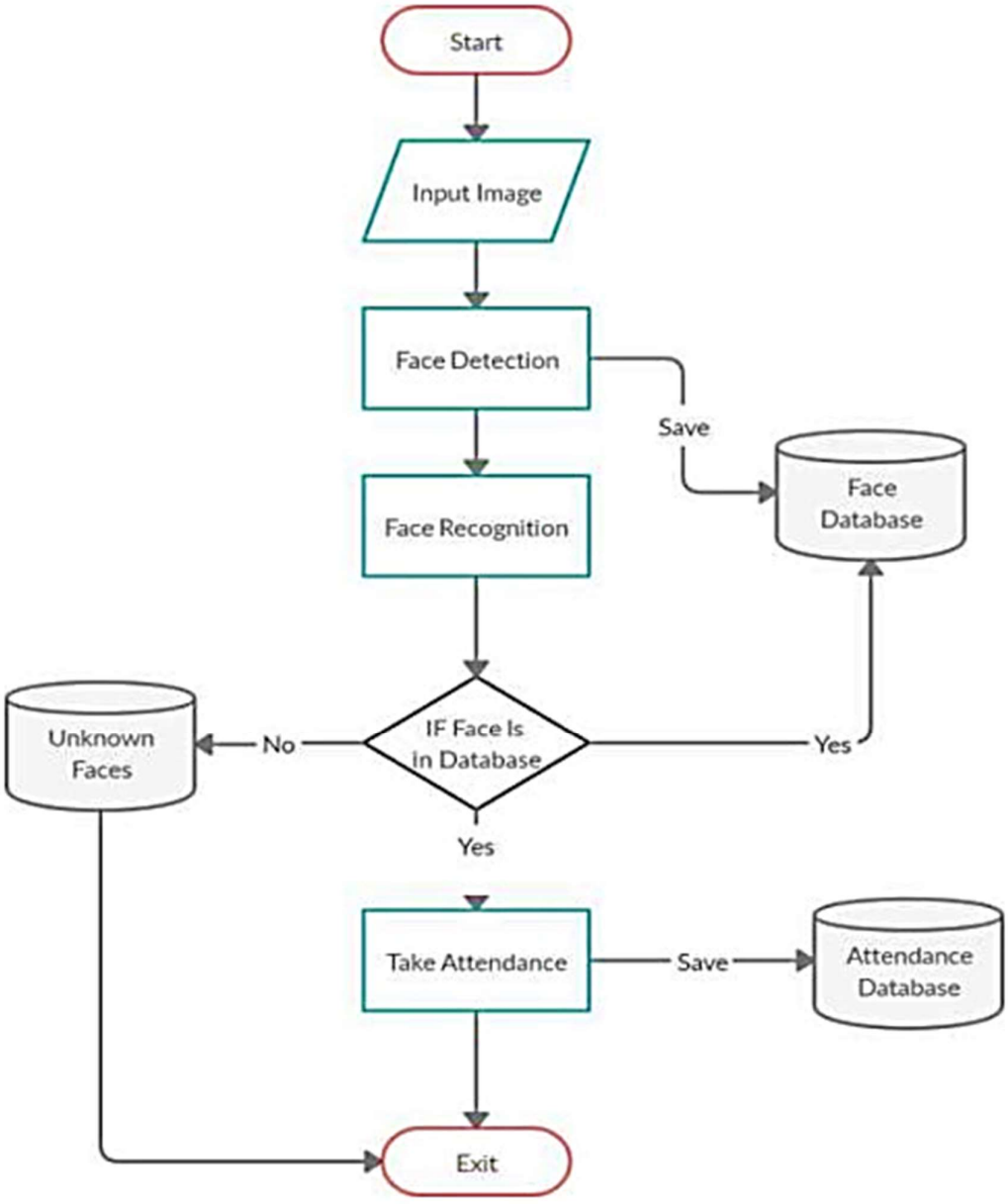


Figure – 4.1.3 Data Flow Diagram

4.2 Application Diagrams

4.2.1 Use Case Diagram

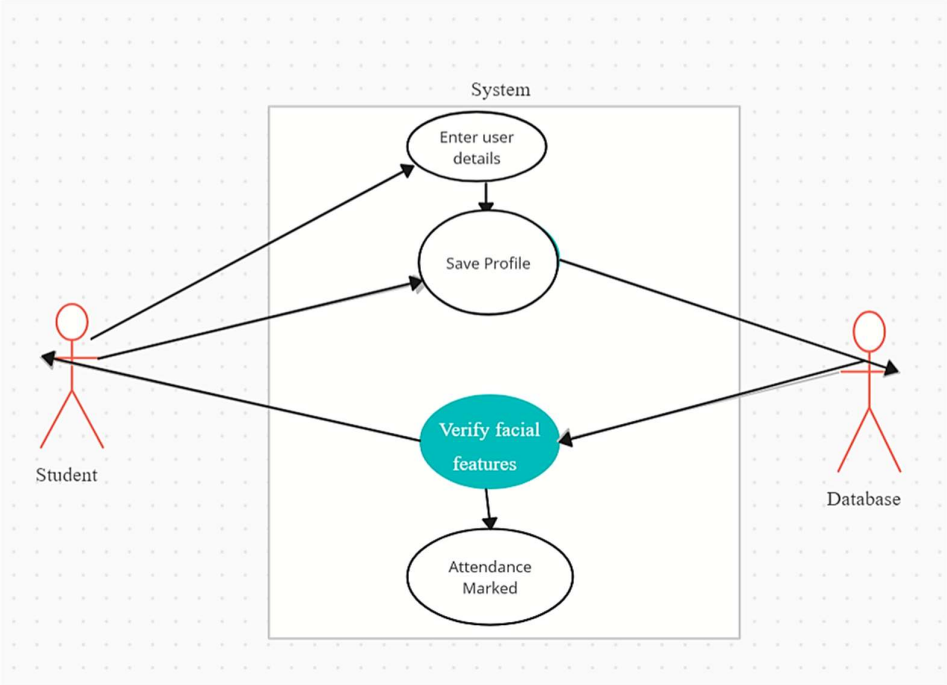


Figure 4.2.1 Use Case Diagram

5. CODING

```
############ IMPORTING LIBRARY #################
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
```

```
################## 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 of MCA Department")
window.configure(background='#172540')

frame1 = tk.Frame(window, bg="#9cbfff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#9cbfff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)
```

```
message3 = tk.Label(window, text="Face Recognition Based Attendance System" ,fg="white",bg="#3c732d"
,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="#ff61e5",bg="#172540" ,width=75
,height=1,font=('comic', 17, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="#ff61e5",bg="#172540" ,width=55 ,height=1,font=('comic', 17, 'bold'))
clock.pack(fill='both',expand=1)
tick()

head2 = tk.Label(frame2, text="                For New Registrations                ", fg="black",bg="#48d9bc"
,font=('comic', 17, ' bold ') )
head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text="                For Already Registered                ", fg="black",bg="#48d9bc"
,font=('comic', 17, ' bold ') )
head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter ID",width=20  ,height=1  ,fg="white"  ,bg="#8c2d3d" ,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="white"  ,bg="#8c2d3d" ,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" ,bg="#e0ed4c" ,fg="black"  ,width=39
,height=1, activebackground = "#3ffc00" ,font=('comic', 15, ' bold '))
message1.place(x=7, y=230)

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

lbl3 = tk.Label(frame1, text="Attendance Details",width=20  ,fg="black"  ,bg="#e0ed4c"  ,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 csvFile1:
      reader1 = csv.reader(csvFile1)
      for l in reader1:
         res = res + 1
   res = (res // 2) - 1
   csvFile1.close()
else:
   res = 0
message.configure(text='Total Registrations till now  : '+str(res))
```

```
######################### MENU BAR #################################

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)
```

```
################## TREE VIEW ATTENDANCE TABLE ####################

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
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')
```

```
###################### BUTTONS ###################################

clearButton = tk.Button(frame2, text="Clear", command=clear  ,fg="white"  ,bg="#c42b33"  ,width=11
,activebackground = "white" ,font=('comic', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2  ,fg="white"  ,bg="#c42b33"  ,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, ' bold '))
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, ' bold '))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take Attendance",
command=TrackImages  ,fg="#2c139c"  ,bg="#67ed4c"  ,width=35  ,height=1, activebackground = "white"
,font=('comic', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy  ,fg="white"  ,bg="#eb4600"  ,width=35
,height=1, activebackground = "white" ,font=('comic', 15, ' bold '))
quitWindow.place(x=30, y=450)
```

```
###################### 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)
```

```
###################### END ######################################

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

#################################################################################
```

6. METHODOLOGY

The methodology of a facial recognition system involves a series of steps and techniques designed to detect and identify faces within images or videos. Here's a general overview of the key components and processes involved in a facial recognition system:

1. Data Collection:

Training Data: Gather a large dataset of facial images for training the model. These images should cover a diverse range of poses, expressions, lighting conditions, and backgrounds.

Testing Data: Separate datasets for testing and validation purposes are also needed to evaluate the performance of the trained model.

2. Preprocessing:

Face Detection: Use a face detection algorithm to locate and extract facial regions from images. Common algorithms include Haar cascades, Single Shot Multibox Detector (SSD), or Faster R-CNN.

Normalization: Preprocess the facial images to ensure consistent lighting, scale, and orientation. Normalization helps improve the robustness of the model.

3. Feature Extraction: - Extract features from the preprocessed facial images. Common techniques include Principal Component Analysis (PCA), Local Binary Patterns (LBP), or deep learning-based feature extraction using Convolutional Neural Networks (CNNs).

4. Model Training: Train a machine learning or deep learning model using the extracted features. Popular deep learning architectures for facial recognition include VGG, ResNet, and FaceNet.

Utilize labeled training data to enable the model to learn the relationships between facial features and corresponding identities.

5. Face Recognition: Apply the trained model to new images or videos to recognize and identify faces.

Distance metrics (e.g., Euclidean distance or cosine similarity) are often used to measure the similarity between facial feature vectors.

6. Post-Processing: Implement post-processing techniques to improve the accuracy of the facial recognition system. This may involve filtering out false positives, refining face bounding boxes, or incorporating additional contextual information.

7. Database Integration: Integrate the facial recognition system with a database containing information about individuals. This enables the system to associate recognized faces with corresponding identities.

8. Privacy and Security Considerations: Implement privacy safeguards, such as data encryption and secure storage, to protect individuals' facial data.

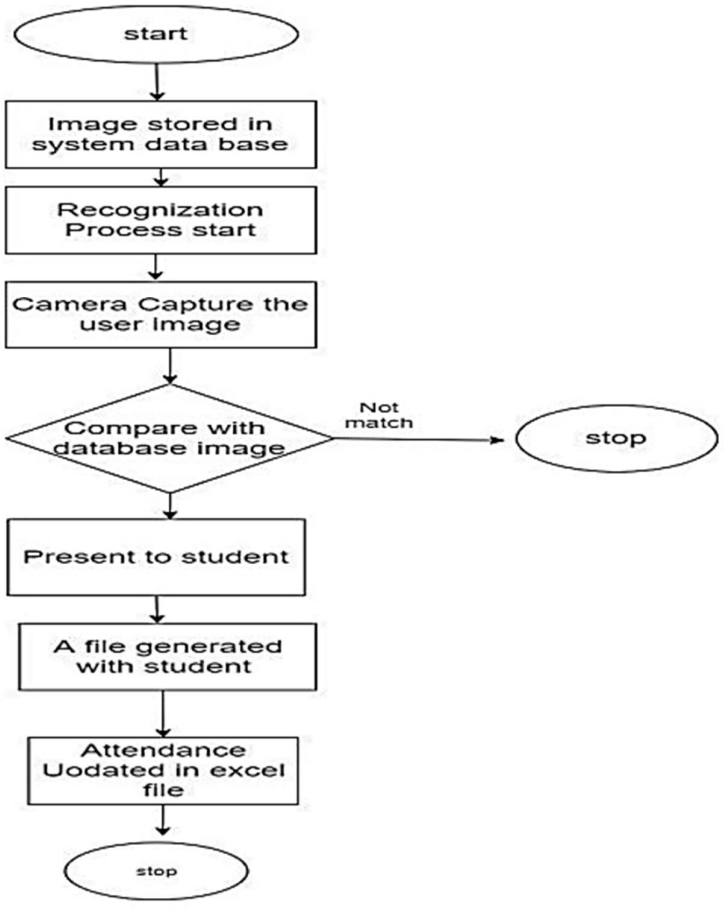
Address ethical concerns related to consent, data usage, and potential misuse of facial recognition technology.

9. Continuous Improvement: Regularly update and retrain the facial recognition model with new data to adapt to changes in the environment, such as variations in lighting conditions and appearance.

10. Evaluation: Assess the performance of the facial recognition system using testing datasets. Common metrics include accuracy, precision, recall, and the F1 score.

It's important to note that facial recognition technology raises ethical and privacy concerns, and its deployment should be done with careful consideration of these issues. Additionally, compliance with local regulations and standards is crucial.

6.1 Flowchart



7. TESTING

Testing was done at various levels

7.1 Unit Testing

All the function were tested by providing input and observing output. This type of testing was done at the time of writing the code. Every function was provided the input and was updated until the desired output was achieved.

7.2 System testing

System testing is the first level of testing where the application was tested as a whole, the goal of this testing was to see whether all the features of the software are working as expected or not.

7.3 Integration Testing

How the function are interacting with each other was tested during integration testing. How the module and function are communicating with each other. If the module are being imported correctly or not, etc were tested during integration testing.

7.4 Acceptance Testing

This type of testing was the last level of testing for this project. For this application was run under real life simulation and the behavior was observed.

8. RESULT

After Testing at all levels, the project dreams to be successfully completed below are some result.

8.1 HOMEPAGE

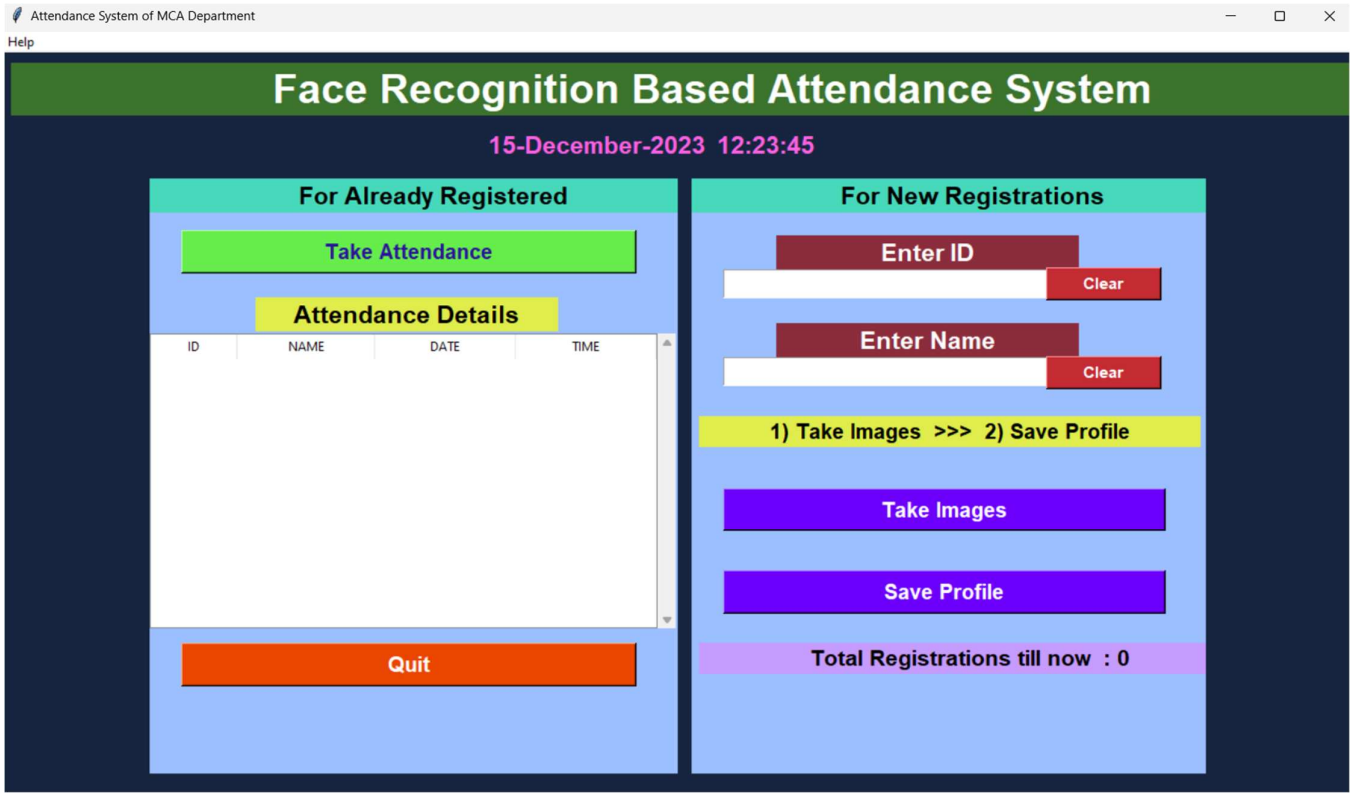


Figure 8.1 Homepage

Budget Analysis

S.N	Particulars	Quantity
1.	No. of hours per day	3 hours
2.	No. of working day per weeks	4 days
3.	Project Period	2 months
4.	No. of programmer	3

9. CONCLUSION

The Face Recognition Based Attendance System project introduces an innovative and efficient solution to traditional attendance tracking in schools. Utilizing facial recognition technology, the system ensures accurate, secure, and real-time attendance records. Its user-friendly interface, commitment to privacy compliance, and adaptability make it a valuable tool for diverse educational environments. The project not only enhances efficiency but also addresses security concerns, promoting a modernized and streamlined approach to attendance management in schools.

Face recognition base attendance system tries to solve a real-world problem by using technology, which is openly available and free to use and distribute. One thing that should be understood that this is not a ready to use product, but a minor project, a proof of concept that the technology that is open source and freely available can be used to solve a real-world problem. Through, the project work and function as intended, but it cannot be deployed as a ready product. There is a lot to change, a lot to an upgrade, many new feature to add, make it more secure, portable and compatible. There is a lot discussed in future scope that can be implemented to make this project of face recognition system A real product which can be used in the organization's, colleges and schools, not just only in UIT RGPV, but other colleges as well.

10. FUTURE ENHANCEMENT

Below are some points that can be improved in the future versions:-

- Better GUI for enhanced user experience.
- Email Notifications of the attendance.
- Subject wise attendance.
- A feature can be added where a student is automatically sent a warning if his attendance is below the threshold.
- Remove duplicate attendance.
- Automatic mail alert/response to the parents regarding the presence and absence of the students can be added.
- The number of training images can be reduced by removing duplicate images of the same person, or images with similar embeddings.
- Make it a live project by hosting it on a cloud provider service such as AWS.

11. BIBLIOGRAPHY

Resources and references used

Books:

1. “Opencv Computer Vision With Python” author Joseph Howse
2. “Facial Recognition a Complete Guide” author Gerardus Blokdyk
3. "Practical Machine Learning and Image Processing" author Himanshu Singh
4. “Hands-on Computer Vision With TensorFlow” author Benjamin Planche
5. “Deep Learning With Python” author Francois Chollet
6. “Fluent OpenCV-Python GUI with SQL Server” author Vivian Siahaan

Links:

1. https://en.wikipedia.org/wiki/Facial_recognition_system
2. https://www.datacamp.com/community/tutorials/face-detection-python-opencv
3. https://realpython.com/face-detection-in-python-using-a-webcam/
4. https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/
5. https://www.geeksforgeeks.org/opencv-python-program-face-detection/
6. https://www.udemy.com/course/python-multiple-face-recognition-using-opencv/
7. https://www.edureka.co/blog/tensorflow-object-detection-tutorial/
8. https://youtu.be/-ZrDjwXZGxI