

FACIAL RECOGNITION ATTENDANCE

MAJOR PROJECT – I

Submitted by

K. Uvan Sankar

17MSS059

Under the Guidance of

Lt. Dr. D. Antony Arul Raj M.Sc.,(CS).,M. Phil., PGDCE., PhD.,

Associate Professor

Department of Software Systems

In partial fulfillment of the requirements for the award of the degree of

MASTER OF SCIENCE IN SOFTWARE SYSTEMS

(Five-years Integrated Course)

of Bharathiar University



DEPARTMENT OF SOFTWARE SYSTEMS

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous college - Affiliated to Bharathiar University

Accredited with 'A' grade by NAAC (3rd Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Coimbatore - 641 014

(December 2020)

DEPARTMENT OF SOFTWARE SYSTEMS

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous college - Affiliated to Bharathiar University

Accredited with 'A' grade by NAAC (3rd Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Coimbatore - 641 014

CERTIFICATE

This is to certify that this project work entitled “**FACIAL RECOGNITION ATTENDANCE**” is a bonafide record of work done by **K. Uvan Sankar (17MSS059)** in partial fulfillment of the requirements for the award of Degree of **Master of Science in Software Systems** (Five year Integrated Course) of Bharathiar University.

Faculty Guide

Head of the Department

Submitted for Viva-Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **K. Uvan Sankar (17MSS059)**, hereby declare that this project work entitled **“FACIAL RECOGNITION ATTENDANCE”** is submitted to PSG College of Arts & Science, Coimbatore in partial fulfillment of the requirements for the award of the degree of Master of Science in Software Systems, is a record of original work done by me under the supervision and guidance of **Lt. Dr. D. Antony Arul Raj M.Sc.,(CS),M. Phil., PGDCE., PhD.**, Associate Professor, Department of Software Systems, PSG College of Arts & Science, Coimbatore.

This report has not been submitted by me for the award of any other Degree/ Diploma/ Associate ship/ Fellowship or any other similar degree to any other university.

Place: Coimbatore

K. Uvan Sankar

Date:

17MSS059

ACKNOWLEDGEMENT

My venture stands imperfect without dedicating my gratitude to a few people who have contributed a lot towards the victorious completion for my project work.

I would like to thank **Thiru L. Gopalakrishnan, Managing Trustee, PSG & Sons Charities**, for providing me a prospect and surroundings that made the work possible.

I take this opportunity to express my deep sense of gratitude to **Dr. T. Kannaian, Secretary**, PSG College of Arts & Science, Coimbatore for permitting and doing the needful towards the successful completion of this project.

I express my deep sense of gratitude and sincere thanks to **Dr. D. Brindha M.Sc., M.Phil., Ph.D., MA (Yoga),** Principal, PSG College of Arts & Science, Coimbatore for her valuable advice and concern on students.

I am very much thankful to **Dr. A. Anguraj, M.Sc., M.Phil., Ph.D.,** Vice Principal (Academics), **Dr. Jayanthi M M.Com., MBA., M.Phil., Ph.D.,** Vice Principal (Student Affairs), **Prof. M Umarani, MBA, M.Phil., Faculty-In-Charge (Student Affairs),** for their support towards my project.

I sincerely thank **Dr. K.V.Rukmani., M.C.A., M.E, Ph.D.,** Head (i/c), Department of Software Systems for her whole hearted help to complete this project successfully by giving valuable suggestions.

I convey my heartiest and passionate sense of thankfulness to my project guide **Lt. Dr. D. Antony Arul Raj M.Sc.,(CS),M. Phil., PGDCE., Ph.D.,** Associate Professor, Department of Software Systems, for her/his timely suggestion which had enable me in completing the project successfully.

I am grateful to **Mr. R. Karthik Kannan**, external guide, **ETOP Technologies**, Chennai, for giving me an expert's insight into the project and for sparing time in his/her hectic agenda, for planning and giving me invaluable guidance and suggestions.

This note of acknowledgement will be incomplete without paying my heartfelt devotion to my parents, my friends and other people, for their blessings, encouragement, financial support and the patience, without which it would have been impossible for me to complete the job.

COMPANY CERTIFICATE



ETOP Technologies Pvt Ltd
Plot Number 4, First Floor
Sridevi Amman Street,
Perumal Nagar Extension,
Old Pallavaram, Chennai,
Tamil Nadu, 600117.
Phone: [+91-44-43849570](tel:+91-44-43849570)

Date: 05/08/2020

INTERNSHIP CONFIRMATION LETTER

Dear Mr. Uvan Sankar.K,

We are pleased to inform you that you are engaged as an AI and ML developer INTERN at ETOP Technologies Pvt Ltd.

The terms of your internship with the company will be as follows:

1. Internship Period will be from 28th June 2020 till 30th November 2020
2. This is an unpaid Internship.

Authorized Signature

A handwritten signature in blue ink, consisting of a large, stylized 'U' followed by a series of loops and a long horizontal stroke.

Website: www.etopdigital.com
Email: sales@etopdigital.com

SYNOPSIS

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

TABLE OF CONTENTS

S.No	CONTENTS	PAGE NO
1	INTRODUCTION	
	1.1 Company Profile	1
	1.2 Project Overview	2
	1.3 Module Description	3
2	SYSTEM ANALYSIS	
	2.1 Existing System	4
	2.2 Proposed System	5
3	SYSTEM CONFIGURATION	
	3.1 Hardware Specification	6
	3.2 Software Specification	6
4	SOFTWARE DESCRIPTION	
	4.1 Front End	7
	4.2 Back End	10
5	SYSTEM DESIGN	
	5.1 System Flow Diagram	11
	5.2 Input Design	13
	5.3 Output Design	14
6	SYSTEM TESTING & IMPLEMENTATION	15
7	SCOPE FOR FUTURE ENHANCEMENT	19
8	CONCLUSION	20
9	BIBLIOGRAPHY	21
10	APPENDIX	
	A. Screenshot	22
	B. Sample Coding	25

1. INTRODUCTION

1.1 COMPANY PROFILE

ETOP Technologies is a team that has a passion for developing and delivering enterprise-grade AI applications. We build software solutions using emerging technologies for start-ups and enterprises. Our goal is to help companies adopt new technologies and simplify complicated issues that arise during technology evolution. Our core competency is in AI/ML, IoT, cloud and mobile.

SERVICES OFFERED

- Machine Learning
- Internet Of Things
- Computer Vision
- Voice Enabled Skills
- Robotic Process Automation
- Full Stack Development

1.2 PROJECT OVERVIEW

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room. Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. So, an automatic attendance system can solve all above problems.

There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique and RFID system. Although it is automatic and a step ahead of traditional method it fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking.

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

Facial Recognition Attendance is an attendance system that can be used to take attendance anywhere. This Facial Recognition Attendance uses technologies like Python, LBPH, and OpenCV. The system detects the face in the camera and then check it with the data saved about the particular person in the database, compares and marks attendance.

1.2 MODULE DESCRIPTION

MODULES

- Check Camera
- Capture Faces
- Train Images
- Recognize and Mark Attendance

Check Camera

To check if camera is connected to the system and if it is working. This module makes sure that the camera is well calibrated for our face recognition attendance system to work well.

Capture Faces

Detect the faces in the camera feed using Haar Cascades Algorithm and capture the images of the faces for the dataset. This uses the camera which is set up in the previous module.

Train Images

Train the model by using the captured images in Capture Faces module as the dataset. Here we have used the images of the faces captured in the previous module.

Recognize and Mark Attendance

The final module uses Haar Cascades to detect the images, then uses LBPH algorithm to recognize the student in the camera feed and then marks the attendance for the student.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

At present attendance marking involves manual attendance on paper sheet by professors and teachers. but it is very time-consuming process and chances of proxy is also one problem that arises in such type of attendance marking. also there are attendance marking system such as RFID, Biometrics etc. but these systems are currently not so much popular in schools and classrooms for students as they have their own advantages and disadvantages. 1.4 The Problems with Current System The problem with this approach in which manually taking and maintains the attendance records is that it is very inconvenient task. Traditionally, students attendances are taken manually by using attendance sheet given by the faculty members in class, which is a time consuming event Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not the ability to compute the attendance percentage becomes a major task as manual computation produces error and also wastes a lots of time this method could easily allows for impersonation and the attendance sheet could be stolen or lost.

DISADVANTAGES OF EXISTING SYSTEM

- Inconvenient and time-consuming
- Difficult to verify a student
- Manual computation is prone to error
- Easy to impersonate for proxy attendance
- Not a very hygiene system
- Low accuracy and slower

2.2 PROPOSED SYSTEM

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasis its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. Using Local Binary Patterns Histograms (LBPH) algorithm in Open CV the student's faces are trained and recognized. Python automates the tasks by providing for the execution of the programs in Computer Vision and GUI of the system along with managing the database of the student attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

ADVANTAGES OF PROPOSED SYSTEM

- Removes the risk of manual errors
- Fast, Hygiene, and Secure
- Proxy Attendance eliminated
- Easy to maintain the records
- Best for Virtual Classrooms
- Accurate detection and recognition

3. SYSTEM CONFIGURATION

3.1 HARDWARE SPECIFICATIONS

PROCESSOR	: Intel i3 or i5
RAM	: 4 GB or Higher
HDD	: 300 GB of available hard disk space
VIDEO CARD	: Nvidia GeForce GTX 940MX or Intel HD Graphics
KEY BOARD	: Standard Windows Keyboard

3.2 SOFTWARE SPECIFICATIONS

FRONT END	: PyCharm 2020.2.1 and Python 3.6
BACK END	: OpenCV and Microsoft Excel
OPERATING SYSTEM	: Ubuntu 18.04 LTS or Windows 10 2004
OPENCV	: Version 4.1.0 or Higher

4.SOFTWARE DESCRIPTION

4.1 FRONT END

PYTHON:

Python is one of those rare languages which can claim to be both *simple* and *powerful*. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

FEATURES OF PYTHON

- Free and Open Source
- Portable
- Interpreted
- Extensive Libraries

- Simple
- Easy to Learn

PYCHARM:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

FEATURES OF PYCHARM

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like matplotlib, numpy and scipy
- It competes mainly with a number of other Python-oriented IDEs, including Eclipse's PyDev, and the more broadly focused Komodo IDE.

HAAR Cascade Algorithm:

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

The algorithm has four stages:

1. Haar Feature Selection
2. Creating Integral Images
3. Adaboost training
- 4. Cascading Classifiers**

LBPH Algorithm:

Local Binary Patterns Histogram (LBPH) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

LBPH Algorithm works in 5 steps:

1. Parameters
2. Training the Algorithm
3. Applying the LBPH Operation
4. Extracting the Histogram
5. Performing the Face Recognition

4.2 BACK END

MICROSOFT EXCEL:

Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. It has been a very widely applied spreadsheet for these platforms, especially since version 5 in 1993, and it has replaced Lotus 1-2-3 as the industry standard for spreadsheets. Excel forms part of the Microsoft Office suite of software.

It also has a variety of interactive features allowing user interfaces that can completely hide the spreadsheet from the user, so the spreadsheet presents itself as a so-called application, or decision support system (DSS), via a custom-designed user interface, for example, a stock analyser, or in general, as a design tool that asks the user questions and provides answers and reports. In a more elaborate realization, an Excel application can automatically poll external databases and measuring instruments using an update schedule, analyse the results, make a Word report or PowerPoint slide show, and e-mail these presentations on a regular basis to a list of participants.

OPENCV:



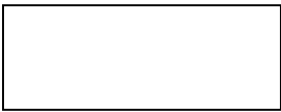
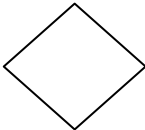
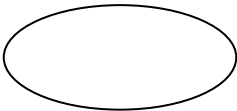
OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

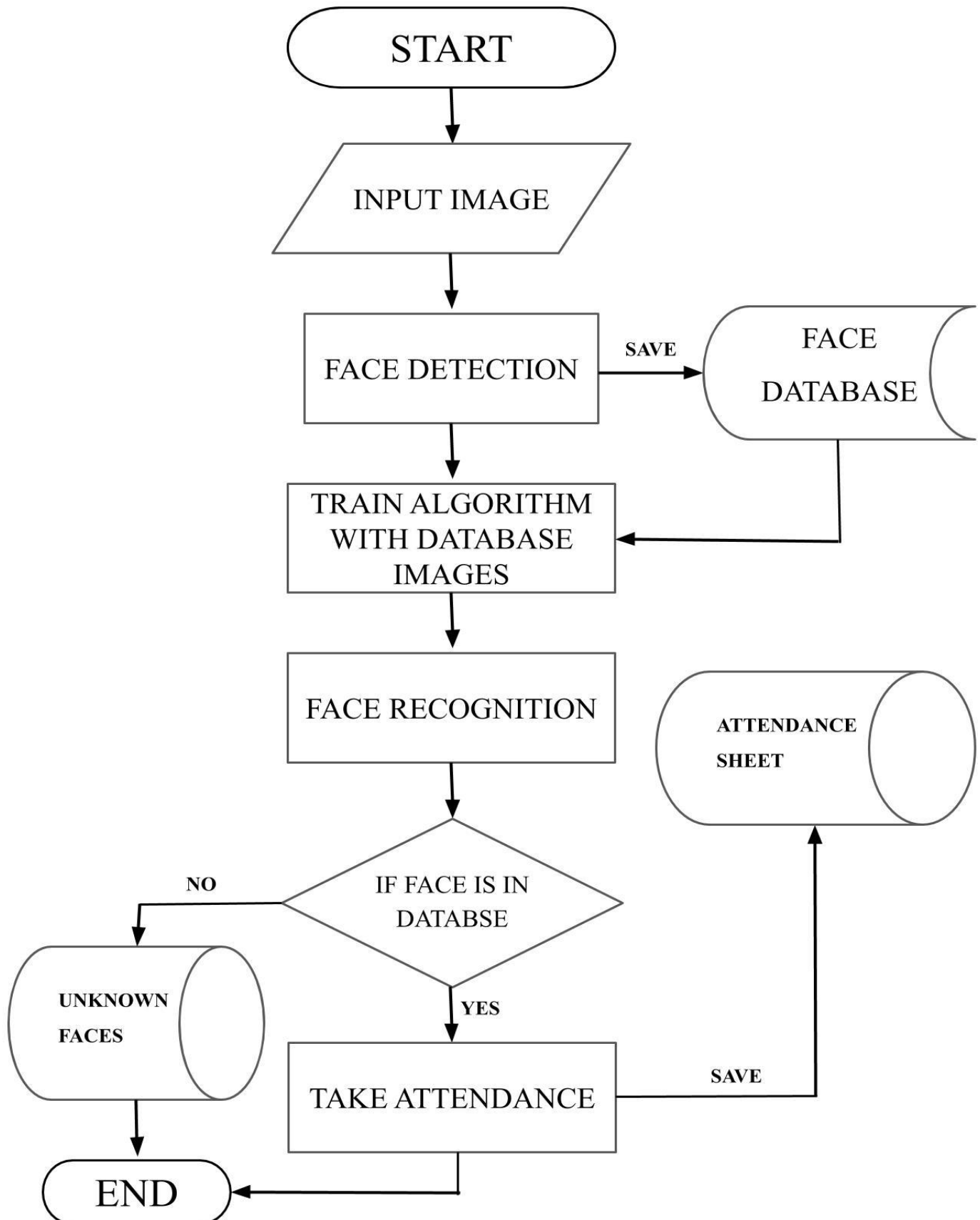
5. SYSTEM DESIGN

5.1 SYSTEM FLOW DIAGRAM

System Flow Diagram is basically a graphical and sequential representation of the major steps involved in a systematic process. A System Flow Diagram shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It gives a clear idea about the whole process.

SYMBOLS USED IN SYSTEM FLOW DIAGRAM

SYMBOL	DESCRIPTION
	An Arrows is a connector that shows relationships between the representative shapes.
	A Parallelogram represents input or output.
	A Rectangle represents a process.
	A Diamond indicates a decision.
	An Oval represents start or end.



5.2 INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer understandable format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. A large number of problems with a system can usually be tracked back to fault input design and method. Every moment of input design should be analysed and designed with utmost care.

The system takes input from the users, processes it and produces an output. Input design is link that ties the information system into the world of its users. The system should be user-friendly to gain appropriate information to the user. The decisions made during the input design are

- To provide cost effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is understood by the user.

System analysis decide the following input design details like, what data to input, what medium to use, how the data should be arranged or coded, data items and transactions needing validations to detect errors and at last the dialogue to guide user in providing input.

Input data of a system may not be necessarily is raw data captured in the system from scratch. These can also be the output of another system or subsystem. Here, the input that we give for training is an image. Also, when we run the program to recognize face, again we are giving the images of the faces as the input.

An image is a 2-Dimensional light intensity function $f(x, y) = r(x, y) \times i(x, y)$. Where, $r(x, y)$ is the reflectivity of the surface of the corresponding image point. $i(x, y)$ Represents the intensity of the incident light. A digital image $f(x, y)$ is discretized both in spatial co-ordinates by grids and in brightness by quantization. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies grey level value at that point.

5.3 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The output is designed in such a way that it is attractive, convenient and informative. As the outputs are the most important sources of information to the users, better design should improve the system's relationships with user and also will help in decision-making. Form design elaborates the way output is presented and the layout available for capturing information. All records from various Tables with same Database name are retrieved and consolidated, then displayed in a print Format.

The record here includes the name of the student, student ID, date, and time when the attendance has been taken. The data is saved as a Microsoft Excel file which can be later used to review the attendance taken by the system.

6. SYSTEM TESTING & IMPLEMENTATION

SYSTEM TESTING

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs.

Software testing should not be confused with debugging. Debugging is the process of analysing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information. Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing.

The test steps include

- Unit testing
- Validation testing
- Acceptance Testing
- Black Box Testing

Unit Testing

Unit testing focuses verification efforts on the smallest of the software design of the module. This is also known as module testing. The modules of the “Facial Recognition Attendance” are all tested separately to check if there are any errors in the same. This test was carried out during the programming stage to make sure everything works.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected and final series of software test begins. Validation testing can be defined in many ways, but a simple definition is the validation succeeds when the software functions in a manner that can be reasonable expected by the customer.

After the validation test has been conducted, one of the three possible conditions exists.

1. The function or performance characteristics conforms to specification and is accepted.
2. A deviation from specification is uncovered and a deficiency list is created.
3. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

Acceptance Testing

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle. This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

SYSTEM IMPLEMENTATION

The purpose of System Implementation can be summarized as follows:

In making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the performing organization (the transition). At a finer level of detail, deploying the system consists of executing all steps necessary to educate the consumers on the use of the new system, placing the newly developed system into production, confirming that all data required at the start of operations is available and accurate, and validating that business functions that interact with the system are functioning properly. Transitioning the system support responsibilities involves changing from a system development to a system support and maintenance modes of operation, with ownership of the new system from the Project Team to the Performing Organization.

System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning

Planning is the first task in the system implementation. Planning means deciding on the method and the time scale to be adopted. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments. The committee considers ideas, problems and complaints user department, it must also consider;

- The implication of system environment
- Self-selection and allocation form implementation tasks
- Consultation with unions and resources available
- Standby facilities and channels of communication

The purpose of Prepare for System Implementation to take all possible steps to ensure that the upcoming system deployment and transition occurs smoothly, efficiently, and flawlessly. In the

implementation of any new system, it is necessary to ensure that the Consumer community is best positioned to utilize the system once deployment efforts have been validated. Therefore, all necessary training activities must be scheduled and coordinated. As this training is often the first exposure to the system for many individuals, it should be conducted as professionally and competently as possible. A positive training experience is a great first step towards Customer acutance of the system.

During System Implementation it is essential that everyone involved should be absolutely synchronized with the deployment plan and with each other. Often the performance of deployment efforts impacts many of the Performing Organization's normal business operations. Examples of these impacts include:

The final process within the System Development Lifecycle is to transition ownership of the system support responsibilities to the Performing Organization. In order for there to be an efficient and effective transition, the Project Manager should make sure that all involved parties are aware of the transition plan, the timing of the various transition activities, and their role in its execution.

7. SCOPE FOR FUTURE ENHANCEMENT

- Can be implemented using Raspberry Pi in University, Schools, and Offices.
- Use of Deep Learning and Neural Networks to detect faces as it will give great accuracy than Haar Cascade Algorithm.
- But it requires lot of computing power and data to train along with a powerful device to run like Nvidia Jetson Nano.
- Nvidia provides custom Deep Learning libraries like Transfer Learning Toolkit to train and deploy in Jetson Nano.
- The training and deployment process in Nvidia's TLT is very complex and time consuming.

8. CONCLUSION

In this project we propose an automated attendance management system. This system, which is based on face detection and recognition algorithms, automatically detects the student when he enters the class room and marks the attendance by recognizing him. The system architecture and algorithms used in each stage are described in this project. Different real time scenarios are considered to evaluate the performance of various face recognition systems. This project also proposes the techniques to be used in order to handle the threats like spoofing. When compared to traditional attendance marking this system saves the time and also helps to monitor the students.

9. BIBLIOGRAPHY

Books:

1. OpenCV with Python by Example
2. Machine Learning for Open CV

Website Reference:

1. <https://opencv.org/>
2. <http://pyimagesearch.com/>
3. <https://medium.com/data-science-101/face-recognition-opencv-e841dc0006c6>
4. <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>

10. APPENDIX

A. SCREENSHOT

Welcome Menu:

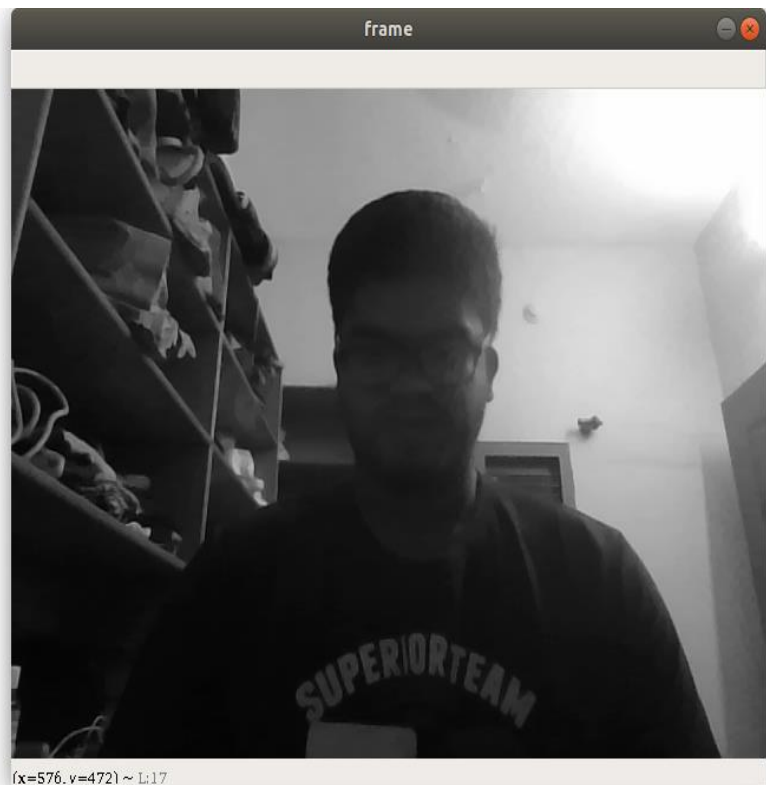
```
*****
***** Face Recognition Attendance *****
*****

***** WELCOME MENU *****
[1] Check Camera
[2] Capture Faces
[3] Train Images
[4] Recognize & Attendance
[5] Quit
Enter Choice: |
```

Check Camera:

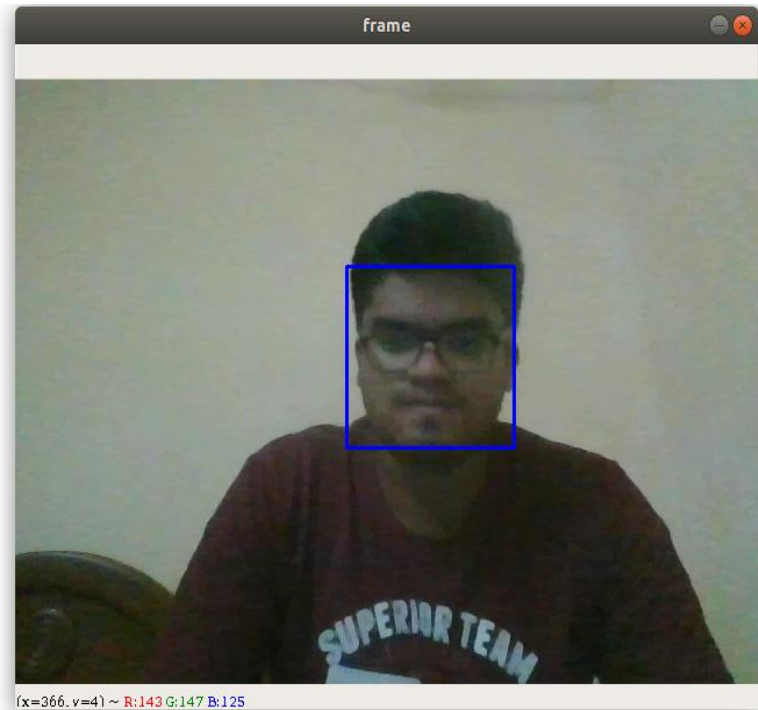
```
*****
***** Face Recognition Attendance *****
*****

***** WELCOME MENU *****
[1] Check Camera
[2] Capture Faces
[3] Train Images
[4] Recognize & Attendance
[5] Quit
Enter Choice: 1
```

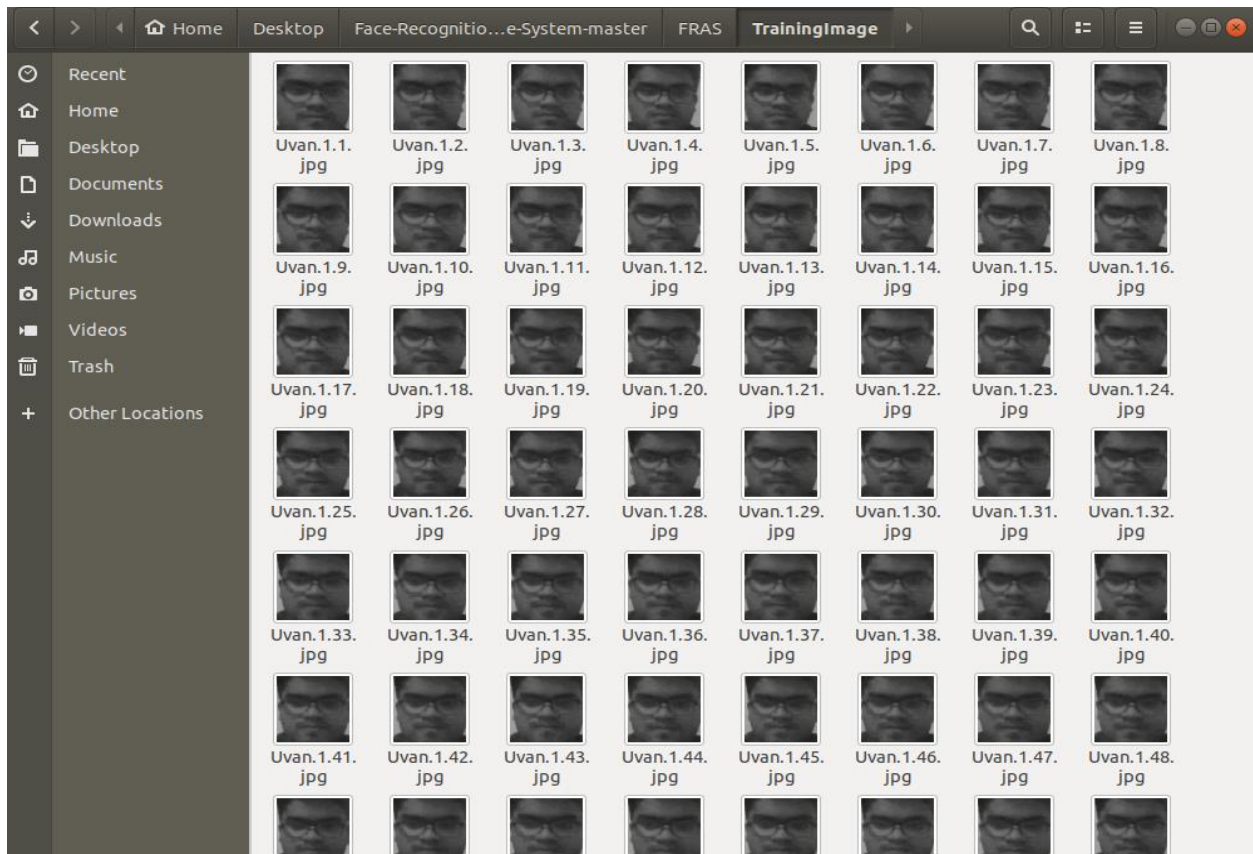


Capture Faces:

```
*****  
**** Face Recognition Attendance ****  
*****  
  
***** WELCOME MENU *****  
[1] Check Camera  
[2] Capture Faces  
[3] Train Images  
[4] Recognize & Attendance  
[5] Quit  
Enter Choice: 2  
Enter Your Id: 1  
Enter Your Name: Uvan
```



Face Database:



Recognize & Attendance:

```
*****  
**** Face Recognition Attendance ****  
*****  
  
***** WELCOME MENU *****  
[1] Check Camera  
[2] Capture Faces  
[3] Train Images  
[4] Recognize & Attendance  
[5] Quit  
Enter Choice: 4
```



Attendance Sheet:

Attendance_2020-09-14_23-58-49.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

<

B. SAMPLE CODING

Welcome Menu:

```
import os

import check_camera

import Capture_Image

import Train_Image

import Recognize


def title_bar():

    os.system('cls')

    print("\t*****")

    print("\t***** Face Recognition Attendance *****")

    print("\t*****")


def mainMenu():

    title_bar()

    print()

    print(10 * "*", "WELCOME MENU", 10 * "*")

    print("[1] Check Camera")

    print("[2] Capture Faces")

    print("[3] Train Images")

    print("[4] Recognize & Attendance")

    print("[5] Quit")
```



```
while True:

    try:

        choice = int(input("Enter Choice: "))

        if choice == 1:

            checkCamera()

            break

        elif choice == 2:

            CaptureFaces()

            break

        elif choice == 3:

            Trainimages()

            break

        elif choice == 4:

            RecognizeFaces()

            break

        elif choice == 5:

            print("Thank You")

            break

        else:

            print("Invalid Choice. Enter 1-4")

            mainMenu()

    except ValueError:

        print("Invalid Choice. Enter 1-4\n Try Again")
```

exit

def checkCamera():

 check_camera.camer()

 key = input("Enter any key to return main menu")

 mainMenu()

def CaptureFaces():

 Capture_Image.takeImages()

 key = input("Enter any key to return main menu")

 mainMenu()

def Trainimages():

 Train_Image.TrainImages()

 key = input("Enter any key to return main menu")

 mainMenu()

def RecognizeFaces():

 Recognize.recognize_attendance()

 key = input("Enter any key to return main menu")

 mainMenu()

mainMenu()

Check Camera:

```
import cv2

def camer():

    cap = cv2.VideoCapture(0)

    while(True):

        ret, frame = cap.read()

        gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

        cv2.imshow('frame', gray)

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

    cap.release()

    cv2.destroyAllWindows()
```

Capture Faces:

```
import csv

import cv2

import os

def is_number(s):

    try:
```

```

        float(s)

        return True

except ValueError:

    pass


try:

    import unicodedata

    unicodedata.numeric(s)

    return True

except (TypeError, ValueError):

    pass

return False


def takeImages():

    Id = input("Enter Your Id: ")

    name = input("Enter Your Name: ")


    if (is_number(Id) and name.isalpha()):

        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)

        sampleNum = sampleNum + 1

        cv2.imwrite("TrainingImage" + os.sep + name + "." + Id + '.' +
                    str(sampleNum) + ".jpg", gray[y:y + h, x:x + w])

        cv2.imshow('frame', img)

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break

    elif sampleNum > 60:

        break

    cam.release()

    cv2.destroyAllWindows()

    res = "Images Saved for ID : " + Id + " Name : " + name

    row = [Id, name]

    with open("StudentDetails" + os.sep + "StudentDetails.csv", 'a+') as csvFile:

        writer = csv.writer(csvFile)

        writer.writerow(row)

    csvFile.close()

else:

```

```
if (is_number(Id)):

    print("Enter Alphabetical Name")

if (name.isalpha()):

    print("Enter Numeric ID")
```

Train Images:

```
import os

import cv2

import numpy as np

from PIL import Image

def getImagesAndLabels(path):

    imagePath = [os.path.join(path, f) for f in os.listdir(path)]

    faces = []

    Ids = []

    for imagePath in imagePath:

        pilImage = Image.open(imagePath).convert('L')

        imageNp = np.array(pilImage, 'uint8')

        Id = int(os.path.split(imagePath)[-1].split(".")[1])

        faces.append(imageNp)

        Ids.append(Id)

    return faces, Ids
```

```

def TrainImages():

    recognizer = cv2.face_LBPHFaceRecognizer.create()

    harcascadePath = "haarcascade_frontalface_default.xml"

    detector = cv2.CascadeClassifier(harcascadePath)

    faces, Id = getImagesAndLabels("TrainingImage")

    recognizer.train(faces, np.array(Id))

    recognizer.save("TrainingImageLabel"+os.sep+"Trainer.yml")

    print("Images Trained")

```

Recognize & Attendance:

```

import datetime

import os

import time

import cv2

import pandas as pd

def recognize_attendance():

    recognizer = cv2.face.LBPHFaceRecognizer_create()

    recognizer.read("TrainingImageLabel"+os.sep+"Trainer.yml")

    harcascadePath = "haarcascade_frontalface_default.xml"

    faceCascade = cv2.CascadeClassifier(harcascadePath)

    df = pd.read_csv("StudentDetails"+os.sep+"StudentDetails.csv")

    cam = cv2.VideoCapture(0)

```

```

font = cv2.FONT_HERSHEY_SIMPLEX

col_names = ['Id', 'Name', 'Date', 'Time']

attendance = pd.DataFrame(columns=col_names)


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)

        Id, conf = recognizer.predict(gray[y:y+h, x:x+w])

        if(conf < 50):

            ts = time.time()

            date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')

            timeStamp = datetime.datetime.fromtimestamp(

                ts).strftime('%H:%M:%S')

            aa = df.loc[df['Id'] == Id]['Name'].values

            tt = str(Id)+"-"+aa

            attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]

        else:

            Id = 'Unknown'

```



```

        tt = str(Id)

    if(conf > 75):

        noOfFile = len(os.listdir("ImagesUnknown"))+1

        cv2.imwrite("ImagesUnknown"+os.sep+"Image"+str(noOfFile) +

                    ".jpg", im[y:y+h, x:x+w])

        cv2.putText(im, str(tt), (x, y+h), font, 1, (255, 255, 255), 2)

    attendance = attendance.drop_duplicates(subset=['Id'], keep='first')

    cv2.imshow('im', im)

    if (cv2.waitKey(1) == ord('q')):

        break

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')

timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

Hour, Minute, Second = timeStamp.split(":")

fileName      =      "Attendance"+os.sep+"Attendance_"+date+"_"+Hour+"-"+Minute+"-
"+Second+".csv"

attendance.to_csv(fileName, index=False)

cam.release()

cv2.destroyAllWindows()

print("Attendance Successfull")

```