SMART SECURITY WITH ADVANCE FACE RECOGNITION

A project Report Submitted in partial fulfilment of the requirements for the degree of B.Sc. in

Information of Science and Telecommunication

Submitted by:

Dibyajyoti Mohanty (19DIS032)

Under The Guidance

Mr. Umesh Prasad Rout

Ravenshaw University

2019-2022

DEPARTMENT OF INFORMATION SCIENCE AND TELECOMMUNICATYION

RAVENSHWA UNIVERSITY

CUTTACK, ODISHA

DECLARATION

I hereby declare that the project entitled **Smart Security Cam with Advance Face recognition** submitted to the Department of IST, Ravenshaw University, Cuttack is recorded of an original work done by us Under the guidance of Mr. Umesh Prasad Rout, Ravenshaw University. And this project is submitted in the partial fulfilment for the award of 'B.Sc. in (IST)' form Ravenshaw University' Cuttack. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

This is to certify that the above declaration is true.

Dibyajyoti Mohanty

(19DIS032)

Department of Information Science and Telecommunication, Ravenshaw University

Cuttack-753003, Odisha, India

CERTIFICATE

This is to certify that the dissertation work entitled **Smart Security Cam with Advance Face recognition** is the work done by

Dibyajyoti Mohanty **(19DIS032),**

Soumesh Brahma **(19DIS012),**

Padmini Mardi **(19DIS055),**

Preeti Priyadarsini Sahoo **(19DIS020),**

Ashutosh Das **(19DIS006)**

submitted in partial fulfilment for the award of 'Bachelor of Science' form Ravenshaw University' Cuttack.

_____ _____ _____

(Course Coordinator) **(External Examiner)** **(Internal Examiner)**

Dept. of Information Science and Telecommunication

Ravenshaw University

Cuttack, Odisha, India

CERTIFICATE

It is Certified that **Dibyajyoti Mohanty (19DIS032), Soumesh Brahma (19DIS012), Padmini Mardi (19DIS055), Preeti Priyadarsini Sahoo (19DIS020) and Ashutosh Das (19DIS006)"** has carried out the project work presented in this report entitled **"SMART SECURITY WITH ADVANCE FACE RECOGNITION"** for the award of **B.Sc. (IST)** from **Ravenshaw University**, **Cuttack** under my supervision. The report embodies results of original work, and studies are carried out by the students and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

I wish them all success in life...

Signature of the Project Guide

Mr. UMESH PRASAD ROUT

Ravenshaw University

Cuttack

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude to those whose contribution in stimulating suggestions and encouragement helped me to coordinate my project.

I would like to express my thankfulness to my project guide, Mr. UMESH PRASAD ROUT, for his constant motivation and valuable help through the project work.

Last but not the least I would like to mention here that I am greatly indebted to each and everybody who acknowledgement has been associated with my project at any stage but whose name does not find a place in this.

PREFACE

The main objective of any Information science student is to get as much of practical knowledge as possible. Being an able to have a practical knowledge by developing a project is a lifetime experience. As practical knowledge is as important as theoretical knowledge, we are thanked of having a project. Through the development of the project, we had a great experience of various strategies that can be applied in development of project. This project is the stepping step for our carrier. We are pleased to present this project. Proper care has been taken while organizing the project so that it is so important also various software engineering concepts have been implemented.

ABSTRACT

"SMART SECURITY CAM WITH ADVANCE FACE RECOGNITION" is an application which can connect to any camera and act as an advance smart security or CCTV. If a burglar tries to entering the house it can notify to the user. And it can work 24*7. Not only burglar, if there is some kind of accident (pipe burst, fire breakout) happening at the absent of the owner, it will notify the user ASAP. This system also acts as a smart door lock. It always unlocks the door for known person, but for unknow person it will shut down the door and send the pic of that person to the owner.

CONTENTS

1.
INTRODUCTION

1.INTRODUCTION

Whether it is past, present or future, there are 3 basic requirements for humans that never change. These are Food, Cloths and house.

During 2020, the home became not just the hub of personal life but of work life as well. As our daily routines were disrupted, our homes turned into full-time schools, cinemas, playgrounds, and workplaces.

But now a days there are some people who ruined these happy moments. We call those people burglar.

Not only burglar but also if any kind of mis happening happen it will inform to the user right away.

1-a. Aim of the Project

"SMART SECURITY CAM WITH ADVANCE FACE RECOGNITION" is application which is used for protecting our home form malicious people as well as inform to the user if there is any kind of misfortune occurs.

Our aim is to protects your belongings when you are not at your home.

First and foremost, a home security system aims to protect your property and those inside it from burglary, home intrusion, fire, and other environmental disasters such as burst pipes.

1-b.

Background of the Project

The smart home security system is a great tool for protecting your property since it provides total control of everything inside and outside of your house.

The market today is undergoing a full-scale transformation, or as we like to call it – a digital revolution. In 2017, there was a 31% increase in connected devices as compared to 2016. It has been estimated that the total number of connected devices has reached 20 billion this year.

Technology is evolving day-by-day, alongside us. And the Internet of Things (IoT) is one of the newest additions to the world of technology. With the IoT, all eligible electronic devices can be interconnected and share data. Home automation, a growing market, is being achieved with the help of IoT devices. These smart devices often work with voice commands and follow the rules of automation.

One of the most crucial aspects of smart home automation with IoT is security. Traditional ways to secure your home are now outdated. At Legrand, we feel you can only live the advantage by upgrading to smarter, durable and safer replacements.

1-c. Why this Application

Home security is an essential part of the Internet of Things. Artificial intelligence and IoT is excellent for smart security. With CCTV cameras, doorbells, smart lights and many other alert systems, closely monitoring your home has never been easier!

Home security is an essential part of our lives. Break-ins are extremely common these days. No should leave their house unprotected. Therefore, taking safety measures to up one's home security system is crucial.

Home security can be improved with the Internet of Things. There are a ton of different IoT devices helpful in monitoring suspicious activity.

There is a high chance that your house could be infiltrated by a thief when you are away. Most thefts happen when homeowners are elsewhere.

Here comes our application to play which has advance face recognition technique which allow only those people who previously present in the database otherwise only allowed by user.

As homeowners seek to safeguard their properties and loved ones, the need for increased technology, convenience, and efficiency for everyday household functions remains top of mind. Whether you're a new homeowner implementing a security system or upgrading your current solution, consider utilizing smart technologies

2.

System Analysis

During 2020, the home became not just the hub of personal life but of work life as well. As our daily routines were disrupted, our homes turned into full-time schools, cinemas, playgrounds, and workplaces.

First and foremost, a home security system aims to protect your property and those inside it from burglary, home intrusion, fire, and other environmental disasters such as burst pipes.

The main purpose of a home security system is to keep your property and the people inside it safer. Harm may come in the form of a burglary, home invasion, fire, flood, or other environmental disaster.

SMART SECURITY CAM WITH ADVANCE FACE RECOGNITION has two operations to protect our loving persons from any kind of danger. This application can connect to any camera by having it number only and that camera can perform its all operation.

First Operation:

By the help of face recognition system, it can recognize the known faces and allow to the house by unlocked the door for 5 second.

But if their faces are not recognized by the camera, then it will take pic of that person and send it to the user phone.

Now if user allow that person, then the door is unlocked for 5 second and if the user does not allow then the door will be remaining locked.

If by some reason the user is not present at the time of the sending the person pic, then it will show for 10 second on the smart phone of the user phone. If there is not respond coming then door will still be locked.

By the help of this process, we can also get the attendance of the employees with time stamp.

Second Operation:

This operation generally used at the time of the night, when every one asleep. Or when there is no one present at the house or office.

At that time if there is any disturbance occur like burglar attack, fire burst out, pipe burst etc. the camera immediately sending a massage to the user smart phone with a warning sound.

There is a thing I am pointing out that if there are any small disturbances like wind flow does not do any effect.

2-a. Hardware Requirement

PROCESSOR	Intel Core i3 7th generation Processor, AMD Rizan 3
OPERATING SYSTEM	Windows 11, Windows 10
RAM	4 GB
ROM	1 TB
MEMORY	256 GB SSD, 1 TB HDD

2-b. Software Requirement

PROGRAMMING LANGUAGE	PYTHON
IDEs	PYCHARM, JUPYTER NOTEBOOK
LIBRARIES	OpenCV, Face-recognition, KivyMD, Numpy, winSound
DEPENDECIES	Cmake, Visual Studio (C++), Dlib

2-c.

Software Requirement Specification (SRS)

SMART SECURITY CAM WITH ADVANCE FACE RECOGNITION application fully created in Python Programming Language.

We use python programming language because:

Python is designed to stress on code readability by utilizing substantial white space and simplicity as it allows programmers to write models and conceptions in less amounts of code lines compared to other languages like C++ or Java.

This makes python a very popular programming language used for desktop stand-alone applications or online/ web applications as well as small or large scales development projects.

Python is Easy to Learn and Use

Python is incredibly easy to learn and use for beginners and newcomers in the industry. The language is the most accessible among all the programming languages available because it has simplified syntax that is not complicated at all and gives more emphasis on natural language. Due to its ease of learning and usage, Python codes can easily be written and executed much faster than other programming languages. One of the main reasons why Python's popularity has exponentially grown is due to its simplicity in syntax so that it could be easy to read and developed by amateur professionals as well.

Python is Handy for Web Development Purposes

According to expert professional web developers, Python is among the most convenient programming languages among the other alternatives. Due to the availability of its vast range of applications with in-built solutions to standard web development tasks, the speed of a single project increases by many times.

The Language is Extensively used in Data Science

Whichever path you choose in the tech world, data will always be a part of it. Currently, Python is extensively used in data science. The amounts of data generated by data analytics tools are increasing every day, hence, experts need to learn programming languages like Python to handle advanced technologies for data analytics. Data professionals also need to stay updated with the latest tech developments in the industry to leverage the most advanced technologies.

Has Multiple Libraries and Frameworks

Python is quite popular due to its hundreds of different libraries and frameworks that can be used by developers. These libraries and frameworks are really useful in saving time which in turn makes the language even more useful. Some of the most popular libraries available in Python are NumPy and SciPy, Django, and others that are used for different purposes.

Python can be used in ML tool

Python is used in big data and machine learning research purposes to enhance development in those fields. Python is extremely useful in the AI domain and is also used in robotics and other tech advancements, besides data science.

Has a Highly Supportive Community

Since is not only one of the most popular programming languages, but is also one of the oldest. Hence, it got the time to gather a growing, supportive community of programmers, developers, and coders. Because of this support, Python learners can easily develop industry-required skills and can be subjected to the right guidance.

Flexibility and Reliability

The Python language is flexible enough to provide developers with ample time to try new experiments. Python experts will not just stop at the usual things, instead, they will try to build new processes, technologies, or applications. It provides the developers with the freedom and flexibility that they need to just learn one language and utilize their best potential and skills.

Python Automates Tasks

Python language can help automate several industrial tasks that require too many tools and modules. It is incredible to know that one can reach an advanced level of automation easily by just using necessary Python codes. The language is considered a performance booster in automation software testing. The developers only need to write a few lines of code for automation tools.

With the process of development of web applications evolving radically in the last few years, here is what makes Python a leading contender or choice among software developers. It can build just about anything.

As a general-purpose language, Python is the single language that you may ever need for developing just about anything.

It can be used for the programming of the front end (client side) with which users interact and back end (server side) database of your website. It can be used for numerical and data analysis for scientific study and research. It can be used to develop artificial intelligence. It can be used to develop both online and offline applications from productivity tools, games and other type of app you can think off.

In short, Python is the jack-of-all-trades of programming language. And mastering it can potentially allow a software developer to be an expert in all types of programming. It is fun and easy to use.

That's why we use choose Python programming language to create our application.

We Use PyCharm and Jupyter note book IDEs to create, compile, test, run and execute our code.

PyCharm:

PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring and rich navigation capabilities.

Intelligent Code Editor

PyCharm's smart code editor provides first-class support for Python, JavaScript, CoffeeScript, TypeScript, CSS, popular template languages and more. Take advantage of language-aware code completion, error detection, and on-the-fly code fixes!

Smart Code Navigation

Use smart search to jump to any class, file or symbol, or even any IDE action or tool window. It only takes one click to switch to the declaration, super method, test, usages, implementation, and more.

Fast and Safe Refactorings

Refactor your code the intelligent way, with safe Rename and Delete, Extract Method, Introduce Variable, Inline Variable or Method, and other refactorings. Language and framework-specific refactorings help you perform project-wide changes.

Built-in Developer Tools

PyCharm's huge collection of tools out of the box includes an integrated debugger and test runner; Python profiler; a built-in terminal; integration with major VCS and built-in database tools; remote development capabilities with remote interpreters; an integrated ssh terminal; and integration with Docker and Vagrant.

Web Development

In addition to Python, PyCharm provides first-class support for various Python web development frameworks, specific template languages, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js, and more.

Python Web frameworks

PyCharm offers great framework-specific support for modern web development frameworks such as Django, Flask, Google App Engine, Pyramid, and web2py, including Django templates debugger, manage.py and appcfg.py tools, special autocompletion and navigation, just to name a few.

Scientific Tools

PyCharm integrates with IPython Notebook, has an interactive Python console, and supports Anaconda as well as multiple scientific packages including Matplotlib and NumPy.

Use PyCharm on Windows, macOS and Linux with a single license key. Enjoy a fine-tuned workspace with customizable colour schemes and key-bindings, with VIM emulation available.

Plugins

More than 10 years of IntelliJ platform development gives PyCharm 50+ IDE plugins of different nature, including support for additional VCS, integrations with different tools and frameworks, and editor enhancements such as Vim emulation.

Cross-platform IDE

PyCharm works on Windows, macOS or Linux. You can install and run PyCharm on as many machines as you have, and use the same environment and functionality across all your machines.

Jupyter note book:

A Jupyter Notebook provides you with an easy-to-use, interactive data science environment that doesn't only work as an integrated development environment (IDE), but also as a presentation or educational tool.

Although Jupyter has been developed for data science applications, which are written in languages like Python, R and Julia, the platform is now used in all kinds of ways for projects. Apart from that, by removing the barriers for data scientists, Jupyter made documentation, data visualisations, and caching a lot easier, especially for hardcore non-technical folks.

A data science enthusiast said, "Jupyter Notebook should be an integral part of any Python data scientist's toolbox. It's great for prototyping and sharing notebooks with visualisations."

Exploratory Data Analysis: Jupyter allows users to view the results of the code in-line without the dependency of other parts of the code. In the notebook, every cell of the code can be potentially checked at any time to draw an output. Because of this, unlike other standard IDEs like PyCharm, VSCode, Jupyter helps in in-line printing of the output, which becomes extremely useful for exploratory data analysis (EDA) process.

Easy Caching In Built-In Cell: Maintaining the state of execution of each cell is difficult, but with Jupyter, this work is done automatically. Jupyter caches the results of every cell that is running — whether it is a code that is training an ML model or a code that is downloading gigabytes of data from a remote server.

Language Independent: Because of its representation in JSON format, Jupyter Notebook is platform-independent as well as language-independent. Another reason is that Jupyter can be processed by any several languages, and can be converted to any file formats such as Markdown, HTML, PDF, and others.

Data Visualisation: As a component, the shared notebook Jupyter supports visualisations and includes rendering some of the data sets like graphics and charts, which are generated from codes with the help of modules like Matplotlib, Plotly, or Bokeh. Jupyter lets the users narrate visualisations, alongside share the code and data sets, enabling others for interactive changes.

Live Interactions with Code: Jupyter Notebook uses "ipywidgets" packages, which provide standard user interfaces for exploring code and data interactivity. And therefore, the code can be edited by users and can also be sent for a re-run, making Jupyter's code non-static. It allows users to control input sources for code and provide feedback directly on the browser.

Documenting code samples: Jupyter makes it easy for users to explain their codes line-by-line with feedback attached all along the way. Even better, with Jupyter, users can add interactivity along with explanations, while the code is fully functional.

2-d. Tools used with descriptions

For **SMART SECURITY CAM WITH ADVANCE FACE RECOGNITION** first we need to access the camera.

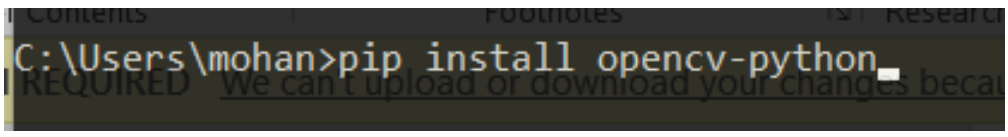
For access the camera we used a library called OpenCV.

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.

For installing OpenCV library first go to the Command Prompt and write.



For using the library, we have to import it first by writing:

```
import cv2
```

Automatic Allocation of the Output Data

OpenCV deallocates the memory automatically, as well as automatically allocates the memory for output function parameters most of the time.

```
import cv2

cap = cv2.VideoCapture(0)

while 1:
    _, fr = cap.read()
    fr = cv2.resize(fr, (600, 400))
    cv2.imshow('dibya', fr)

    if cv2.waitKey(1) == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

So, if a function has one or more input arrays (cv::Mat instances) and some output arrays, the output arrays are automatically allocated or reallocated.

The size and type of the output arrays are determined from the size and type of input arrays.

If needed, the functions take extra parameters that help to figure out the output array properties.

The imgcodecs module handles reading and writing image files.

When you operate on an input image and create an output image, you can save it as a jpg or a png file with a simple command.

```
#To save the image
cv2.imwrite('dibya.png', original_frame)
```

Image processing operations

When you write a Computer Vision algorithm, there are a lot of basic image processing operations that you will use over and over again.

Building GUI

OpenCV provides a module called high-up that handles all the high-level user interface operations.

This module has functions that can be used to create windows to display images and/or video.

There is also a waiting function that will wait until you hit a key on your keyboard before it goes to the next step. Video analysis

Video analysis includes tasks such as analysing the motion between successive frames in a video, tracking different objects in a video, creating models for video surveillance, and so on.

Object detection

Object detection refers to detecting the location of an object in a given image.

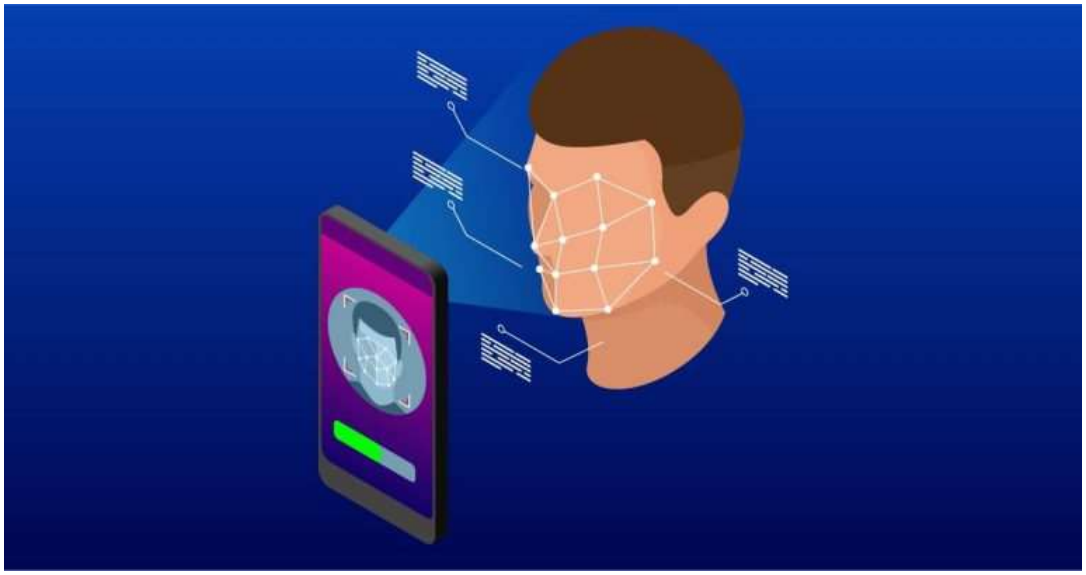
This process is not concerned with the type of object, it will just tell you the location of a given image.

Now the next process is to recognize the face. For that purpose, we have to install another library called Face-recognition.

Face Recognition:

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time. Law enforcement may also use mobile devices to identify people during police stops.

But face recognition data can be prone to error, which can implicate people for crimes they haven't committed. Facial recognition software is particularly bad at recognizing African Americans and other ethnic minorities, women, and young people, often misidentifying or failing to identify them, disparately impacting certain groups.

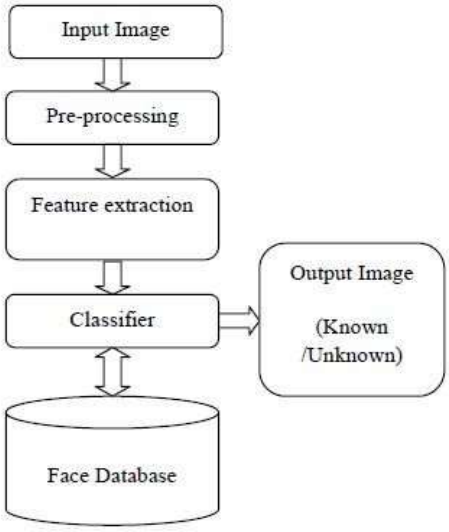


HOW FACE RECOGNATION WOKS:

You might be good at recognizing faces. You probably find it a cinch to identify the face of a family member, friend, or acquaintance. You're familiar with their facial features — their eyes, nose, mouth — and how they come together.

That's how a facial recognition system works, but on a grand, algorithmic scale. Where you see a face, recognition technology sees data. That data can be stored and accessed. For instance, half of all American adults have their images stored in one or more facial-recognition databases that law enforcement agencies can search, according to a Georgetown University study.

So how does facial recognition work? Technologies vary, but here are the basic steps:

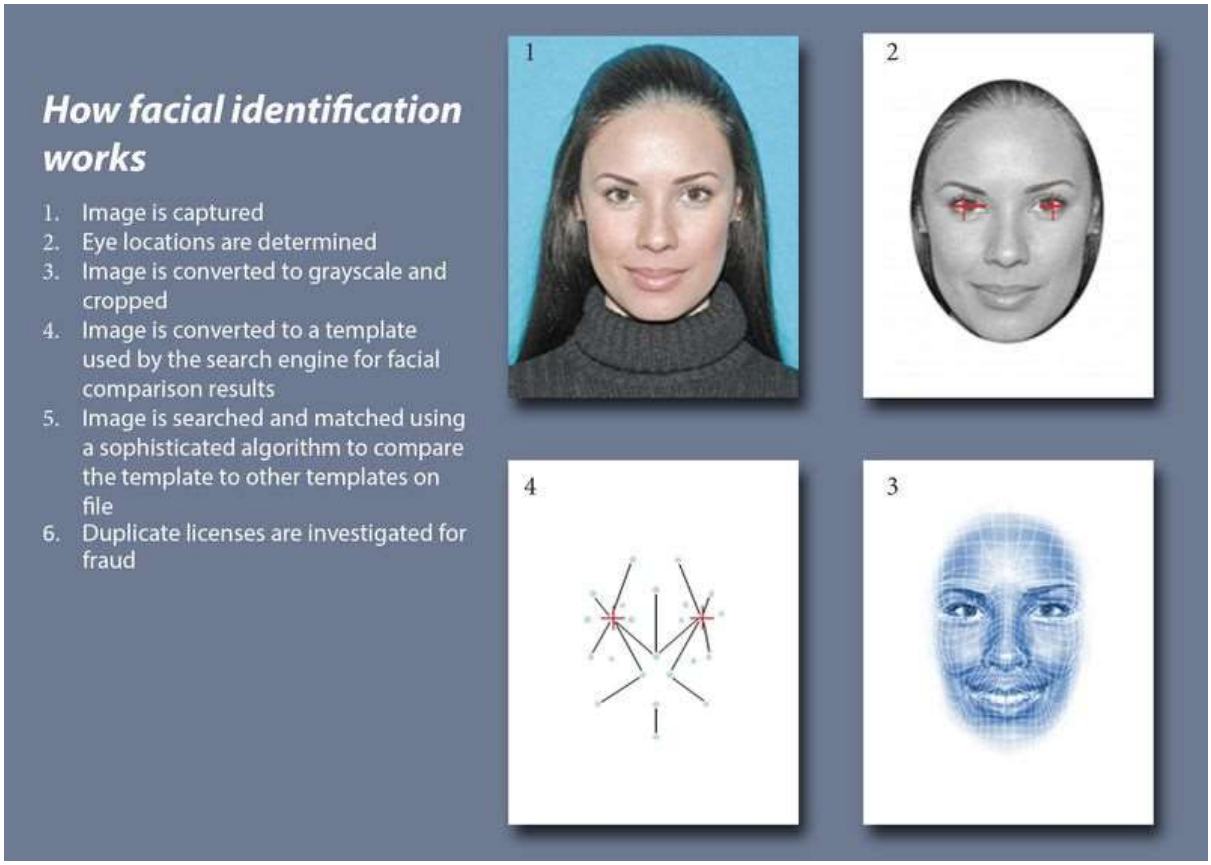


Step 1. A picture of your face is captured from a photo or video. Your face might appear alone or in a crowd. Your image may show you looking straight ahead or nearly in profile.

Step 2. Facial recognition software reads the geometry of your face. Key factors include the distance between your eyes and the distance from forehead to chin. The software identifies facial landmarks — one system identifies 68 of them — that are key to distinguishing your face. The result: your facial signature.

Step 3. Your facial signature — a mathematical formula — is compared to a database of known faces. And consider this: At least 117 million Americans have images of their faces in one or more police databases. According to a May 2018 report, the FBI has had access to 412 million facial images for searches.

Step 4. A determination is made. Your faceprint may match that of an image in a facial recognition system database.



Before activating this library, we have to install Cmake and dlib.

Let's break down the actual code

```
# Get user supplied values

imagePath = sys.argv[1]

cascPath = sys.argv[2]
```

You first pass in the image and cascade names as command-line arguments. We'll use the ABBA image as well as the default cascade for detecting faces provided by OpenCV.

```
# Create the haar cascade

faceCascade = cv2.CascadeClassifier(cascPath)
```

Now we create the cascade and initialize it with our face cascade. This loads the face cascade into memory so it's ready for use. Remember, the cascade is just an XML file that contains the data to detect faces.

```
# Read the image

image = cv2.imread(imagePath)

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

Here we read the image and convert it to grayscale. Many operations in OpenCV are done in grayscale.

```
# Detect faces in the image

faces = faceCascade.detectMultiScale(

    gray,

    scaleFactor=1.1,

    minNeighbors=5,
```

```
    minSize=(30, 30),
    flags = cv2.cv.CV_HAAR_SCALE_IMAGE
)
```

This function detects the actual face and is the key part of our code, so let's go over the options:

- The detectMultiScale function is a general function that detects objects. Since we are calling it on the face cascade, that's what it detects.
- The first option is the grayscale image.
- The second is the scaleFactor. Since some faces may be closer to the camera, they would appear bigger than the faces in the back. The scale factor compensates for this.
- The detection algorithm uses a moving window to detect objects. minNeighbors defines how many objects are detected near the current one before it declares the face found. minSize, meanwhile, gives the size of each window.

This function returns 4 values: the x and y location of the rectangle, and the rectangle's width and height (w, h).

We use these values to draw a rectangle using the built-in rectangle () function.

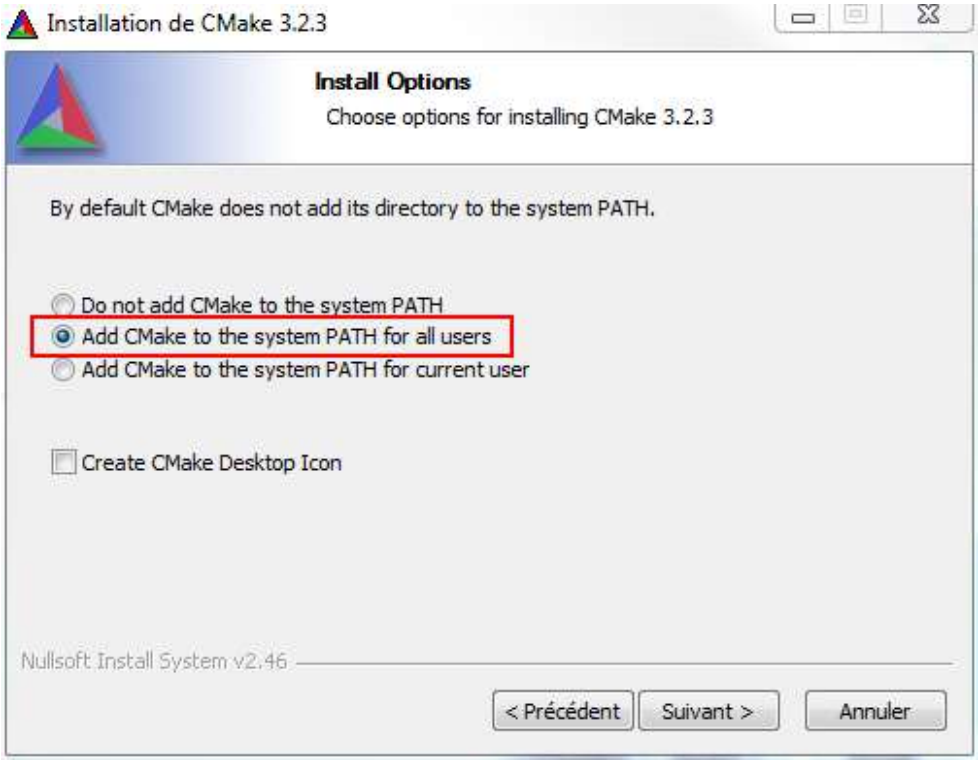
Cmake:

CMake is an open-source, cross-platform tool that uses compiler and platform independent configuration files to generate native build tool

files specific to your compiler and platform. The CMake Tools extension integrates Visual Studio Code and CMake to make it easy to configure, build, and debug your C++ project

Cmake is a prerequisite library so that face recognition library installation doesn't give us an error. After the installation is completed, let's import them into our code editor. Some of these libraries are included in Python that's why we can import them without installing them.

- To download Cmake go to the www.cmake.org
- Click on download button
- Then choose the right variant for your system.
- After downloading installing the Cmake

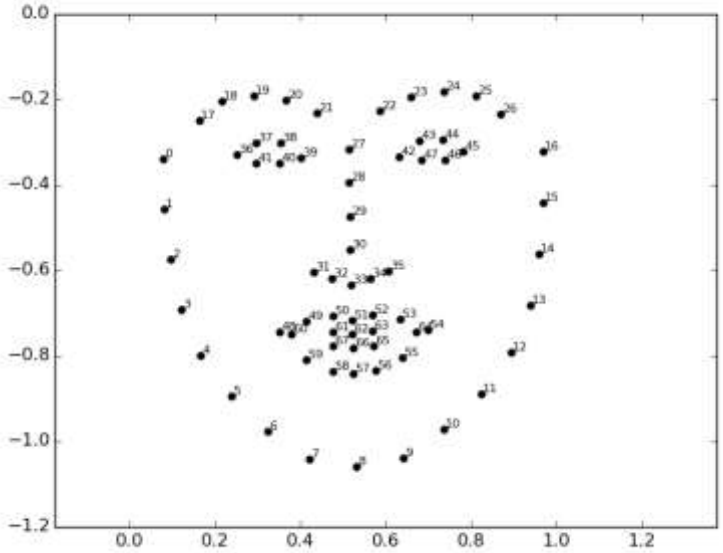


- Select “add Cmake to the system path for all users.”
- Go to the path of its installation and copy the path and past it to the environment path option.
- Then open CMD and write `pip install` cmake

dlib:

It's a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face like image below. These points are identified from the pre-trained model where the iBUG300-W dataset was used.

Cmake is a prerequisite library so that face recognition library installation doesn't give us an error. After the installation is completed, let's import them into our code editor. Some of these libraries are included in Python that's why we can import them without installing them.



For installing dlib we have to open CMD and write

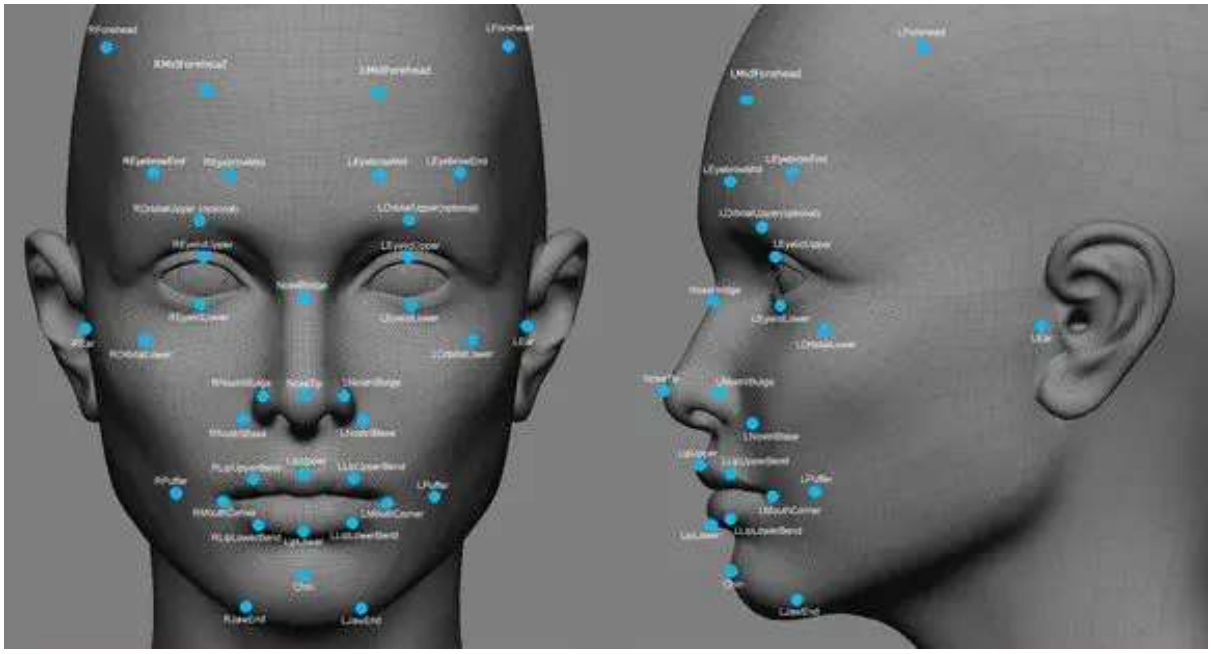
```
pip install dlib
```

Dlib provides two methods to perform face detection:

1. HOG + Linear SVM: dlib.get_frontal_face_detector ()

2. MMOD CNN: dlib.cnn_face_detection_model_v1(modelPath)

The HOG + Linear SVM face detector will be faster than the MMOD CNN face detector but will also be less accurate as HOG + Linear SVM does not tolerate changes in the viewing angle rotation.

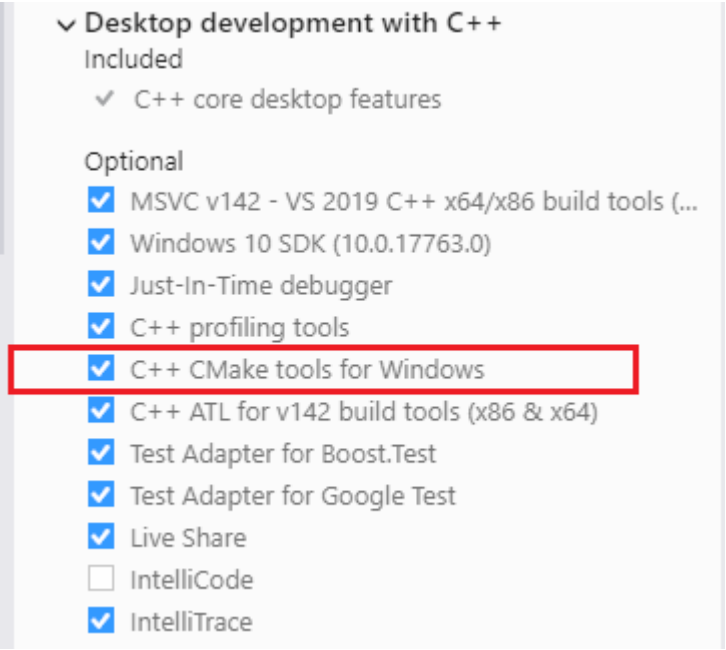


For more robust face detection, use dlib's MMOD CNN face detector. This model requires significantly more computation (and is thus slower) but is much more accurate and robust to changes in face rotation and viewing angle.

VS Code:

We have to install VS code also for active our face recognition

The Visual studio can be downloaded in the link https://visualstudio.microsoft.com/visual-cpp-build-tools/. After finishing the installation, you need to install additional packages for C, C++ programming, which is Packages CMake tools for Windows



Now for run the face recognition we have to import this:

```
import cv2
import face_recognition
```

We can install them in one line using PIP library manager: pip install cmake face_recognition numpy opencv-python

KivyMD:

Kivy is a Python library that facilitates the creation of cross-platform applications that can run on Windows, Linux, Android, OSX, iOS, and Raspberry pi too

KivyMD is an extension of the Kivy framework. KivyMD is a collection of Material Design widgets for use with Kivy, a GUI framework for making mobile applications. It is similar to the Kivy framework but provides a more attractive GUI.

KivyMD is a collection of Material Design compliant widgets for use with Kivy, a framework for cross-platform, touch-enabled graphical applications.

For Installing KivyMD open CMD and write the following code.

```
pip install kivymd==0.104.2
```

for using KivyMD we have to import it first by writing:

```
from kivymd.app import MDApp
```

This is how we used to write KivyMD code

```
from kivymd.app import MDApp
from kivymd.uix.screen import Screen
from kivymd.uix.label import MDLabel, MDIcon
from kivymd.uix.button import
MDRectangleFlatButton

class DemoApp(MDApp):
    def build(self):
        self.theme_cls.theme_style = "Dark"
        screen = Screen()
```

```
        label = MDLabel(text="hello world",
halign = 'center', theme_text_color="Custom",
text_color = (236/255.0, 98/255.0, 81/255.0,
1),
                        font_style="H1")
        i_label = MDIcon(icon='language-
python', halign='center')
        btn = MDRectangleFlatButton(text="hello
world", pos_hint={'center_x': 0.1,
'center_y':0.9})
        screen.add_widget(btn)
        return screen
DemoApp().run()
```

And This is the output:

To create this project, we have to use several tools.

- ❖ Programming Language:

- Python 3.9.5
  - 

❖ IDEs:

- Pycharm 2021.3
  - 

- Jupyter Notebook v6.4.11
  - 

❖ Libraries:

- OpenCV 4.6.0.66

- Face-recognition 1.3.0

- Cmake 3.22.1

- **Dlib 19.22**

- KivyMD 0.104.2

3.

Design and Development

3. Design and Development

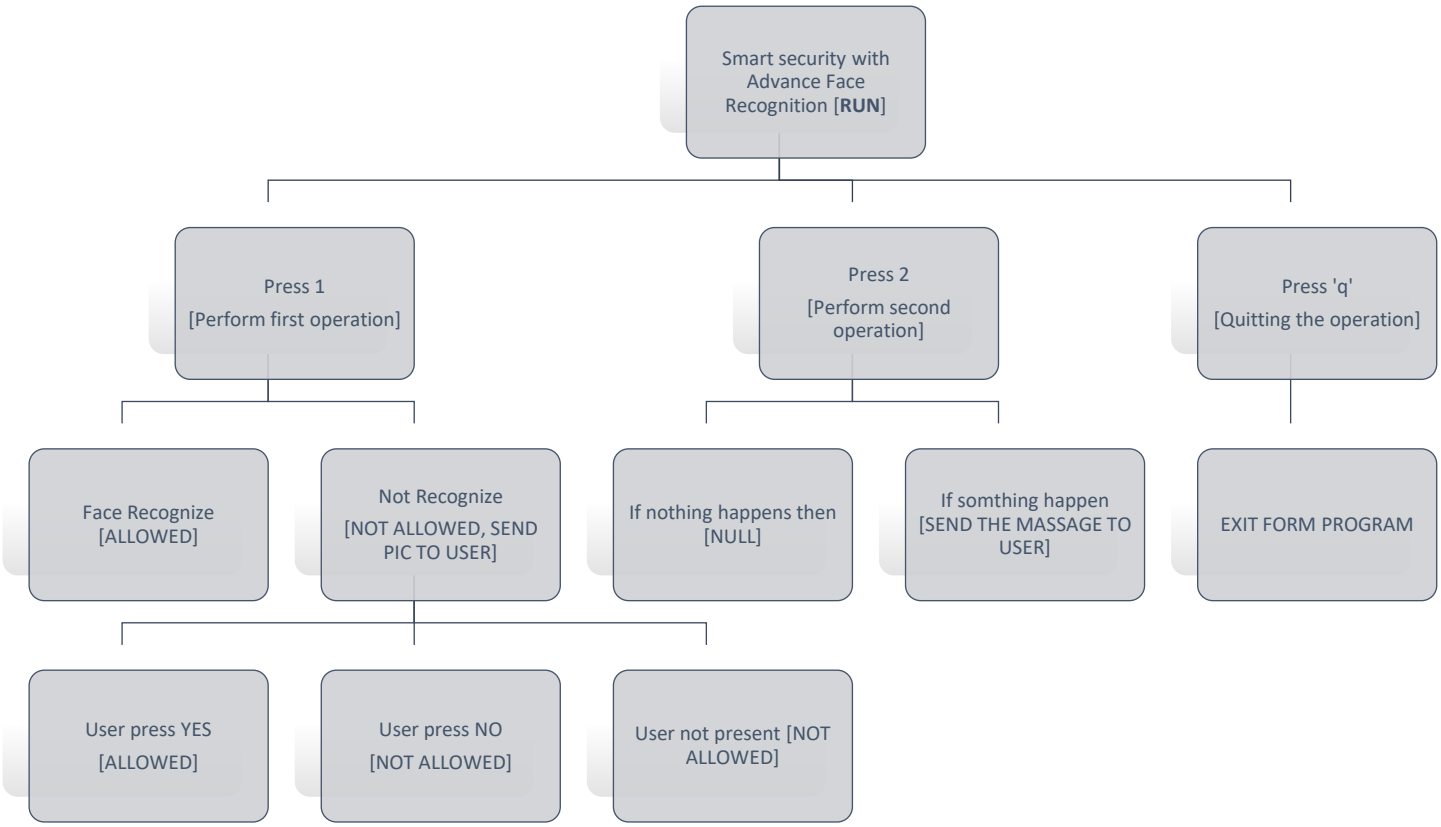
After accruing all the necessary tools, now we are going to design and develop our Project.

First, we create a data flow diagram to understand the project structure and then according to the that diagram we are create our project.

a. Data flow diagram

b. Working process with Codes

3-a. Data Flow Diagram



Data Flow Diagram of **Smart Security cam with advance face Recognition**

3-b. Working Process with Codes

As you know that we have two operations to complete.

So, if you run the application, it will ask for first operation or for second operation.

```
from pacs.Branch.secure import secure
from pacs.Branch.faceRecognition3 import
faceRecognition3
```

These two lines import the function of the two operations.

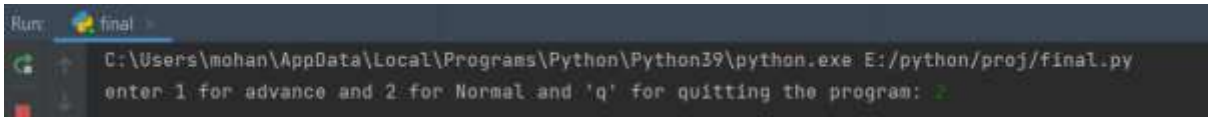
```
while 1:
    x = input("enter 1 for first and 2 for
second and 'q' for quitting the program: ")
    if x == '1':
        f = faceRecognition3()
        f.adva()

    elif x == '2':
        normal = secure()
        normal.norm()

    elif x == 'q':
        break
```

press 1 for first operation

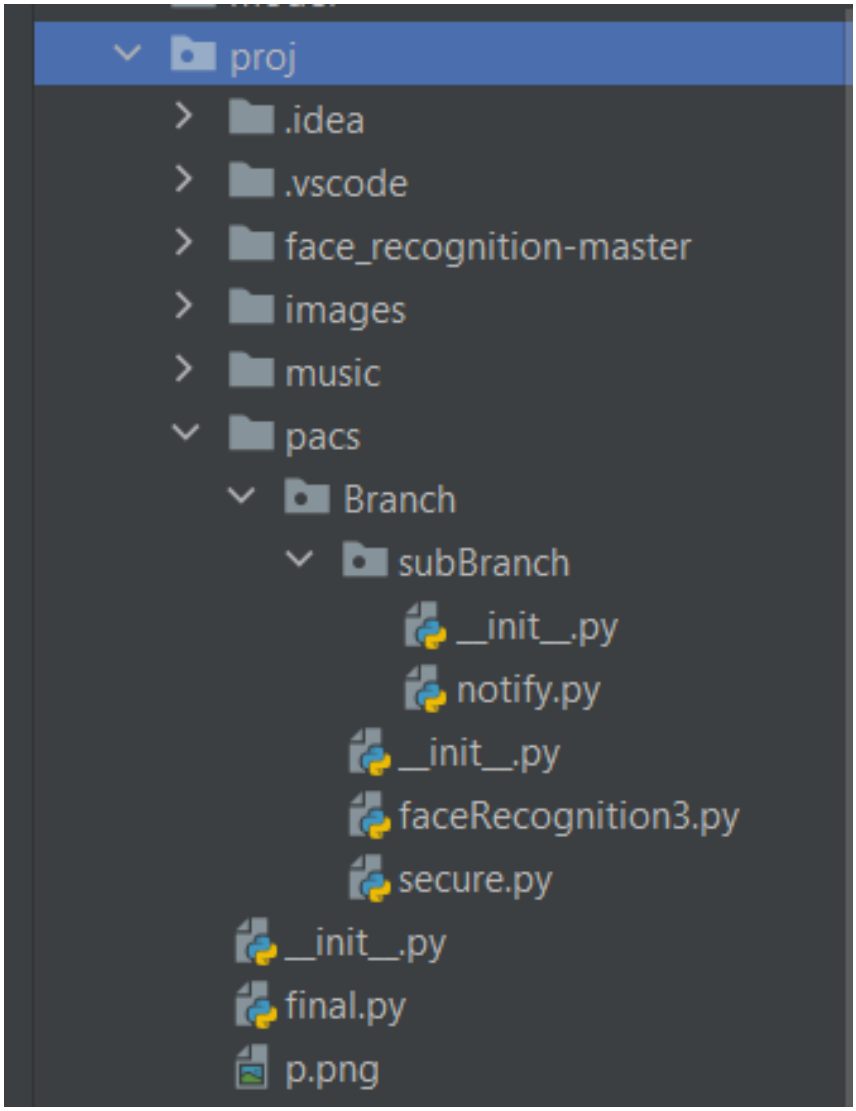
press 2 for second operation

press q for quitting the program

When we run the program the output:



Then according to the user preference, he can select the operation which he intends to select.



This is the structure of our total project files.

FIRST OPERATION:

❖ First, we have to import necessary libraries which are used to build our program.

```
oimport cv2
  import face_recognition
  import numpy as np
  import time
  import os
```

❖ Choose the camera by entering the camera number. In my case I use my webcam so I enter 0

```
o def __init__(self):
      self.path = 'images'
      self.images = []
      self.personName = []
      self.myList = os.listdir(self.path)
      self.cam = int(input("enter camera:
  "))
      # self.cam = input("enter camera:
  ")
```

❖ TRAINING THE FACE

```
o for cu_img in self.myList:
      current_img =
  cv2.imread(f'{self.path}/{cu_img}')
      self.images.append(current_img)  #
  It will read the images & store in a
  image list

  self.personName.append(os.path.splitext
  (cu_img)[0])  # It will get the image
  name & store in a personName list
```

```
# this function used to encode the face
def faceEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img,
cv2.COLOR_BGR2RGB)
        encode =
face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList

encodeFaces =
faceEncodings(self.images)
print("ALL ENCODINGS COMPLETE .....")
```

❖ After Recognize the face now we have to set a condition. Whether the detect face present at the date base or not.

```
o while 1:
    _, frame = cap.read()
    frame = cv2.resize(frame, (600,
400))
    _, fr = cap.read()
    fr = cv2.resize(fr, (600, 400))
    originalFrame = frame.copy()
    faces = cv2.resize(frame, (0, 0),
None, 0.25, 0.25)
    faces = cv2.cvtColor(faces,
cv2.COLOR_BGR2RGB)

    facesCurrentFrame =
face_recognition.face_locations(faces)
# It will give the face location
    encodeCurrentFrame =
face_recognition.face_encodings(faces,
facesCurrentFrame)  # It will give the
```

```
face encodes

    # this loop used to compare and
give distance between data faces with
current live face
    for encodeFace, faceLoc in
zip(encodeCurrentFrame,
facesCurrentFrame):
        matches =
face_recognition.compare_faces(encodeFa
ces, encodeFace)
        faceDis =
face_recognition.face_distance(encodeFa
ces, encodeFace)
```

- The code is to compare the current face which appear in the camera with the faces that are present at the data base.

```
matchIndex = np.argmin(faceDis)

if matches[matchIndex]:
    for e_fr in faceDis:
        if e_fr <= 0.45:
            name =
self.personName[matchIndex].upper()
```

- This code checks the distance between the faces. Minimum distance will be the recognized face.
- Some cases the face is also recognize even if that face is not present at the date. That type of faces is closely similar to the faces that are present at the database.
- For that reason, I minimize the distance condition. Now it will recognize if that particular person is present otherwise not.
- I use this condition:
    - `if e_fr <= 0.45:`

❖ Now if that face is present at the database, then the door will be unlocked for 5 second and locked.

```
if matches[matchIndex]:
        for e_fr in faceDis:
            if e_fr <= 0.45:
                name =
    self.personName[matchIndex].upper()
                print(f'{name} is here,
    door unlocking for 5 second')
                time.sleep(3)
                print('locked')
                break
```

This is the picture camera seen and recognize it.

Output:

```
enter 1 for advance and 2 for Normal and 'q' for quitting the program: 1
enter camera: 0
ALL ENCODINGS COMPLETE .....
DIBYA is here, door unlocking for 5 second
locked
```

- Now if the person who comes his face is not present at the date base, then it will take pic of that person and send it to the user smart phone with an option yes or no.

```
import cv2
from datetime import datetime
```

- We need to import these two libraries for send the pic to the user smart phone with a time stamp

```
class notify:
    def __init__(self, cap):
        self.cap1 = cap

    def pic(self, cap, fr):
        self.cap1 = cap

        while 1:
            original_frame = fr.copy()
            cv2.imwrite('p.png',
original_frame)

            permission = input("If you
want to allow this person press 'y' if not then
'n': ")
```

- By the help of this code the massage will shown to the user with the pic and options.

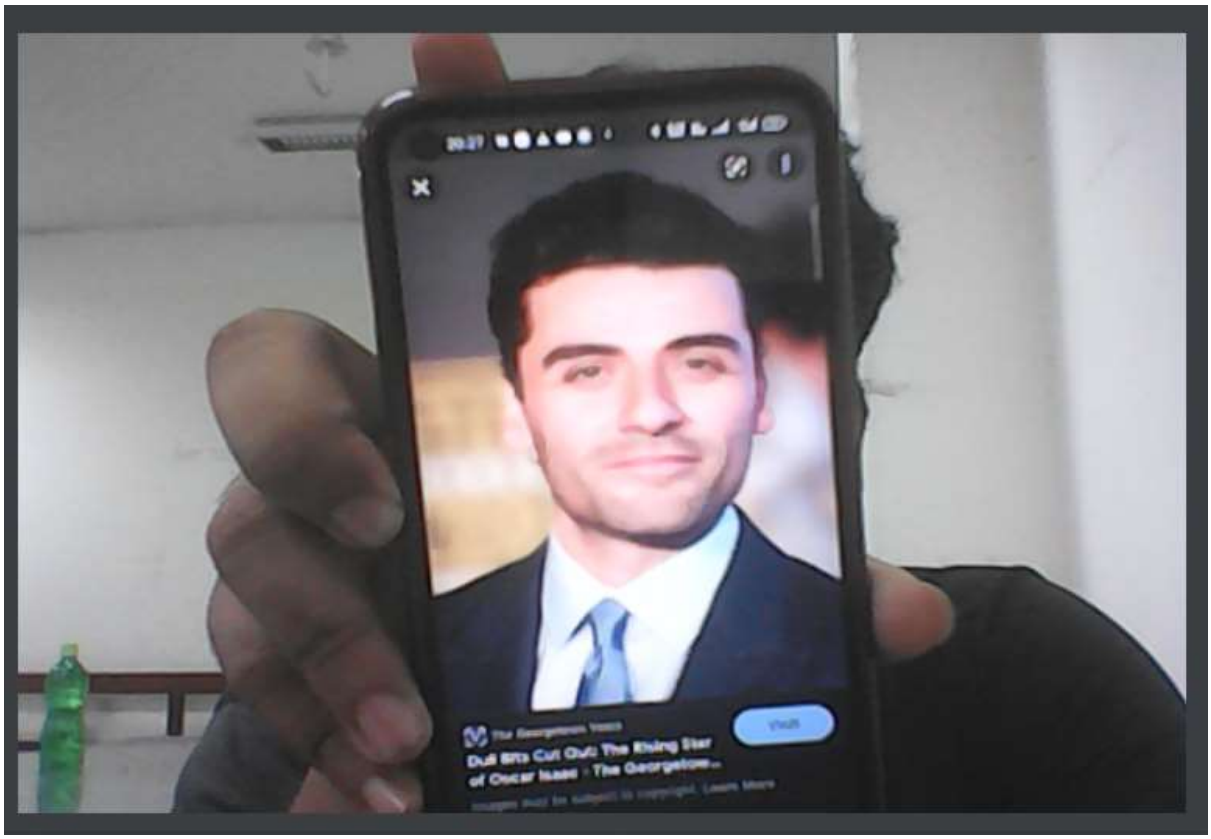
```
if permission == 'y':
    print ("Door is unlock for you 5
```

```
  sec.")
  elif permission == 'n':
      print ("you are not allowed....")

  break
```

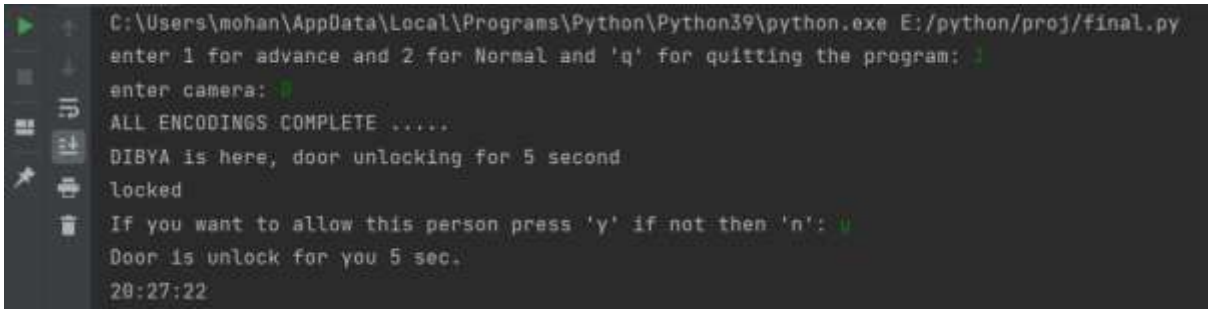
- BY the help of this code, it will be decided whether the person will enter or not.

```
o now = datetime.now()
  current_time = now.strftime("%H:%M:%S")
  print(current_time)
```

- This gives the exact time stamp of that person



- This is the pic of **actor Oscar Isaac** whose pic is not present in our data base.
- So, it shows this output:



- Now if the user is not present at near the smart phone, then this code will execute after 10 second and door will be locked after 10 second.

```
from threading import Timer

def ti(answer):
    if answer == 'y':
        print ("Door is unlock for you 5 sec.")
    else:
        print ("you are not allowed....")

timeout = 10
# t = Timer (timeout, print, ['Sorry, times up'])
t = Timer (timeout, quit ())
t.start()

answer = input (f"If you want to allow this person press 'y' within {timeout} second: ")
ti(answer)
t.cancel()
```

  - The whole user's smart phone related code we written in a separate .py file for more transparency.

- We have to import that file to the current file.

```
o from ..Branch.subBranch.notify import
  notify
```

- the following code will be the library of the above imported file.

```
o else:
      pis = notify(cap)
      pis.pic(cap, fr)
```

- With that note out first operation is complete and for that we have to close our camera.

```
o cv2.imshow("camera", originalFrame)
      if cv2.waitKey(1) == ord('q'):
          break

  cap.release()
  cv2.destroyAllWindows()
```

- If we press 'q' then the program will be quitting.
- And our first operation is complete.

SECOND OPERATION:

- First, we have to import necessary libraries which are used to build our program.

```
❖ #DETECTING IF SOMEONE BREAKING THE
  HOUSE
```

```
import cv2
import winsound
```

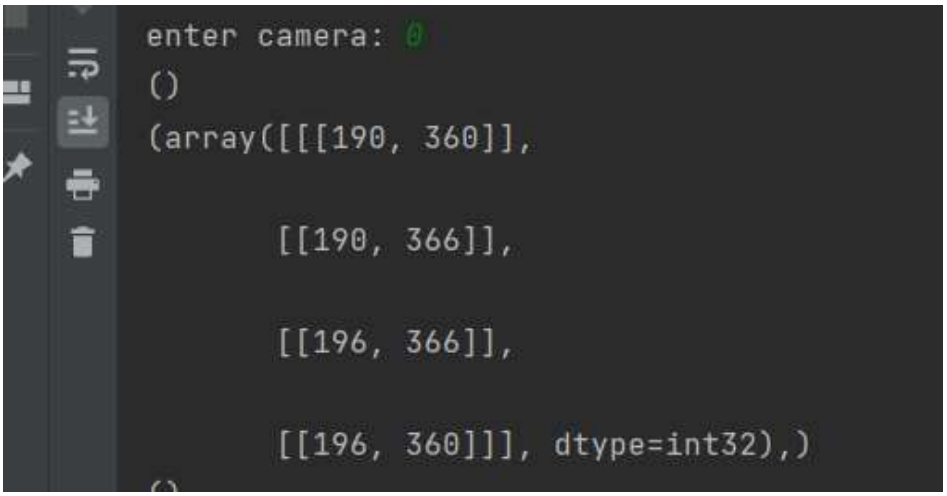
- Choose the camera by entering the camera number. In my case I use my webcam so I enter 0

```
o def __init__(self):
        self.path = 'images'
        self.images = []
        self.personName = []
        self.myList = os.listdir(self.path)
        self.cam = int(input("enter camera:
  "))
        # self.cam = input("enter camera:
  ")
```

- Now we have to give the find the contours of the live video which is capture by the operated camera.

```
❖ while True:
        _, frame1 = self.cap.read()
        frame1 = cv2.resize(frame1, (600, 400))
        _, frame2 = self.cap.read()
        frame2 = cv2.resize(frame2, (600, 400))
        diff = cv2.absdiff(frame1, frame2)
        gray = cv2.cvtColor(diff,
  cv2.COLOR_RGB2GRAY)
        blur = cv2.GaussianBlur(gray, (5, 5),
  0)
        _, thresh = cv2.threshold(blur, 20,
  255, cv2.THRESH_BINARY)
        dilated = cv2.dilate(thresh, None,
  iterations=3)
        contours, _ = cv2.findContours(dilated,
  cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

- For finding the contours we have to turn the live video and find the threshold of that live video camera.
- Contours is important for doing any kind of operation in the live video.
- Because It will convert the live video into numbers.



- This type of value shown if there is any disturbance occur in the camera.
- If in the camera there will be no action occur then the contours will be null and it shown () this.

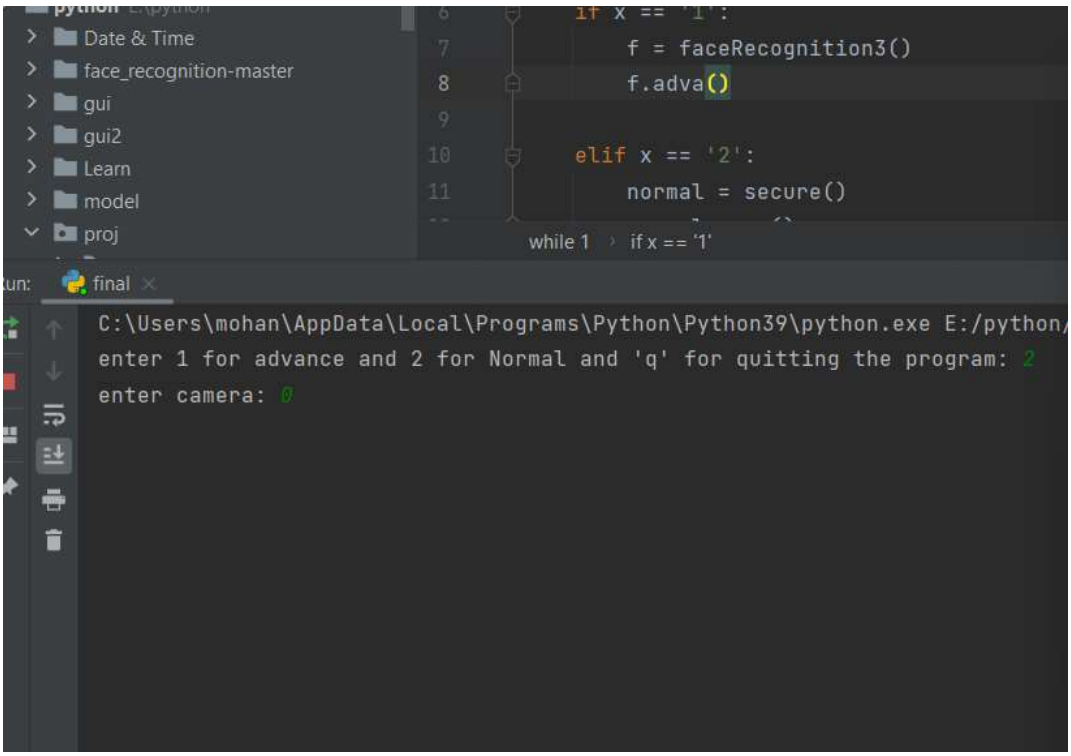
❖ Now we have minimized the contours rate because it will give value with just small number of disturbances, which will be hectic for use. Because it will give notification even if there is a small wind is flowing.

```
❖ for c in contours:
      if cv2.contourArea(c) < 5000:
          continue
```

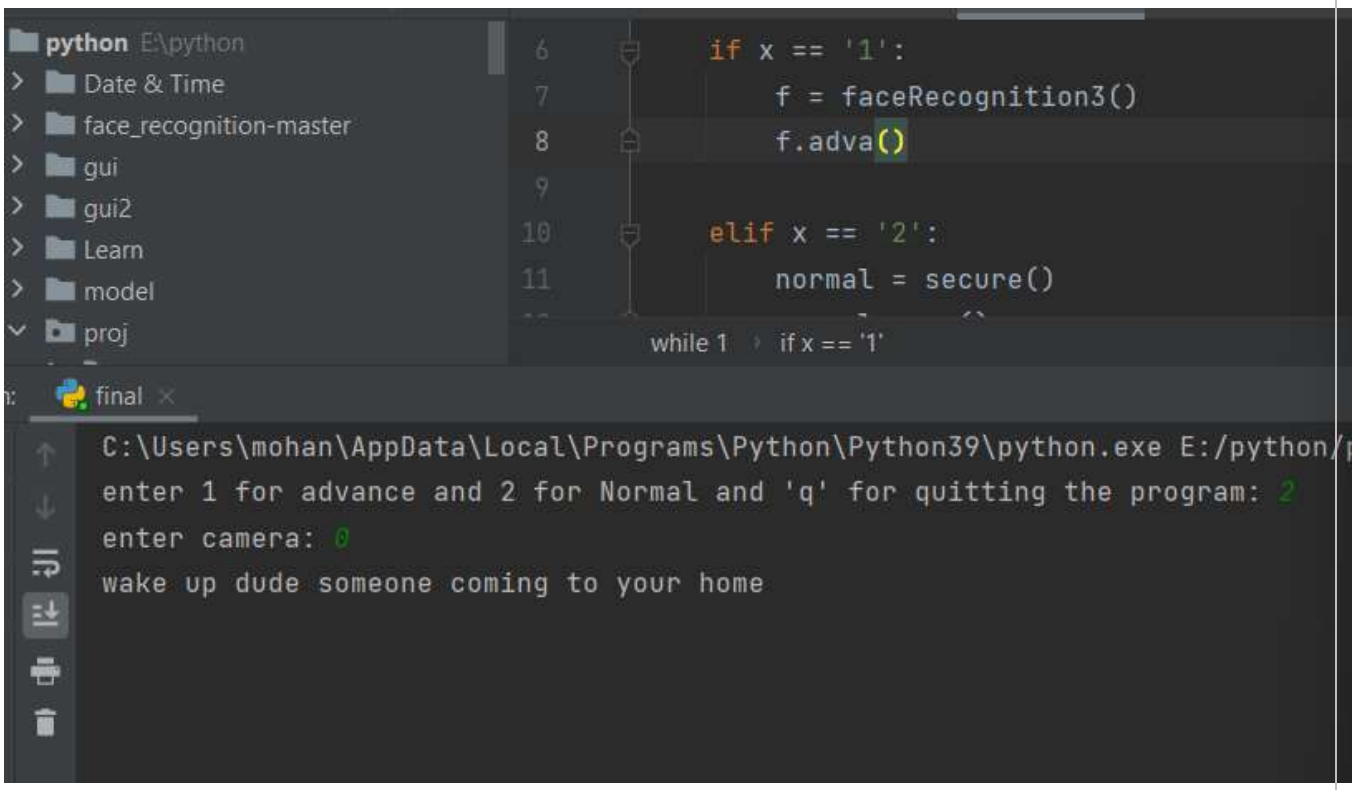
❖ Now if there is big disturbance occur then it will send the notification with a warning sound.

```
for c in contours:
    if cv2.contourArea(c) < 5000:
        continue
    x, y, w, h = cv2.boundingRect(c)
    cv2.rectangle(frame1, (x, y), (x + w, y
+ h), (0, 255, 0), 2)
    print("wake up dude someone coming to
your home")
    winsound.Beep(2500, 500)
    break
```

- The output is:
- If there is not any action going on the screen then the out screen will be empty



- If there is some action going on the camera then the output screen shows

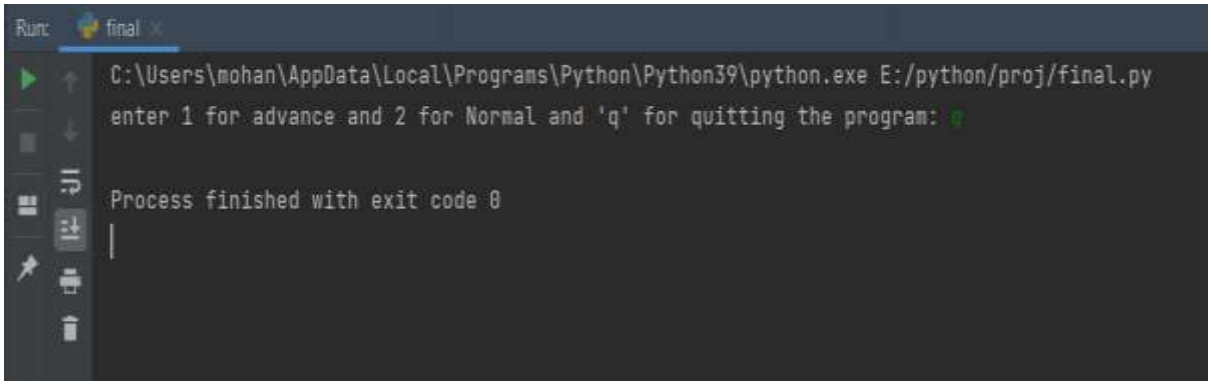


- With that note out first operation is complete and for that we have to close our camera.

```
o cv2.imshow("camera", originalFrame)
      if cv2.waitKey(1) == ord('q'):
          break

  cap.release()
  cv2.destroyAllWindows()
```

  - If we press ‘q’ then the program will be quitting.
  - And our second operation is also complete.

- And if we again press enter then the program will be quit. Or you will be exit from the program.



This is the full details of our project.

4.

FUTURE VISION

4.FUTURE VISION

Our **"SMART SECURITY CAM WITH ADVANCE FACE RECOGNITION"** is sole purpose is to protect our family, friends and valuable things from dangerous people.

So, as the people becoming smarter, we have to also become smart and digitalize our Home, Office, Shop, Factory etc.

That's why in the coming days we plane to advance our project by editing some new and advance features.

- In case of first operation, we are trying to add a feature where user can easily add his/her family or friends in the database to recognize the face.
- If someone try to force fully enter the house even if the door is locked then it'll contact to the nearest police station with the picture of that person.
- In case of second operation, we are trying to send the live footage of the camera if some disturbance occurs.
- Now this application connects to only one camera at a time. So, we are trying connect multiple cameras to perform multiple operation at a time.
- We are trying to install a speaker system also along with the camera so that if the person presents at the phone tell anything it will heard by the person present at the camera.
- If multiple people try to entering the house and in those people some of faces are recognized and some of are not then it will not

open the door and send the pic to the user/owner and it will be allowed by the user.

- We are trying to add schedule so that it on/off/change to other operation automatically when the time is come.

5. CONCLUSION

This project is designed to meet the requirements of all the people who loves their family, friends and colleagues. It makes feels safe and protective.

It provides the facilities of advance smart security for the users. It has been developed in **PYTHON** keeping in the specifications of the system. For designing the system, we have used simple data flow diagram.

Over all the project teaches us the essential skills like:

- Understanding about the Machine learning.
- Know how to handle advance python program in real time.
- Understanding the role of smart security in our life in this digitalized world.

6. BIBLIOGRAPHY

- Face Recognition:
  - https://realpython.com/face-recognition-with-python/
  - https://github.com/ageitgey/face_recognition
- OpenCV:
  - https://github.com/opencv/opencv
  - https://www.javatpoint.com/opencv#:~:text=OpenCV%20is%20a%20Python%20open,Learning%2C%20face%20recognition%2C%20etc.
- KivyMD:
  - https://kivymd.readthedocs.io/en/latest/
- Smart home security:
  - https://www.tsp.space/smart-home-blog/why-your-homes-cyber-security-is-more-important-than-ever/
- Dlib and Cmake:
  - https://pypi.org/project/dlib/
  - https://www.geeksforgeeks.org/how-to-install-cmake-for-windows-in-python/