THIRD GENERATION ATM MACHING USING ADVANCE IMAGE PROCESSING

A PROJECT REPORT

Submitted by

ARUN KUMAR M [211419104024]
BALASUBRAMANIYAN R [211419104038]
BANDICHAITANYAKRISHNAREDDY [211419104039]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"THIRD GENERATION ATM MACHINE USING ADVANCE IMAGE PROCESSING"** is the bonafide work of **"ARUN KUMAR M (211419104024) , BALASUBRAMANIYAN R (211419104038), BANDI CHAITANYA KRISHNA REDDY (211419104039)** who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

DR.P.J.SATHISH KUMAR,M.E.,Ph.D,
SUPERVISOR
ASSOCIATE PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the End Semester Project Viva-Voce Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **ARUN KUMAR M (211419104024), BALASUBRAMANIYAN R (211419104038), BANDI CHAITANYA KRISHNA REDDY (211419104039)** hereby declare that this project report titled "**THIRD GENERATION ATM MACHINE USING ADVANCE IMAGE PROCESSING**", under the guidance of **Dr.P.J.SATHISH KUMAR M.E.,Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ARUN KUMAR M

BALASUBRAMANIYAN R

BANDI CHAITANYA KRISHNA REDDY

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR, M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our parents, friends, project Guide **DR.P.J.SATHISH KUMAR M.E.,Ph.D.**, and coordinator **Dr.N.PUGHAZENDI M.E., Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

ARUN KUMAR M

BALASUBRAMANIYAN R

BANDI CHAITANYA KRISHNA REDDY

ABSTRACT

Banks give ATM cards to client to mileage the services like cash pullout, Leg change, balance inquiry etc. But physical cards have some problems. It can be stolen, skimmed, reproduced, commandeered, damaged or expired. Due to this problem, we need to suppose an alternate way to give better security. Numerous experimenters are allowing about card less sale through ATM proposed a abstract model for card less Electronic ATM through which client can do cash pullout, balance inquiry, fund transfer etc. We've anatomized their protocol and plant some excrescencies on this. This protocol doesn't specify what if it's off us sale. Either, guests get different orders of services but this protocol cannot determine which client will get which order of services. For this application we are using face detection with the account number, aadhar number, pin number, mobile number and name. When we recognize the face then it will give access to enter the ATM details. After that it will send message to the phone number. if we enter the password it will give transaction access for this purpose we are using deep learning.The face recognition algorithm will use deep learning techniques to identify the user's face and match it against a database of known faces. The fingerprint recognition algorithm will use minutiae-based techniques to identify the unique features of the user's fingerprints and match them against a database of known fingerprints. The iris recognition algorithm will use pattern recognition techniques to identify the unique features of the user's iris and match them against a database of known irises.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS

AI	Artificial Intelligence
OTP	One Time Password
CNN	Convolutional Neural Network
ATM	Automated Teller Machine
DNN	Deep Neural Network
PCA	Principal Component Analysis
FAR	False Acceptance Rate
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
IOC	Intersection Over Union
NMS	Non-Maximum Suppression
MTCNN	Additive Angular Margin Loss Face
LFW	Labeled Faces in the Wild
CACD	Cross-Age Celebrity Dataset
IJB-A	IARPA Janus Benchmark A
LFW-6000	Labeled Faces in the Wild 6000
YOLO	You Only Look Once
FPN	Feature Pyramid Network

1. INTRODUCTION

1.1 OVERVIEW

A computer-implemented method for card less use of an automated teller machine (ATM) is provided. The method includes receiving as an input, a user-identified ATM that the user wishes to use. The method also includes generating and transmitting a one-time password (OTP) for the user to enter at the identified ATM. The method further includes receiving and verifying the OTP entered into the ATM, and on successful verification, authorizing access to services available through the ATM, without use of a card to reduce the threat involved in ATM machines that were installed in remote area, also the issue related to fraudulent sale like misusing others card to withdraw plutocrat and etc. So in order to overcome these challenges, we've developed result that will work the ML & AI to circumscribe card access to only the authorized druggies those are linked by face recognition algorithm.

This method is useful in many fields such as the military, for security, schools, colleges and universities, airlines, banking, online web applications, gaming etc. this system uses powerful python algorithm through which the detection and recognition of face is very easy and efficient.

Surveillance cameras are an essential security precaution in all public places. In a centralized surveillance system, videos collected from different cameras are stored in a centralized server. If any security threat is caused by the presence of an individual in a particular place, the law enforcing team will have to identify the current location of the particular person involved in the event as early as possible. Though the videos collected from surveillance cameras help to identify the person's presence in a location, checking the person of interest from a large collection of videos is a herculean task if it is done manually, The complexity of the task depends on the number of cameras involved in the surveillance process. Deep Learning-based video analytics can help us to automate this identification task. Deep Learning is a powerful tool to do image classification.

1.2 PROBLEM DEFINITION

Skimming is a common form of ATM fraud where criminals attach a device to the ATM card reader to capture card information, which is then used to create counterfeit cards. By incorporating image processing techniques such as object detection, facial recognition, and OCR (Optical Character Recognition), ATMs can detect and prevent skimming attempts. For example, a facial recognition system could verify the identity of the user before allowing a transaction to occur, and an OCR system could verify the authenticity of the card being used. Overall, utilizing image processing can improve the security of third generation ATMs and provide a better experience for users.

2. LITERATURE SURVEY

2.1 ATM Security System using Fingerprint Authentication

Author Name **:** Christiawan, B. A. Sahar, A. F. Rahardian and E. Muchtar

Year of Publish **:**2018

Fingerprint Based ATM is a desktop application where fingerprint of the user is used as a authentication. The finger print minutiae features are different for each human being so the user can be identified uniquely. Instead of using ATM card Fingerprint based ATM is safer and secure. There is no worry of losing ATM card and no need to carry ATM card in your wallet. You just have to use your fingerprint in order to do any banking transaction.

2.2 An IOT Based ATM Surveillance System

Author Name **:** V. Jacintha, J. Nagarajan, K. T. Yogesh, S. Tamilarasu and S. Yuvaraj

Year of Publish **:** 2019

In the present scenario, majority of the population uses the ATM machine to withdraw cash. At the same time, there are many ATM robberies that have occurred in many areas, even if the CCTV cameras are placed in the ATM center. Hence the security system needs tobe changed. In order to reduce these kinds of robberies, we present a security system for ATM theft by using a smart and effective technology. This system also analyses various physical attacks on ATM's. In our proposed system we use Face Recognizing Camera to capture the face of the person, who is entering. Tilt and vibration sensors are used to detect the irregular activities that are done on the ATM machine.

2.3 Advanced ATM System Using Iris Scanner

Author Name **:** Banerjee, S. Mookherjee, S. Saha, S. Ganguli, S. Kundu

Year of Publish **:** 2015

Nowadays we are experiencing a radical increase in skimming in the Automated Teller Machine (ATM) systems. So, actuation in advancement and security of the ATM machines is required. An automated teller machine (ATM) is an electronic telecommunications device that helps customers of banking departments in transactions and transfer of money in their accounts. The customer enters their unique personal identification number (PIN), i.e. stored in the chip of the card. Due to an increase in the installation of ATM and the number of ATM cardholders, the number of cases of fraudulence has also increased radically. The advancement in technology has resulted in an increase in various skimming activities. So, developments are incorporated in the existing systems to make it more secure, convenient and reliable. The employed secured system must have high speed and must be durable. The presented design is unique because of biometric scanners such as Iris scanner and the two-way check with fingerprint scanner makes it more reliable.

2.4 Design and implementation of ATM alarm data analysis system

Author Name **:** Y. Cheng, W. Shang, L. Zhu and D. Zhang

Year of Publish 2015

Nowadays, people pursuit of fast and convenient way of life, fast and convenient service of ATM is made for people to avoid waiting in line at the bank for a long time. In order to serve people conveniently, it is need to monitor the ATM equipment to guarantee its normal operation, and deal with the unexpected problems in time. Therefore, this paper builds a cloud platform for alarm service, does some alarm analysis, which appears at different times in different locations of the ATM machine. This can provide better service for ATM users. This system is called ATM Alarm Data Analysis System.

2.5 A New Vision for ATM Security Management: The Security Management Platform

Author Name **:** C. Porretti, R. Lahaije and D. Kolev

Year of Publish **:** 2017

The aim of this paper is to describe a new vision for ATM Security Management that is proposed by the GAMMA project, and implemented by its "core" prototype called Security Management Platform. GAMMA is an FP7 project with the goal of developing solutions capable to manage emerging ATM vulnerabilities. The GAMMA vision recognizes the opportunities opened by a collaborative framework for managing security, building a solution based on the self-protection and resilience of the ATM system, with the possibility to share security information in a distributed federated environment. This concept is implemented with the Security Management Platform prototype, and can be conceptualized as a network of distributed nodes embedded within the ATM system, providing interfaces to (ATM) internal and external security stakeholders.

2.6 Face Recognition Application for Automatic Teller Machines (ATM)

Author Name **:** H. R. Babaei, O. Molalapata and A. A. Pandor

Year of Publish **:**2014

Automatic Teller Machines (ATMs) are widely used in our daily lives due to their convenience, wide-spread availability and time-independent operation. Automatic retraction of forgotten card or cash by ATMs is a problem with serious consequences (lost time and money), typically caused by user inattention/negligence. In this work, we propose a scheme in which the retraction rate of an ATM is decreased using face detection and recognition methods via ATM's built-in camera. The short time frame of ATM usage and severe motion artifacts make this problem very different from an ordinary face authentication or face recognition problem. We evaluate the proposed system under challenging conditions of real ATM usage. The experimental results on multiple databases reveal that our proposed system is promising for mitigating card/cash forgetting issue and improving ATM user experience.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In existing system RFID card is used as ATM card, IR sensor in order to sense the presence of the card holders and to turn on Fan and Light, if ATM is tampered then SMS is sent to two main stations via GSM. Based on WI fall detection get security, that network access is not that much secured. In existing system RFID card is used as ATM card, IR sensor in order to sense the presence of the card holders and to turn on Fan and Light, if ATM is tampered then SMS is sent to two main stations via GSM. The biometrics like finger print and eye ball authentication are prone for easy spoofing. Used SVM classification for face recognition.

3.1.1 Disadvantages of Existing System

While object detection is a valuable and widely used technology, there are some potential disadvantages to consider:

- Complexity: Object detection involves complex algorithms and requires significant computational resources. This means that it can be time-consuming and expensive to implement and run.
- Accuracy: Object detection algorithms may not always correctly identify objects in images or video footage. This can lead to false positives (objects mistakenly identified as present) or false negatives (objects not detected).
- Data requirements: To train an object detection model, a large amount of labeled data is required. This can be difficult to obtain and can be expensive to collect and annotate.
- Variability: Objects can appear in different sizes, orientations, and lighting conditions, which can make it challenging for object detection algorithms to identify them accurately.

3.2 PROPOSED SYSTEM

In proposed method, apply deep neural network and face of haar cascade classifier.
Then the OTP based transaction is very simple with the more security.

- Face recognition
- OTP based
- Cascade classifier
- Video streaming
- Pre-process

3.2.1 Advantages of Proposed System

- Enhanced situational awareness: Aerial imagery can provide a broad view of a large area, enabling authorities to identify and track human activity over a wide geographic area.
- Improved security: Identifying humans in aerial images can enhance security measures, such as identifying potential threats or monitoring crowds.
- Increased efficiency: By automating object detection tasks, businesses and organizations can increase efficiency and reduce costs.

3.3 FEASIBILITY STUDY

A smart ATM can be a viable solution for banks to enhance customer experience and reduce operational costs. However, before implementing a smart ATM, it is essential to conduct a feasibility study.

Firstly, from a technical perspective, smart ATMs require specific hardware and software to operate. They need to integrate with the bank's core banking system, ATM network, and mobile banking app to provide seamless transactions. They also require stable and secure internet connectivity to communicate with banking systems and enable online transactions.

Secondly, from an economic perspective, the cost of implementing a smart ATM should be justified by the potential benefits. Smart ATMs are more expensive than traditional ATMs due to their advanced features, such as biometric authentication and real-time updates. Banks should evaluate the return on investment and weigh the cost of implementation against potential benefits, such as increased revenue from reduced operational costs and more customers.

3.4 SOFTWARE REQUIREMENTS

- Operating System : Windows 10 (64 bit)
- Software : Python
- Tools : Open CV

3.5 HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above
- GPU

3.6 TECHNOLOGIES USED

- Python
- Deep Learning

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artefacts or classes, in order to better understand, alter, maintain, or document information about the system. It is based on diagrammatic representations of software components.

Some UML diagrams are:

- Use case diagram
- Class diagram
- Activity diagram
- Collaboration Diagram
- Sequence Diagram

4.1.1 Use-Case Diagram

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

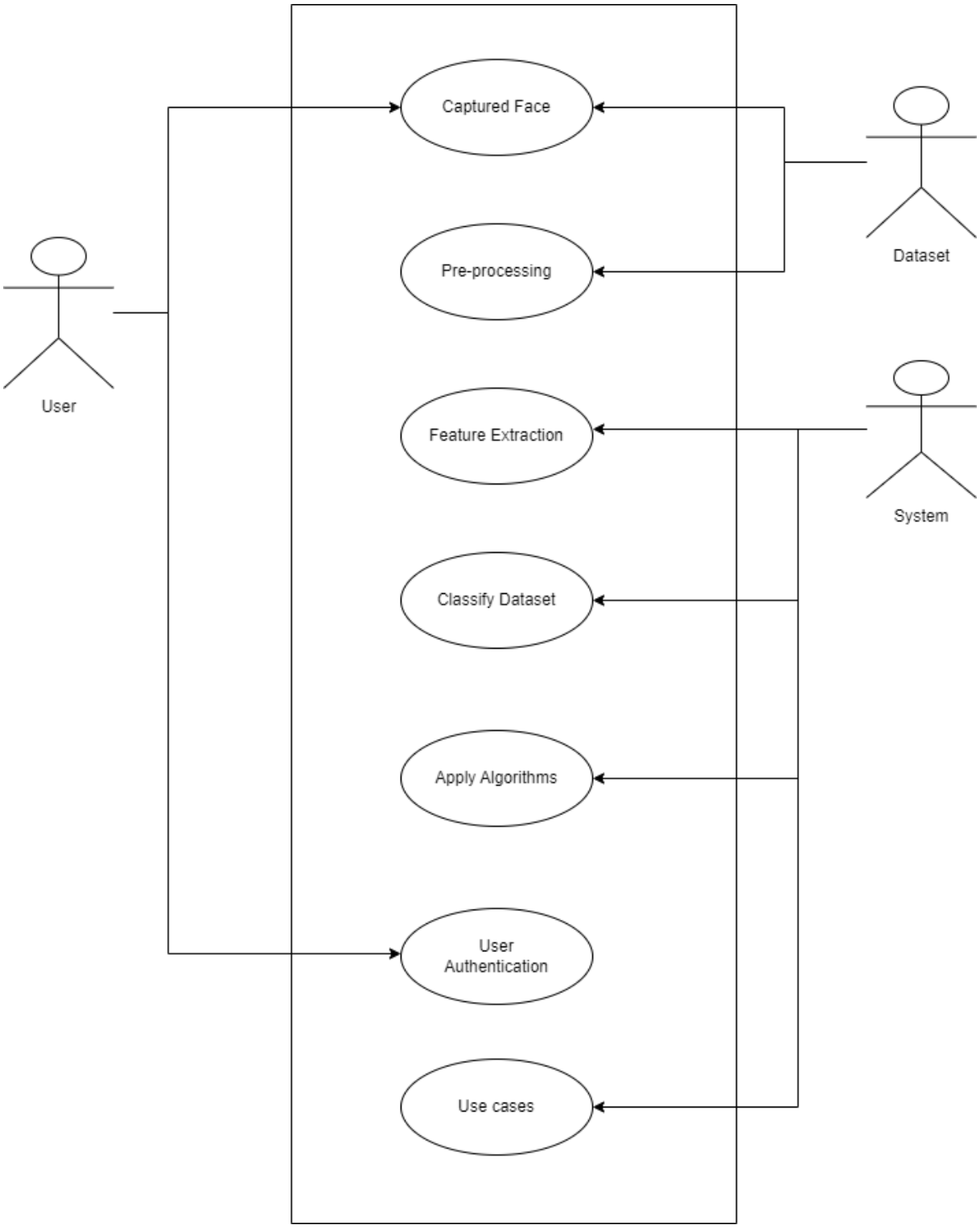


Fig 4.1.1 Use-Case Diagram

4.1.2 Activity Diagram

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow

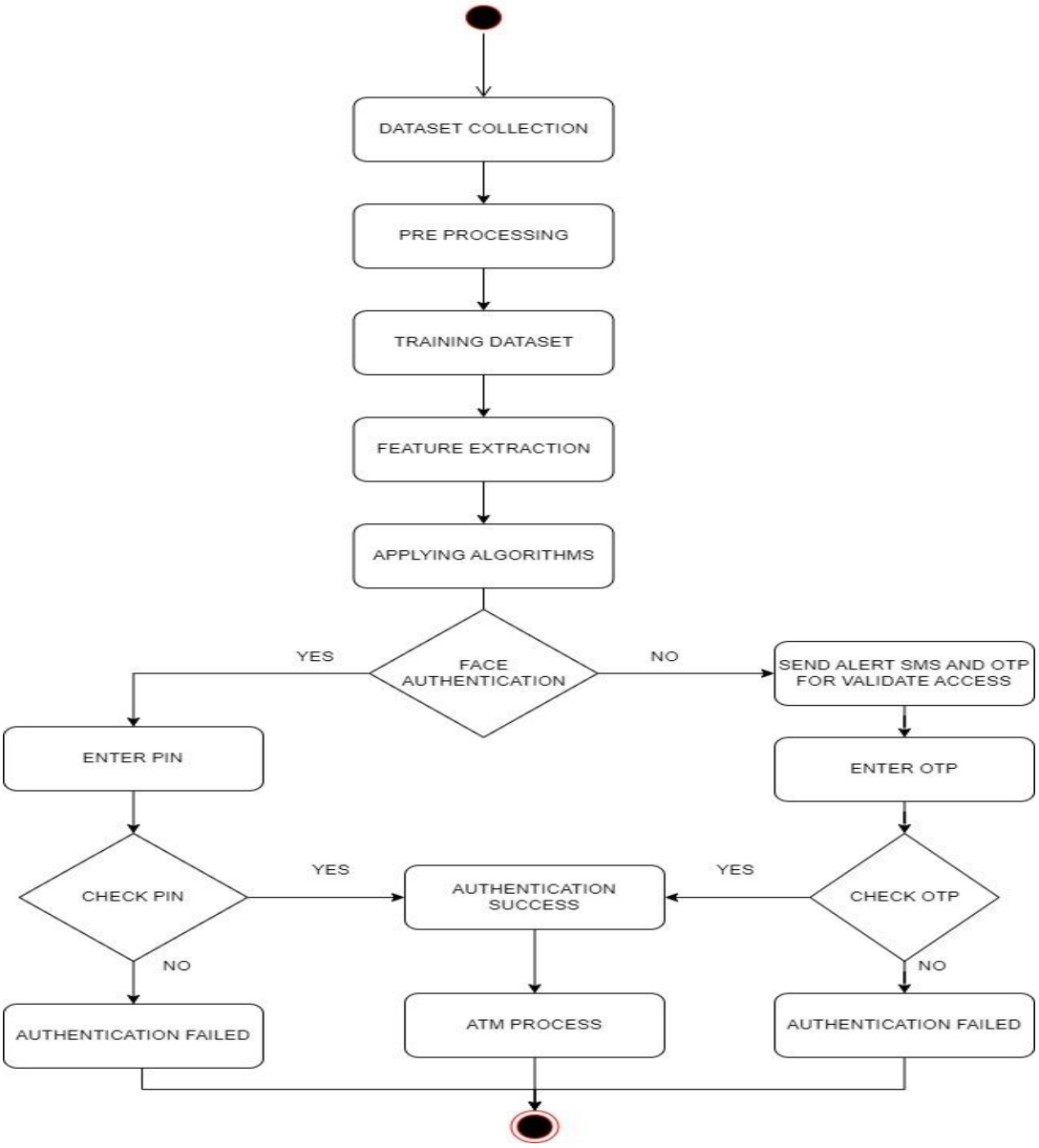


Fig 4.1.2 Activity Diagram

4.1.3 Class Diagram

A class diagram is a visual representation of class objects in a model system, categorized by class types. Each class type is represented as a rectangle with three compartments for the class name, attributes, and operations. Objects are represented as ovals that contain class names inside class name compartments.

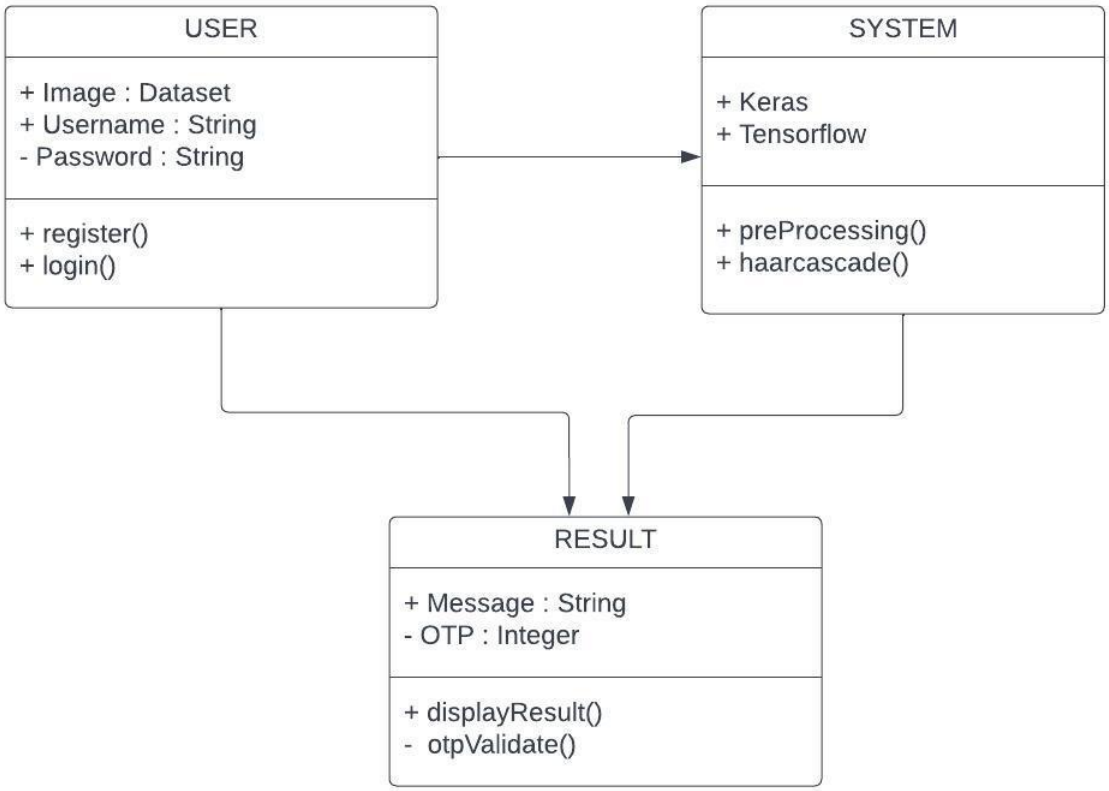


Fig 4.1.3 Class Diagram

4.1.4 Sequence Diagram

A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects. For example, lifelines in a sequence diagram for a banking scenario can represent a customer, bank teller, or bank manager.

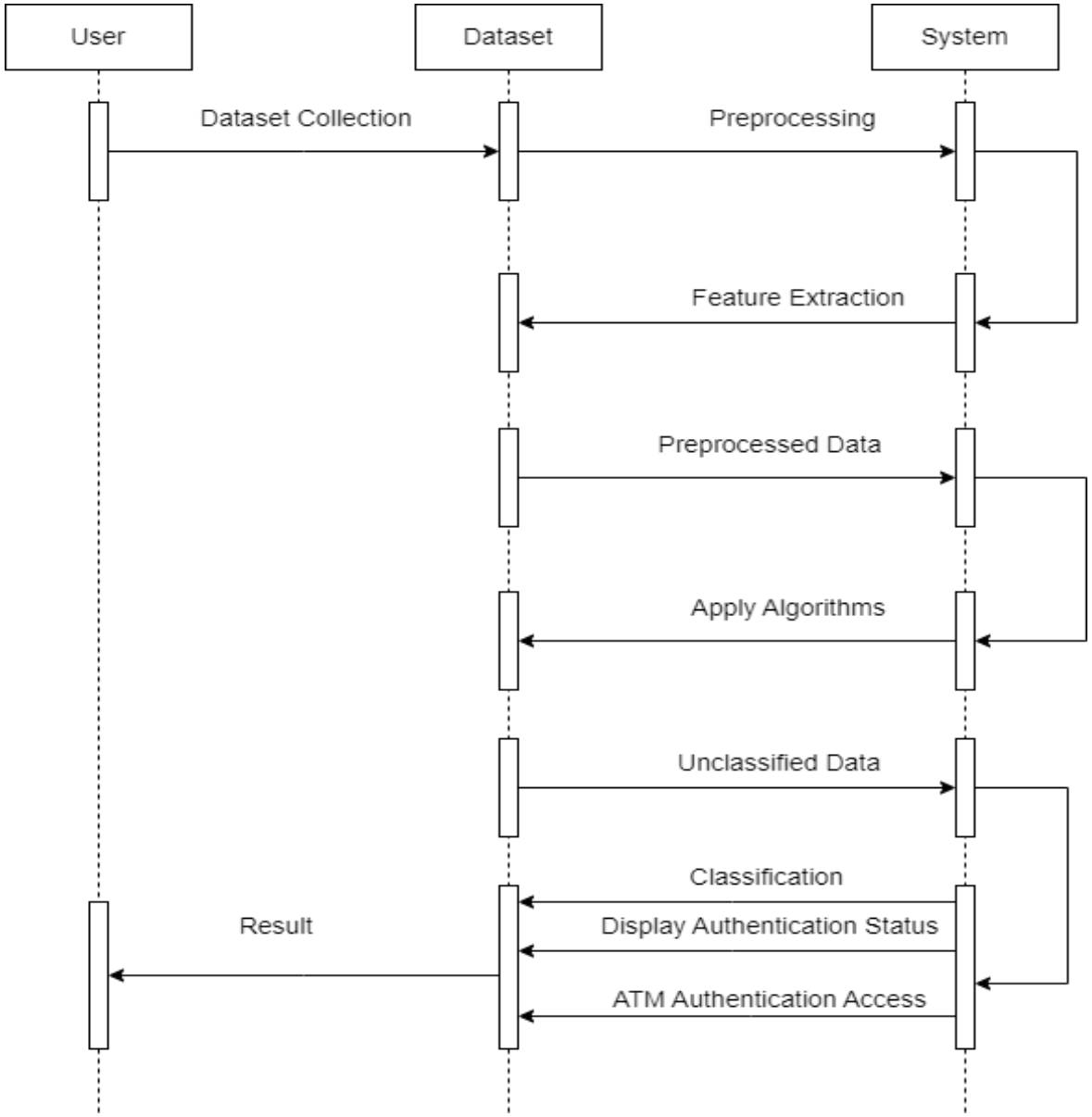


Fig 4.1.4 Sequence Diagram

4.1.5 State Chart Diagram

State chart diagram describes the flow of control from one state to another state. Statesare defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.

State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams

- To model the dynamic aspect of a system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

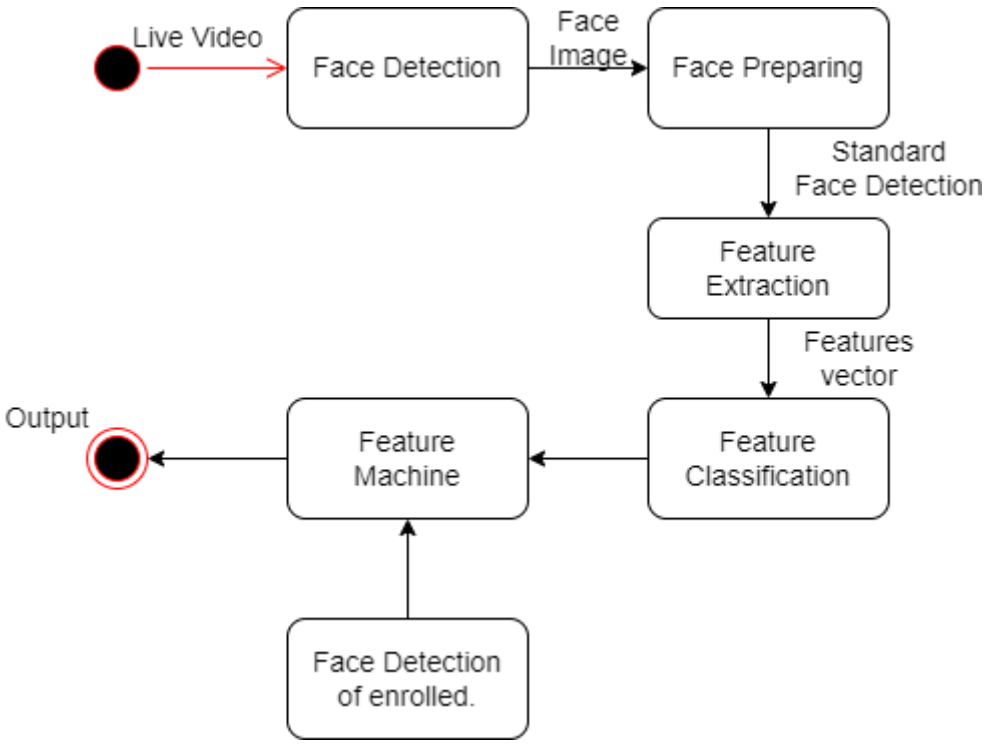


Fig 4.1.5 State Chart Diagram

4.1.6 Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviour is explained by the provided and required interfaces.

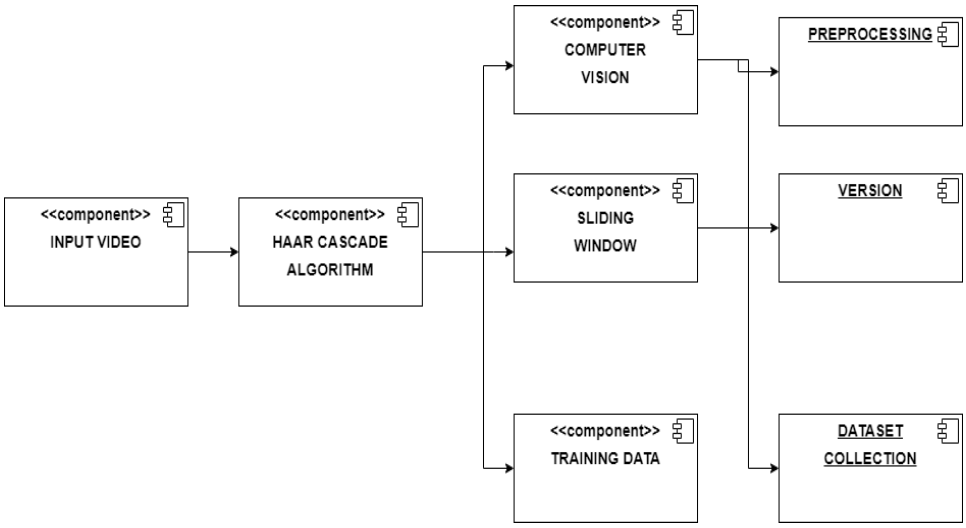


Fig 4.1.6 Component Diagram

4.1.7 Deployment Diagram

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

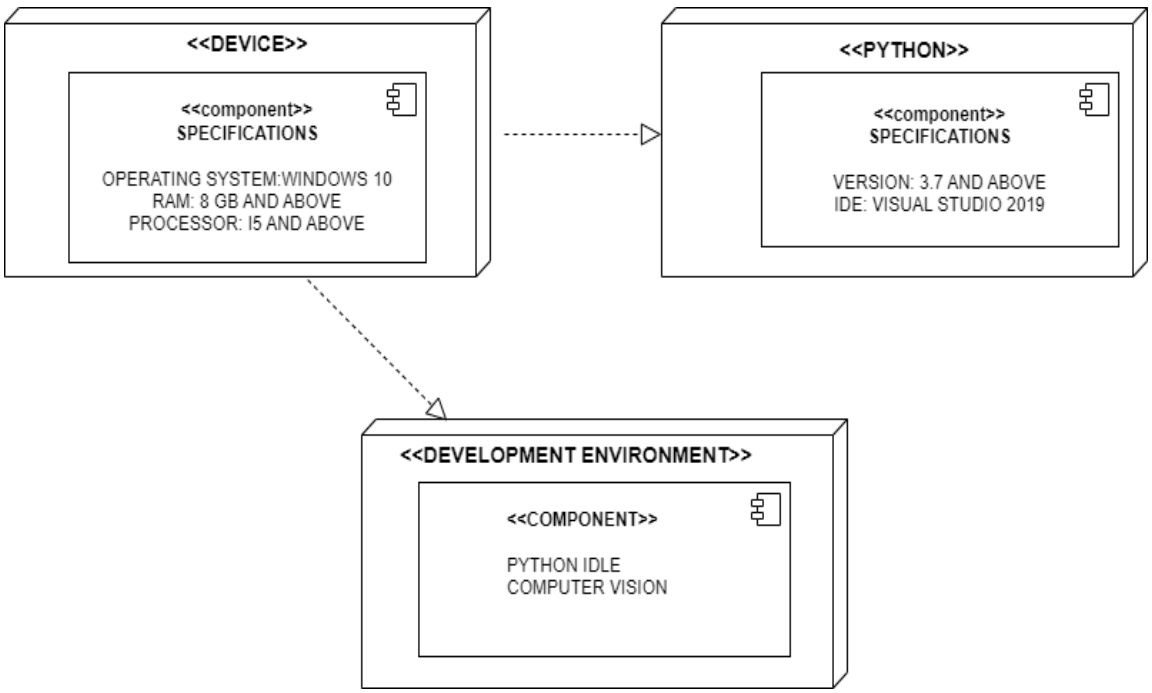


Fig 4.1.7 Deployment Diagram

5. SYSTEM ARCHITECTURE

5.1 ARCHITECTURE DIAGRAM

An architecture diagram is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements and components. It is also defined as a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.This diagram gives a top-level view of a software's structure. To elaborate, it generally includes various components that interact with each other and how the software interacts with external databases and servers.It's useful for explaining software to clients and stakeholders; and assessing the impact of adding new features or upgrading, replacing, or merging existing applications.

An architecture diagram typically consists of several key elements:

- Components: These are the building blocks of the system or application, such as servers, databases, APIs, and user interfaces.
- Relationships: These represent the connections and interactions between components, such as how data flows between servers and databases, or how users interact with the user interface.
- Layers: These represent the different levels or tiers of the system, such as the front-end, back-end, and database layers.
- Security and Access Controls: These represent the security measures put in place to protect the system from unauthorized access, data breaches or cyber-attacks.
- Deployment: This refers to the physical infrastructure on which the system or application is deployed, such as cloud-based servers or on-premises data centers.

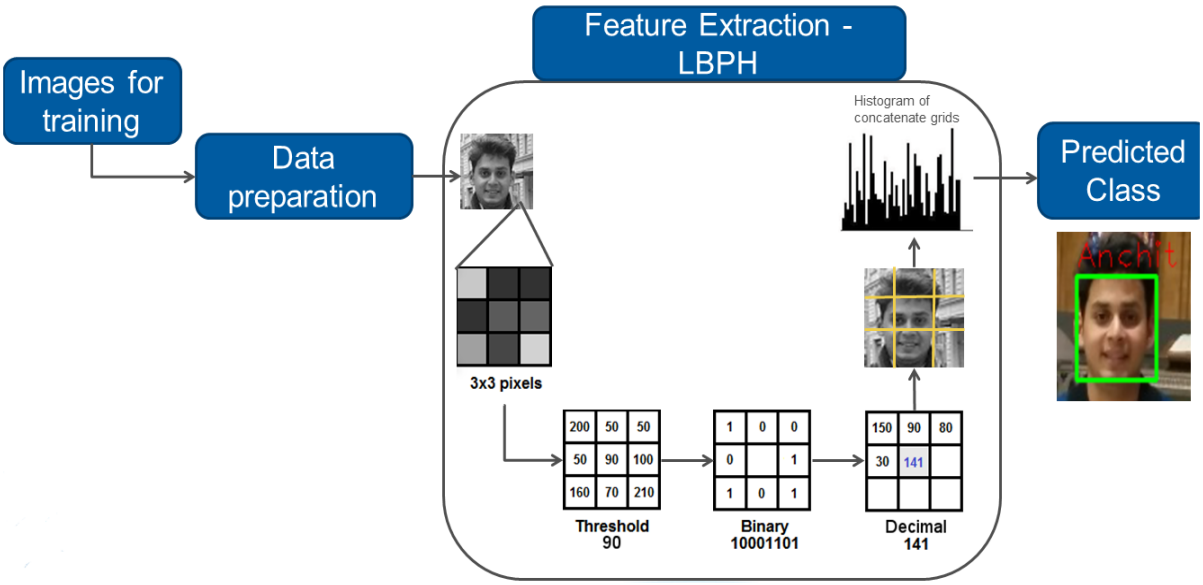


Fig 5.1 Architecture Diagram

The efficiency of object detection has evolved significantly with the emergence of convolutional neural networks (CNNs). AlexNet is one such model which has been shown to outperform most handcrafted models such as the VJ detector , the Scale Invariant Feature Transform (SIFT), and the Histogram of Oriented Gradients (HOG). Since then, CNNs have made a huge leap in object detection and many other computer vision applications such as feature extraction, autonomous driving, and others. Based on the region of interest (ROI), most deep learning architectures are categorized into two types: one-stage and two-stage architectures. Single-stage detectors are directly approached models without any intermediate object proposals called end-to-end object detection models whereas, two-stage object detection models have a two-way approach with a regional proposal stage followed by object detection and bounding box regression.

5.2 ALGORITHMS

5.2.1 HAAR CASCADE

Haar Cascades are machine learning object detection algorithms that are used to identify faces in an image or a real-time video. The Haar Cascade algorithm uses edge or line detection features that are proposed by Viola and Jones within their research paper named "Rapid Object Detection employing Boosted Cascade of Simple Features".

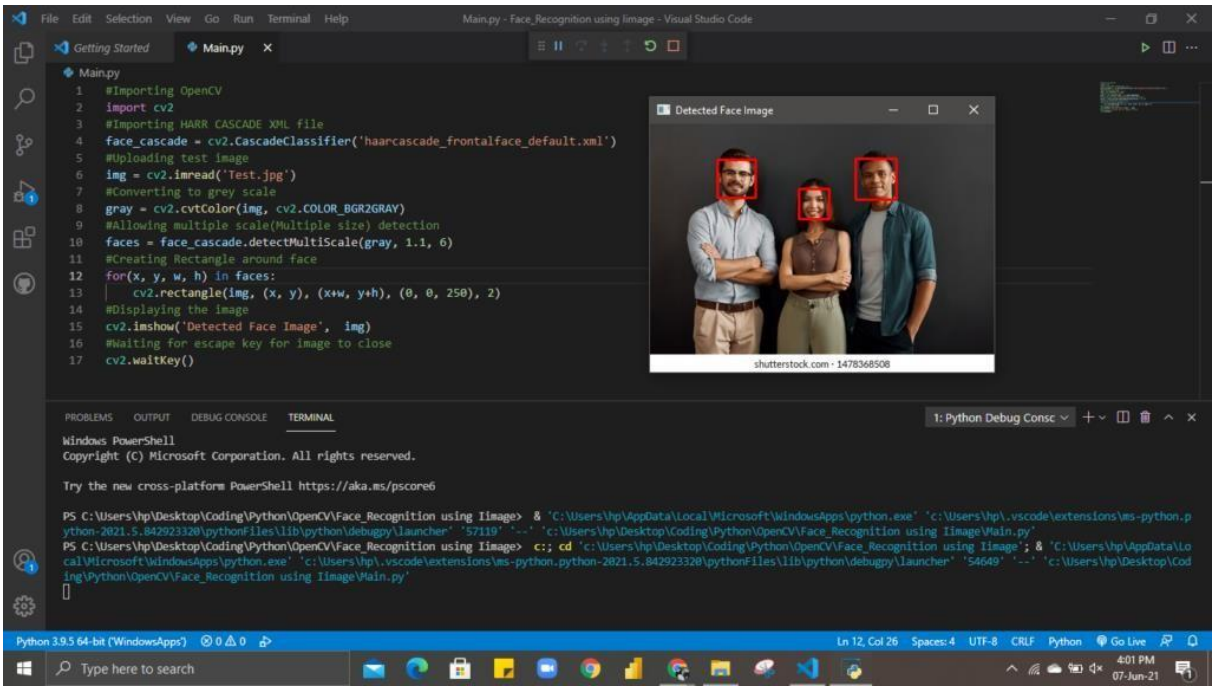


Fig 5.2.1 Basic face detection using Haar Cascade Algorithm.

STEP 1: Importing OpenCV

In the command prompt type (pip install OpenCV-python) using this command, we can install and set up OpenCV to python. OpenCV provides a real-time optimized Computer Vision library, tools, and hardware and the same will be used in our project.

STEP 2 – Importing XML file

You can download the file using this link and save it in the project folder – https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade.

STEP 3 – Importing the test image.

imread() is a method of OpenCV to read the input. Similarly, imshow() is a method to display the processed input in the form of output.

STEP 4 – Converting to grey Scale.

This project works on images that are in greyscale and hence we convert the image to greyscale for ease of face detection.

STEP 5 – Detecting Multi-scale faces.

This function allows to detect objects of different sizes in the input image and hence an image with multi people with different sizes of the face can also be detected. Parameters of the functions are – detectMultiScale (InputArray image, double scaleFactor=1.1, int minNeighbors=6).

STEP 6 – Mentioning sides of the rectangle for face detection.

This function helps us to mention the dimensions thickness and color of the rectangle that will be visible during the face detection.cv2.rectangle(image, start_point, end_point, color, thickness)

STEP 7 – Displaying the detected image.

Yay! You have come so far. Now it's time to display the image that has been detected. We also add the feature to close the image tab only when a key is pressed.

6. SYSTEM IMPLEMENTATION

6.1 MODULE DESIGN SPECIFICATION

The system is made up of four main parts:

- Video Streaming
- Preprocessing
- Algorithm Implementation
- Database

6.1.1 Video Streaming

Video streaming technology is one way to deliver video over the Internet. Using streaming technologies, the delivery of audio and video over the Internet can reach many millions of customer using their personal computers, PDAs, mobile smart phones or other streaming devices.

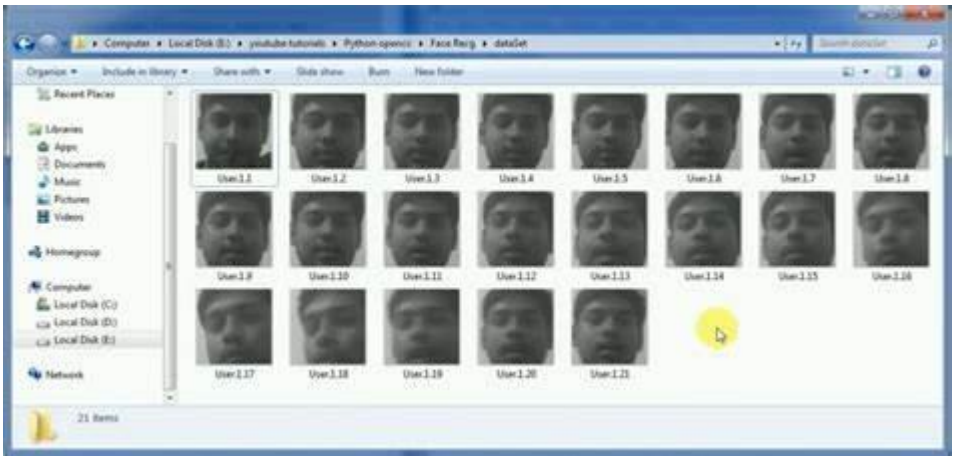


Fig 6.1.1 Video Streaming

6.1.2 Preprocessing

Pre-processing is a common name for operations with images at the lowest level of abstraction -- both input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. In this step we have to reduce the complexity of the picture of license plate using resize and the conversion. Using these pre-process we can change the size of the video using resize.

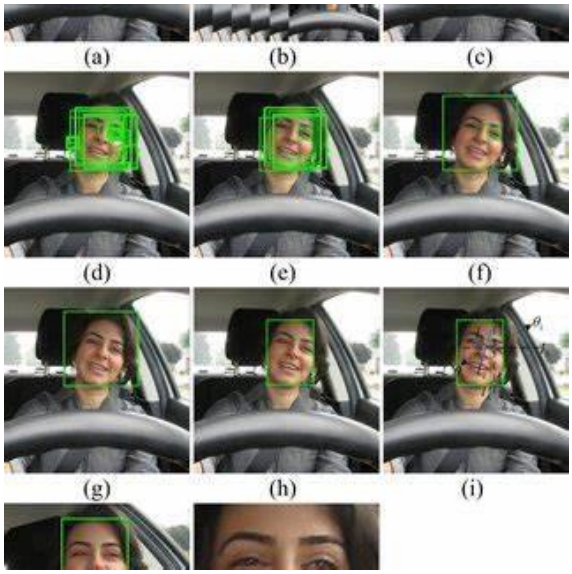


Fig 6.1.2 Preprocessing

6.1.3 Algorithm Implementation

It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001.

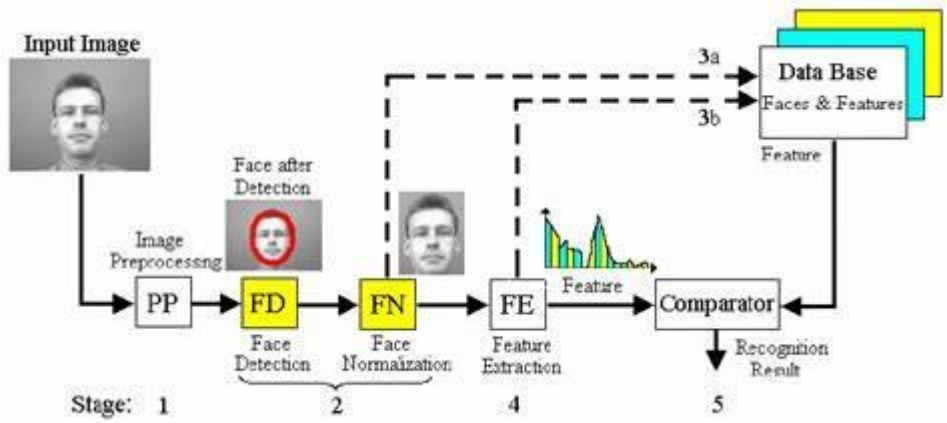


Fig 6.1.3 Algorithm Implementation

The first step in face recognition is to detect faces in an image or video stream. This involves algorithms that can identify areas of an image that contain faces. Popular algorithms for face detection include Viola-Jones algorithm, Histogram of Oriented Gradients (HOG), and Convolutional Neural Networks (CNN).

Next, the system needs to extract features from the aligned face images. This involves identifying unique characteristics of the face that can be used for identification. Popular feature extraction algorithms include Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), and Principal Component Analysis (PCA).

Finally, the system can classify the face into a known identity or verify whether the face belongs to a known identity. This involves comparing the extracted features to the features in the database and calculating a similarity score. Algorithms for classification and verification include Support Vector Machines (SVM), Neural Networks, and K-Nearest Neighbors (KNN).

A face recognition system can also incorporate continual learning to update its database and improve its accuracy over time. This involves algorithms that can learn from new face data and adapt to new environments.

6.1.4 Database

Database is a collection of digitized images. It is maintained primarily to support research in image processing, image analysis, and machine vision. For the data base creation we should take images of particular person like mukesh. The apply theimages of that person with the help of face recognition package using combination of libray.

face_recognition.api.batch_face_locations(images,number_of_times_to_upsample=1, batch_size=128)

Returns an 2d array of bounding boxes of human faces in a image using the cnn face detector If you are using a GPU, this can give you much faster results since the GPU can process batches of images at once. If you aren’t using a GPU, you don’t need this function.

Parameters:	• **Images** – A list of images (each as a numpy array) • **Number_of_times_to_up sample** – How many times to up sample the image looking for faces. Higher numbers find smaller faces. • **Batch_size** – How many images to include in each GPU processing batch.
Returns:	A list of tuples of found face locations in css (top, right, bottom, left) order

If you are into any sort of image processing, computer vision or machine learning, chances are high that you might have come across/used dlib somewhere in your journey.According to dlib's github page, dlib is a toolkit for making real world machine learning and data analysis applications in C++. While the library is originally written in C++, it has good, easy to use Python bindings.

Fig 6.1.4 Database

It is important to note that the accuracy of face recognition systems can vary depending on various factors, such as the quality of the images, lighting conditions, and the algorithms used. Therefore, it is crucial to carefully consider the use case and potential limitations of face recognition technology before implementation.

7. SYSTEM TESTING

7.1 WHITE BOX TESTING

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which also known as glass box is testing, structural testing, clear box testing, open box testing and transparent box testing. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

The white box testing contains various tests, which are as follows:

- Path testing
- Loop testing
- Condition testing
- Testing based on the memory perspective
- Test performance of the program

7.2 BLACK BOX TESTING

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer. In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

The test procedure of black box testing is a kind of process in which the tester has specific knowledge about the software's work, and it develops test cases to check the accuracy of the software's functionality.

It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function.A tester knows about the definite output of a particular input, but not about how the result is arising. There are various techniques used in black box testing for testing like decision table technique, boundary value analysis technique, state transition, All-pair testing, cause-effect graph technique, equivalence partitioning technique, error guessing technique, use case technique and user story technique.

7.3 TEST CASES

Test Report **:**01

Product **:** Creating Datasets for User Registration

Use Case **:** Upload Images

Test Case Id	Test Case/Action to be Performed	Expected Result	Actual Result	Pass/Fail
1	Upload the video as an input	Uploaded	As Expected	Pass

Table 7.3.1 Test Case for Creating Datasets.

Test Report **:** 02

Product **:** Detecting the User with Existing Datasets

Use Case **:** Search and Detect the User Who Registered

Test Case Id	Test Case/Action to be Performed	Expected Result	Actual Result	Pass/Fail
1	Search the appropriate type of user from the recorded datasets.	Searched Successfully	As Expected	Pass
2	Intimate the user if an unknown person is usinghis verification	Intimated Successfully	As Expected	Pass

Table-7.3.2 Test Case for Search and Detect the user with existing Dataset

Test Report : 03

Product : Send Email to the User to Verify the Identity

Use Case : Search and Detect the Valid User Identity

Test Case Id	**Test Case/ Action to be Performed**	**Expected Result**	**Actual Result**	**Pass/Fail**
1	Search the user folder and check whether that user exist in the file system.	Searched Successfully	Searched Successfully	Pass
2	Sending OTP to the user who is currently trying to log in to the ATM account.	Verified successfully	Verified Successfully	Pass
3	Show the detected number plates from the recorded videoframe by frame.	Detected Successfully	Detected Successfully	Pass

Table-7.3.3 Test Case for Search and Detect the Valid User.

Test Report **:** 04

Product **:** Uploading OTP to Verify Identity

Use Case **:** Detect OTP of the User for the Account Accessibility

Test Case Id	**Test Case/ Action to be Performed**	**Expected Result**	**Actual Result**	**Pass/Fail**
1	Verify the user by validating the photo of the user who is currently trying to access.	Sent successfully	Sent Successfully	Pass
2	Validating the OTP of the user via the email and if not then doesn't grant the permission	Validated successfully	Validated Successfully	Pass

Table-7.3.4 Test Case to detect the OTP of the user for the account accessbility

8. CONCLUSION AND FUTURE ENHANCEMENTS

8.1 CONCLUSION

The use of face recognition technology in smart ATM systems can also increase the efficiency of transactions, as it eliminates the need for physical ATM cards. This can save time for customers and reduce the risk of card skimming or cloning. Furthermore, facial recognition technology can be beneficial for customers who have trouble remembering their PINs or have difficulty insert in grand withdrawing cards.

However, it is important to note that the use of facial recognition technology raises concerns about privacy, accuracy, and bias. Therefore, it is crucial to implement privacy protection measures and ensure that the technology is accurate and unbiased. Furthermore, banks and financial institutions must be transparent about the use of facial recognition technology and communicate their policies clearly to customers. For example, the use of artificial intelligence (AI) can enable ATMs to provide personalized recommendations to customers based on their transaction history, location, and preferences. This can enhance customer engagement and provide a more tailored banking experience. Smart ATMs can also incorporate voice assistant chatbots to provide customer support and assistance, making banking more accessible and convenient for customers.

Overall, the future of smart ATM security systems looks promising with the use of advanced face recognition technology. With proper implementation and measures in place, facial recognition can provide a secure, convenient, and efficient banking experience for customers while mitigating risks associated with ATM-related fraud.

8.2 FUTURE ENHANCEMENTS

Facial recognition technique seems more challenging as compared to other biometrics, thus more efficient algorithm can be developed. The flaws in face recognition technique like the inability to detect face when beard, aging, glasses and caps can be rectified and eliminated or reduced. If the cost of retina or iris recognition reduces, it can be used instead of face recognition. This can also be are placement of all the smart cards which can easily been lost since our face can be our identity it will be useful and a super-fast to verify the user identity. Advanced 3D face recognition technology can be used to improve the accuracy and reliability of face recognition systems, making it more difficult for fraudstersto bypass the security system using fake images. The privacy of users can be protected by ensuring that the

facial images captured by the ATM are securely stored and not used for any other purpose than authentication. Additionally, privacy policies should be clearly communicated to users, and their consent should be obtained before using facial recognition technology. Facial recognition analytics can be used to detect and prevent fraud by analyzing facial features and expressions of the user during the transaction

APPENDICES

A.1 CODING

Face_rec.py

```
import cv2, sys, numpy, os
import urllib
import numpy as np
import time
import os
from subprocess import call
import time
import os
import glob
import smtplib
import base64
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import sys
import random
#DO THE CHANGES HERE
gmail_user = "balarengasamy12102001@outlook.com"
gmail_pwd = "bala1210"
FROM = 'balarengasamy12102001@outlook.com'
TO = ['balarengasamy12102001@outlook.com'] #must be a list
otp_=random.randint(10000,100000)
pins=1234
def mail():

    msg = MIMEMultipart()
    time.sleep(1)
    msg['Subject'] ="SECURITY"

    #BODY with 2 argument
```

```
#body=sys.argv[1]+sys.argv[2]
#DO THE CHANGES HERE
body="Authenthication..... your otp for logging in :"+str(otp_)

#otp_text="your otp for logging in :"+str(otp_)
msg.attach(MIMEText(body,'plain'))
#msg.attach(MIMEText(otp_text,'plain'))
time.sleep(1)



###IMAGE
fp = open("1.jpg", 'rb')
time.sleep(1)
img = MIMEImage(fp.read())
time.sleep(1)
fp.close()
time.sleep(1)
msg.attach(img)
time.sleep(1)


try:
    server = smtplib.SMTP("smtp.office365.com", 587) #or port 465 doesn't seem to work!
    print ("smtp.gmail")
    server.ehlo()
    print ("ehlo")
    server.starttls()
    print ("starttls")
    server.login(gmail_user, gmail_pwd)
    print ("reading mail & password")
    server.sendmail(FROM, TO, msg.as_string())
    print ("from")
    server.close()
    print ('successfully sent the mail')
```

```
    except:
        print ("failed to send mail")

size = 4
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'
n=input("enter your name : ")
f =open("users/"+n+".txt","r")
a = f.read()
a =a.split(" ")
pins =int(a[0])
TO[0] = a[1]
print('Training...')
# Create a list of images and a list of corresponding names
(images, labels, names, id) = ([], [], {}, 0)
for (subdirs, dirs, files) in os.walk(datasets):
    for subdir in dirs:
        names[id] = subdir
        subjectpath = os.path.join(datasets, subdir)
        for filename in os.listdir(subjectpath):
            path = subjectpath + '/' + filename
            label = id
            images.append(cv2.imread(path, 0))
            labels.append(int(label))
        id += 1
(width, height) = (130, 100)

# Create a Numpy array from the two lists above
(images, labels) = [numpy.array(lis) for lis in [images, labels]]

# OpenCV trains a model from the images
# NOTE FOR OpenCV2: remove '.face'
model = cv2.face.FisherFaceRecognizer_create()
model.train(images, labels)
```

```
# Part 2: Use fisherRecognizer on camera stream
face_cascade = cv2.CascadeClassifier(haar_file)
##with open("1.txt", mode='a') as file:
webcam = cv2.VideoCapture(0)

##url="http://192.168.43.1:8080/shot.jpg"
while True:

   (_, im) = webcam.read()
##   imgPath=urllib.urlopen(url)
##    imgNp=np.array(bytearray(imgPath.read()),dtype=np.uint8)
##    im=cv2.imdecode(imgNp,-1)
   gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
   faces = face_cascade.detectMultiScale(gray, 1.3, 5)
   for (x,y,w,h) in faces:
      cv2.rectangle(im,(x,y),(x+w,y+h),(255,255,0),2)
      face = gray[y:y + h, x:x + w]
      face_resize = cv2.resize(face, (width, height))
      #Try to recognize the face
      prediction = model.predict(face_resize)
      cv2.rectangle(im, (x, y), (x + w, y+ h), (0, 255, 0), 3)

      if prediction[1]<500:
         #port.write('B')
        # print (names[prediction[0]])
         cv2.putText(im,names[prediction[0]],(x-10, y-10),
cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
         print('The accessing person is ',str(n))

         if names[prediction[0]]==n:

            print("The detected face person is : ",names[prediction[0]])
            print('you can proceed your transaction')
```

```
            pin=int(input('enter your pin: '))
            if pin ==pins:
                print("You can continue further")
                exit()



            else:
                mail()
                check_otp=int(input("Enter the otp :"))
                if check_otp==otp_:
                    print("You can continue further")
                    exit()
                else :
                    print("Incorrect pin ... exiting the portal")



        else:
            im2=im
            cv2.putText(im2,'unknown ',(x-10, y-10), cv2.FONT_HERSHEY_PLAIN,1,(0, 0,
255))
            print("The detected person is unknown ")
            cv2.imwrite('1.jpg',im2)
            mail()
            check_otp=int(input("enter the otp : "))
            if check_otp==otp_:
                pin=int(input('enter your pin:'))
                if pin == pins:
                    print("You can continue further")
                    exit()
                else:

                    print("Incorrect pin ... exiting the portal")
                    exit()
```

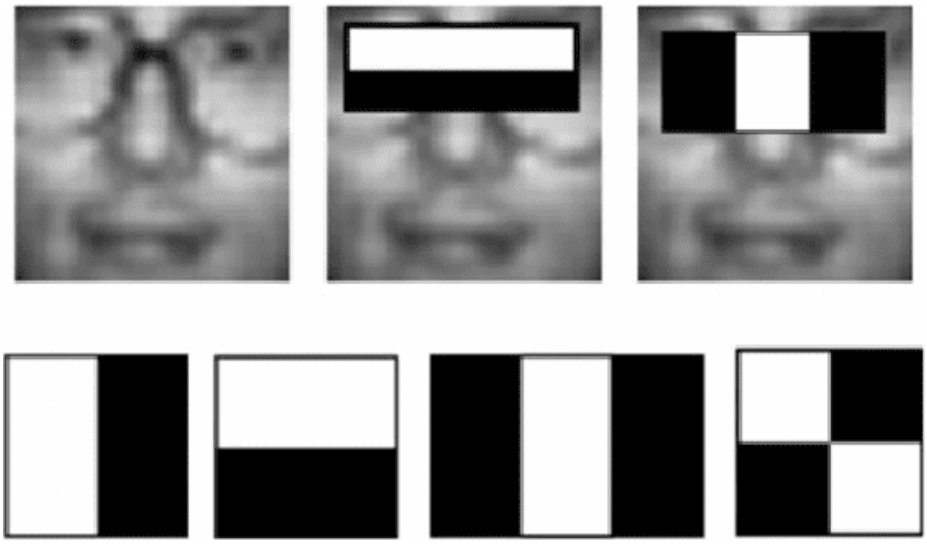
```
        else:
            cv2.putText(im,'Scanning',(x-10, y-10), cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
    cv2.imshow('OpenCV', im)
    key = cv2.waitKey(10)
```

Create.py

```
#creating database
import cv2, sys, numpy, os
import urllib.request
import numpy as np
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'  #All the faces data will be present this folder
sub_data = input("enter the name of the person :")
password = input("enter the password:")
outlook = input("enter the outlook account:")
users = 'users'
file = 'users/'+sub_data+'.txt'
####sub_data = 'hai'  #These are sub data sets of folder, for my faces I've used my name

path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
    os.mkdir(path)
    open(file,"x")
    f = open(file,"w")
    f.write(password+' ')
    f.write(outlook)
    f.close()
(width, height) = (130, 100)    # defining the size of images


face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0) #'0' is use for my webcam, if you've any other camera attached
use '1' like this
```

```
##url="http://192.168.43.1:8080/shot.jpg"
# The program loops until it has 30 images
ofthe face.count = 1
while count < 101:
   (_, im) = webcam.read()
##    imgPath=urllib.urlopen(url)
##
       imgNp=np.array(bytearray(imgPath.read()),dtype
=np.uint8)##       im=cv2.imdecode(imgNp,-1)
   gray = cv2.cvtColor(im,
   cv2.COLOR_BGR2GRAY)faces =
   face_cascade.detectMultiScale(gray, 1.3,
   4) for (x,y,w,h) in faces:
      cv2.rectangle(im,(x,y),(x+w,y+h),(2
      55,0,0),2)face = gray[y:y + h, x:x +
      w]
      face_resize = cv2.resize(face, (width, height))
      cv2.imwrite('%s/%s.png' % (path,count),
      face_resize)
   count += 1

   cv2.imshow('Op
   enCV', im)key =
   cv2.waitKey(10)
   if key == 27:
      break
```

A.2 SAMPLE SCREENS



FigA2.1 Face Detection

In the fig A2.1 First introduced in 2001, Haar cascades are a class of object detection algorithms

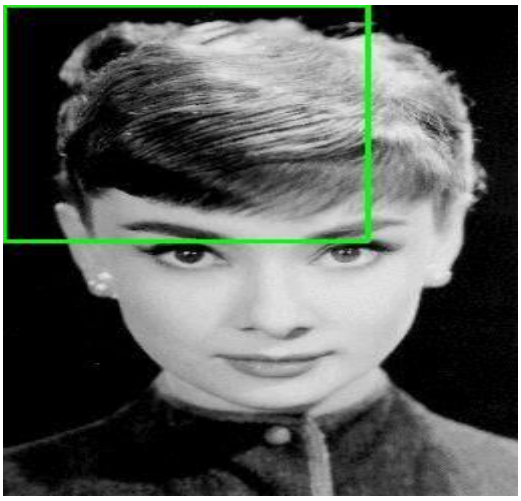


Fig A2.2 Sliding Image

In the fig A2.2 An example of a sliding window, moving from left-to-right and top-to-bottom, to locate the face in the image

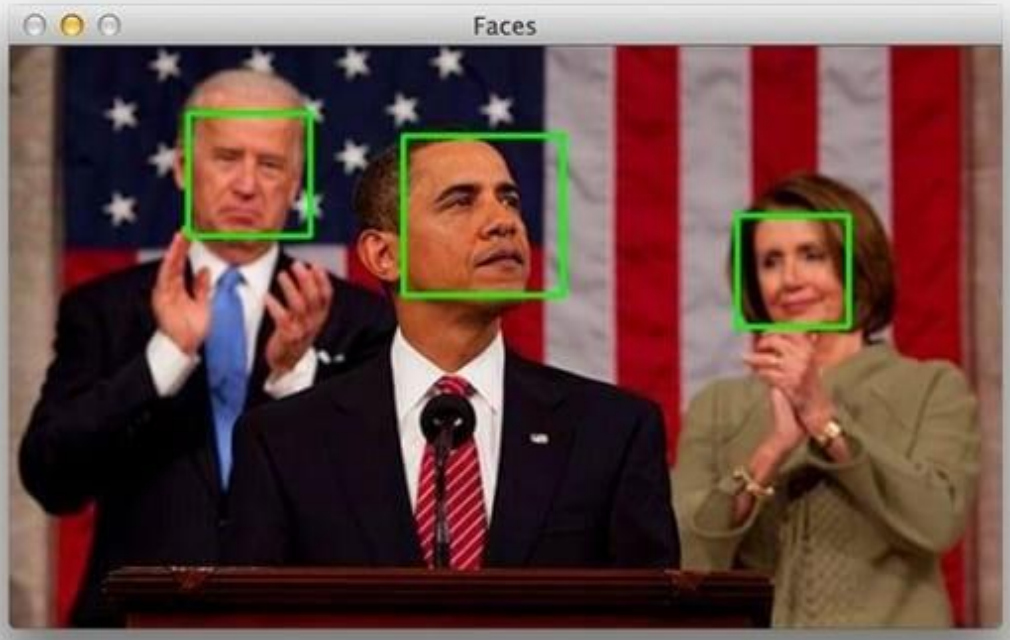


Fig A2.3 Detecting Face

Fig A2.3 Shows that in the frame 3, Detecting faces in images using OpenCV's pre-trained Haar cascades.

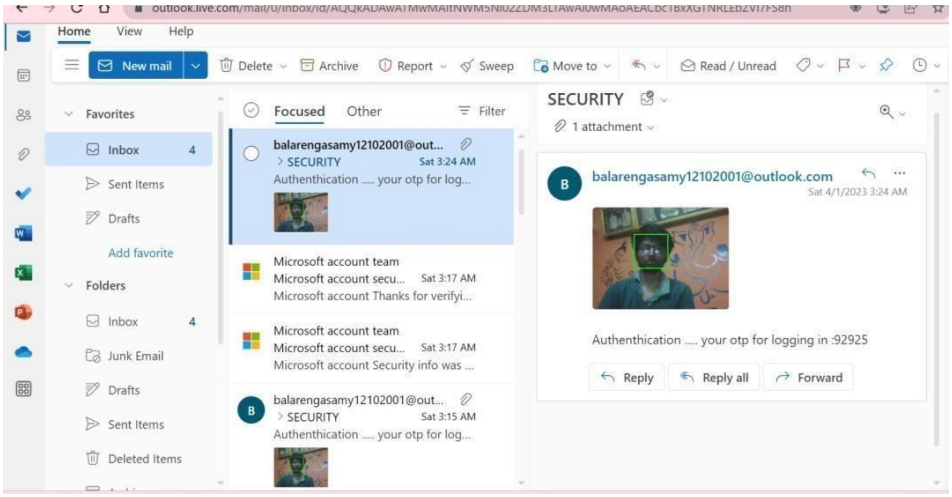


Fig A2.4 Notification to Mail

Fig A2.4 Verify the identity of user,if identity doesn't exist it will notify your outlook account by sending a mail with image of that user.

REFERENCES

[1] A. M. Iyabode, Y. N. Nureni, A. F. Adebayo, O. A. Olamide, "Cardless-Electronic Automated Teller Machine(EATM) with Biometric Authentication," International Journal of Engineering Trends and Technology (IJETT), vol. 30, no. 1, 2015

[2] Mike Flacy. "SMART ATM USES QR CODES INSTEAD OF CARDS TO DISPENSE CASH", Internet: http://www.digitaltrends.com/cool-tech/smart-atm-uses-qr-codesinstead-of-cards-to-dispense-cash/, Jun. 16, 2012 [Jul. 05, 2018]

[3] T. Maqua, R. Neff, M. Wbbeling, Improve ATM Withdrawal Security and Usability with your Smartphone

[4] ATM Market Place. "ATM cardless cash access: Why the QR code matters (alot) to FIs," Internet: http://www.atmmarketplace.com/articles/atm-cardless-cash-accesswhy-the-qr-code-matters-a-lot-to-fis/, Nov. 4, 2013 [Jul. 05, 2018]

[5] Victoria Woollaston. "Next-generation cash machines set to replace bank cards with facial recognition", Internet: http://www.dailymail.co.uk/sciencetech/article-2365166/Nextgeneration-cash-machines-set-replace-bank-cards-facialrecognition.html, Jul. 16, 2013 [Jul. 05, 2018]

[6] Odusina, A. Olumide, Automated Teller Machine Usage And Customers' Satisfaction In Nigeria Elite Research Journal of Accounting and Business Management Vol. 2(3) pp. 43 - 47, July, 2014. Available online http://www.eliteresearchjournals.org/erjabm/index.html.

[7] Md. Mosabber Hossain, "Understanding of ATM (Automated Teller Machine) in Bangladesh", B.Sc Thesis BRAC university, Bangladesh, 2006

[8] A. B. Garko, Enhancing “The Current Automated Teller Machine (ATM) In Nigerian Banking Sector”. JORIND vol 2, pp 59-64 December, 2011.

[9] R. Simutis, D. Dilijonas, L. Bastina, J. Friman, & P. Drobinov, “The optimization of Cash Management for ATM network. Information Technology and Control”, Vol.36, No.1A, January 2015.

[10] O. A. Fabumni, “Appraisal of the Use Automated Teller Machine in the banking industry in Nigeria”. HTTP: https://www.unilorin.edu.ng/studproj/cis/0730gc071.pdf. Retrived july 25th , 2015.

[11] "Smart ATM Security using Fingerprint Biometric Authentication and OTP Verification" by S. S. Sagar, S. M. Jadhav, and S. S. Suryawanshi. This paper presents a system for secure ATM transactions using fingerprint biometric authentication and OTP verification.

[12] "Design and Implementation of a Smart ATM Security System using Multi-Factor Authentication" by S. A. Al-Jarrah and M. Al-Mazraawi. This paper proposes a smart ATM security system that uses multi-factor authentication, including fingerprint and password authentication.

[13] "A Novel Approach for Smart ATM Security using Face Recognition" by S. N. Mhatre and P. N. Pimple. This paper describes a smart ATM security system that uses face recognition for authentication.

[14] "Smart ATM System for Enhanced Security using IoT and Biometrics" by P. Ramakrishnan and S. K. Karthikeyan. This paper proposes a smart ATM system that uses IoT and biometric authentication to enhance security.

[15] "Smart ATM Security System using Raspberry Pi" by M. R. Zaman and M. N. Islam. This paper presents a smart ATM security system that uses Raspberry Pi as the main controller.