BANK TRANSACTION USING FACIAL IDENTIFICATION SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

ALISHETTY ACHYUTHANANDA (Reg. No. 3711030)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

MARCH 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119
www.sathyabamauniversity.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of ALISHETTY ACHYUTHANANDA (Reg. No. 3711030**)** who carried out the project entitled **"BANK TRANSACTION USING FACIAL IDENTIFICATION SYSTEM"** under our supervision from Dec 2020 to Mar 2021.

Internal Guide

Mr.MURARI D KAMALESH.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I, ALISHETTY ACHYUTHANANDA (Reg. No. 3711030**)** hereby declare that the Professional Training Report on "**BANK TRANSACTION USING FACIAL IDENTIFICATION SYSTEM"** done under the guidance of **Mr.MURARI D KAMALESH.,** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, School of Computing and Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr. L. LAKSHMANAN, M.E., Ph.D., Heads of the Department, Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.MURARI D KAMALESH.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways For the Completion of the project.

ABSTRACT

The objective of this project is to develop a robust automated algorithm for indoor room identification in higher level security purpose with high recognition rates in varying environment. Facial Recognition software has a liveness detection which prevents hackers from using a picture of the customer for impersonation purposes. It also applies to other biometric modalities such as fingerprints where the liveness detection does exactly that – it assesses the 'liveness' of the facial image as it is known. The recognition system also allows customers to access their bank accounts from computers. Facial recognition is one of numerous ways banks can decrease friction in their customers' experience and increase efficiency and accessibility. First, Haar Cascade based algorithm has been applied for fast and simple face detection from the input image. The face image is then being converted into grayscale image. After that, the iris candidates are extracted from the intensity valleys from the detected face. Costs of each iris candidates are calculated. Finally, the iris candidates are paired up and the cost of each possible pairing is computed by a combination of mathematical models.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DFD	-	DATA FLOW DIAGRAM
SQL	-	STRUCTED QUERY LANGUAGE
SET	-	SECURE ELECTRONIC TRANACTION
OTP	-	ONR TIME PASSWORD
UML	-	UNIFIED MODELLING LANGUAGE

CHAPTER 1

INTRODUCTION

Over the last decade, we have seen an increase in the use of technology in many business sectors to simplify and better engage customers. This is especially true in the banking and finance sector. Since the start of the digital revolution facial recognition has been gaining prominence over touch and type-based interactions due to the convenience it offers without compromising on the security of transactions. Despite an increase in the use of EMV cards (Europay, MasterCard, Visa) coupled with password creation policies, there has been a surge in banking fraud cases. As a result of the billions that are lost by major banking institutions, there has been a call to switch to biometric facial recognition to curb this issue. It means that banking software will rely on face scans which it then compares with similar ones that were uploaded by the bank's personnel into their system so as to verify the customer's identity. The aim is to authenticate the identity and only allow a transaction to go through if the account owner's identity is positively identified. This customer ID authentication process is known as KYC (Know Your Customer).

MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine Learning methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms:

It can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Unsupervised machine learning algorithms

It is used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

Semi-supervised machine learning algorithms

It falls somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn't require additional resources.

Reinforcement machine learning algorithms

It is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

PROBLEM DEFINITION

The biometric facial recognition software with banking software is more conventional methods. In fact, using passwords comes with a rather serious caveat. People create passwords based on what they know. So, it is easy for a hacker to employ a number of tactics to crack the password. Another major flaw is that people can have too many passwords eg. for social media accounts, emails, and e-wallets as well. Also, creating a complex password can make it easier to forget and when a banking customer requests for a temporary code via email to reset it, then a hacker can intercept the inbox. Using facial recognition means that the banking customer has only one face which can allow them access to all their bank accounts.

PROJECT DESCRIPTION

Facial recognition is one of numerous ways banks can decrease friction in their customers' experience and increase efficiency and accessibility.This project make Identity Verification and Account Withdrawals Allowing customers to make withdrawals from their bank accounts.The biometric facial-recognition software helps minimize fraud where online hackers unlawfully use passwords and other data to steal from banking institutions. The software verifies a person's identity before processing any transaction. Our goal is to provide an extremely frictionless, personalized experience with a focus on security.

CHAPTER 2

LITERATURE SURVEY

1. Facial-Recognition Payment: An Example of Chinese Consumers

Authors: Wen Kun Zhang ; Min Jung Kang, IEEE Access, Year: 2019

The emergence and use of facial-recognition payment technology has brought new challenges. Although credit-card payment is quick and easy, it is easy to lose a card or forget the password. Because people use simple passwords and reuse them on different accounts and services, passwords can be shared and cracked. QR payment is inseparable from smart phones, smart phones may be lost, signals may be unstable, and batteries may be exhausted. However, facial-recognition technology, which detects and describes feature vectors without physical contact, directly contributes to overall efficiency, performance, and accuracy. Currently, studies of technical issues of facial-recognition technology and facial-recognition payment systems are very popular. There are many studies that emphasize the working principle of the facial-recognition system, the system's reliability, and the future development trend. However, for non-technical issues, such as from the perspective of consumers, research on the characteristics of facial-recognition payment and the factors affecting consumer's intent to use is rare. Therefore, the purpose of this study is to explore the factors influencing consumers' willingness to use facial-recognition payment systems. This study has selected security, visibility, and expected effort and social image as the feature variables of the facial-recognition payment system. Results in this paper shows that the safety, security, visibility and social image will affect consumers' intent to use the system. It can also influence consumers' intent to use through perceived usefulness. The amount of effort expected not only has direct influence on intent to use but also influences the intent to use by the mediating factor of perceived usefulness. In this article, Openness characteristic (consumer's personality) has a moderating effect on the relationship between security, expected effort and usage intention.

2. Secure multifactor authentication payment system using NFC
Authors: Anirudhan Adukkathayar ; Gokul S Krishnan ; Rajashree Chinchole, 2015 10th International Conference on Computer Science & Education (ICCSE)

The latest trend of making financial transactions is done by the use of cards or internet banking. A person may have multiple bank accounts across several banks which makes it difficult for him/her to manage the transactions i.e. he/she either has to carry several cards or use a bunch of bank websites for accomplishing his/her transaction purposes. This situation demands the need of a simple, secure and hi-tech system for achieving the purposes of making transactions. We propose such a system that uses the latest technologies like NFC and multifactor authentication which can be used on any NFC enabled Smartphone. The multi factor authentication system uses a 4-digit PIN as the knowledge factor, an NFC enabled Smartphone, instead of cards, as the possession factor and the face of the user as the inherence factor for the purpose of authentication. The proposed system which can be implemented as cross-platform mobile application, not only allows the user to make secure transactions, but also allows him/her to make transactions from his/her multiple accounts.

3. Biometric Face Recognition Payment System
Authors: Surekha. R. Gondkar Saurab. Dr. C. S. Mala International Journal of Engineering Research & Technology NCESC - 2018 Conference Proceedings

Use of payment cards in various places such as shopping, restaurants, lodges and online payment for booking hotels, movie tickets, flight and train tickets etc are increasing day by day. So the problem is that a person has to carry payment cards along with him and keep the cards secure to use it all the time. This also lacked security. In the present work the biometric face recognition payments is used in all kinds of payments. Thus it avoids the need to memorize different passwords. Face recognition payment system is safe, secure and even easy to use. It is reliable and more efficient compared to other payment technologies. A general design of online payment system using face recognition is proposed. The methods adopted for face recognition are by finding the eigenfaces and Euclidean distance.

4. Facial Recognition in Banking – Current Applications
Author:Niccolo Mejia,2019 Conference Proceedings

Facial recognition software is making its way into the mainstream, with consumer applications such as the ability to unlock one's smartphone with their face. The banking sector has been at the forefront of enterprise adoption of AI since machine learning became the hot topic of the business world in the early years of the decade; as such, it makes sense that facial recognition technology would start to make its way into banking.There are a handful of companies offering facial recognition software to banks that at face value seem to have the requisite talent in their C-suite that we look for when vetting a company on their claims to leveraging AI. These companies offer software with applications ranging from physical security to the ability for customers to make withdrawals with their faces.Facial recognition is one of numerous ways banks can decrease friction in their customers' experience and increase efficiency and accessibility. Some experts think that this is how banks can succeed in the future as AI and other technologies make more and more services accessible without any down time.

5. "Face Detection and Recognition for Bank Transaction ", International Journal of Emerging Technologies and Innovative Research
Authors Sudarshan Dumbre, Shamita Kulkarni, Devashree Deshpande, P.V.Mulmule Journal of Emerging Technologies and Innovative Research 2018

There is a crucial need for improving security in banking region. With the birth of the Automatic Teller Machines, banking became a lot easier though with its own troubles of insecurity. Due to tremendous increase in the number of criminals and their activities, the ATM has become insecure. ATM systems today use no more than an access card and PIN for identity verification. An attempt is made for developing a system that integrates facial recognition technology into the identity verification process and use of RFID card for handling multiple accounts in same card with Raspberry pi controller. The development of such a system would serve to protect consumers and financial institutions alike from intruders and identity thieves. This paper proposes an automatic teller machine security model that would combine a RFID card, a PIN, and electronic

facial recognition that will go as far as with holding the fraudsters' card. If this technology becomes widely used, faces would be protected as well as PINs. However, it obvious that manes biometric features cannot be replicated, this proposal will go a long way to solve the problem of Account safety making it possible for the actual account owner alone have access to his accounts. The combined biometric features approach is to serve the purpose both the identification and authentication.

6. Continuous User Identity Verification Using Biometric Traits for Secure Internet Services

Authors: 1Dr.SHAIK ADBUL MUZZER, 2GOSALA SUBHASIN

Nowadays, it becomes serious concern to provide more security to web services. So, secure user authentication is the fundamental task in security systems. Traditionally, most of the systems are based on pairs of username and password which verifies the identity of user only at login phase. Once the user is identified with username and password, no checks are performed further during working sessions. But emerging biometric solutions substitutes the username and password with biometric data of user. In such approach still single shot verification is less efficient because the identity of user is permanent during whole session. Hence, a basic solution is to use very short period of timeouts for each session and periodically request the user to input his credentials over and over. But this is not a proper solution because it heavily affects the service usability and ultimately the satisfaction of users. This paper explores the system for continuous authentication of user using his credentials such as biometric traits. The use of continuous biometric authentication system acquires credentials without explicitly notifying the user or requiring user interaction that is, transparently which is necessary to guarantee better performance and service usability.

CHAPTER 3
AIM AND SCOPE OF THE PROJECT

OBJECTIVE

Facial Recognition software has a liveness detection which prevents hackers from using a picture of the customer for impersonation purposes. It also applies to other biometric modalities such as fingerprints where the liveness detection does exactly that – it assesses the 'liveness' of the facial image as it is known. The recognition system also allows customers to access their bank accounts from computers. Facial recognition is one of numerous ways banks can decrease friction in their customers' experience and increase efficiency and accessibility.

EXISTING SYSTEM

In previous days they used only single level authentication like OTP generation. It was not more secured. Secure electronic transaction (SET) It involves many levels of encryption, using many combinations of symmetric, cryptography, asymmetric cryptography and hashing. It does not assume that each agent has his own private key so that the only problem which is remained is the distribution of the public keys, but allows cardholders to decide their asymmetric key.

Disadvantages

1. User must have credit card
2. It is not cost-effective when the payment is small
3. None of anonymity and it is traceable
4. Network effect - need to install client software (an e-wallet).
5. Cost and complexity for merchants to offer support, contrasted with the comparatively low cost and simplicity of the existing SSL based alternative.
6. Client-side certificate distribution logistics.

PROPOSED SYSTEM

This uses machine learning techniques to get a high degree of accuracy from what is called “training data”. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.

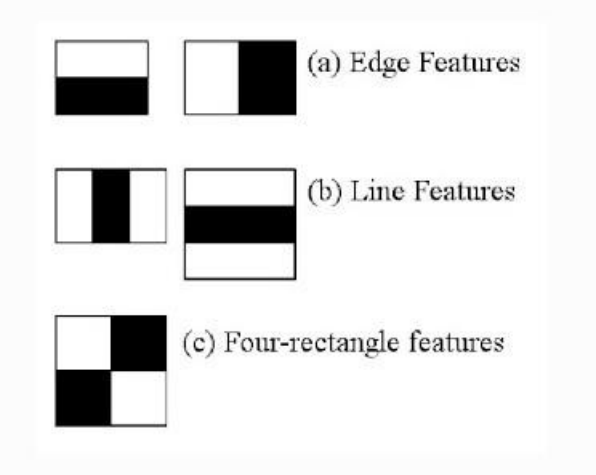

Fig 3.1 Figure showing the features

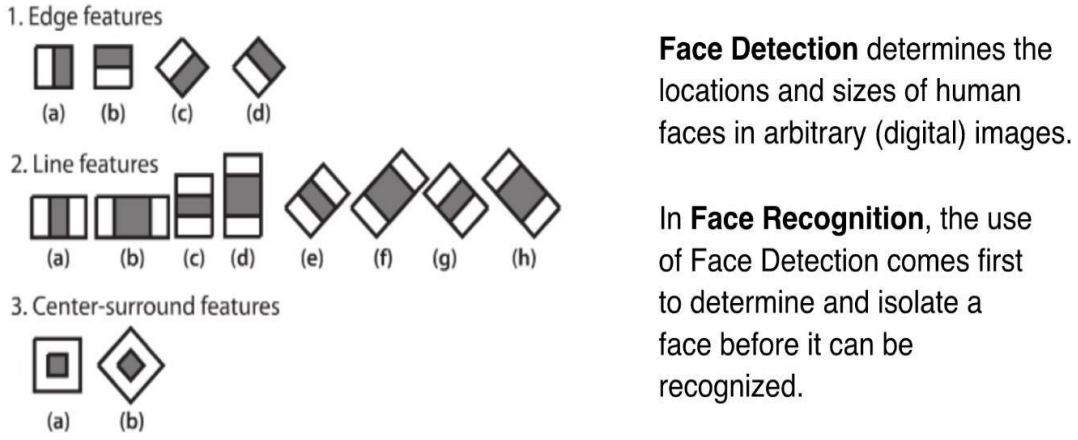

Fig 3.2 Haar Cascades algorithm

The cascade classifier consists of a collection of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called decision stumps. Each stage is trained using a technique called boosting. Boosting provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. Positive indicates that an object was found and negative indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

The stages are designed to reject negative samples as fast as possible. The assumption is that the vast majority of windows do not contain the object of interest. Conversely, true positives are rare and worth taking the time to verify.

- A true positive occurs when a positive sample is correctly classified.
- A false positive occurs when a negative sample is mistakenly classified as positive.
- A false negative occurs when a positive sample is mistakenly classified as negative.

To work well, each stage in the cascade must have a low false negative rate. If a stage incorrectly labels an object as negative, the classification stops, and you cannot correct the mistake. However, each stage can have a high false positive rate. Even if the detector incorrectly labels a nonobject as positive, you can correct the mistake in subsequent stages. Adding more stages reduces the overall false positive rate, but it also reduces the overall true positive rate.

Cascade classifier training requires a set of positive samples and a set of negative images. You must provide a set of positive images with regions of interest specified to be used as positive samples. You can use the Image to label objects of interest with bounding boxes. The Image Labeler outputs a table to use for positive samples. You also must provide a set of negative images from

which the function generates negative samples automatically. To achieve acceptable detector accuracy, set the number of stages, feature type, and other function parameters.

Advantage

- The key advantage of a Haar-like feature over most other features is its calculation speed.
- Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video
- Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

1. Economic Feasibility
2. Technical Feasibility
3. Social Feasibility

SYSTEM ARCITECTURE

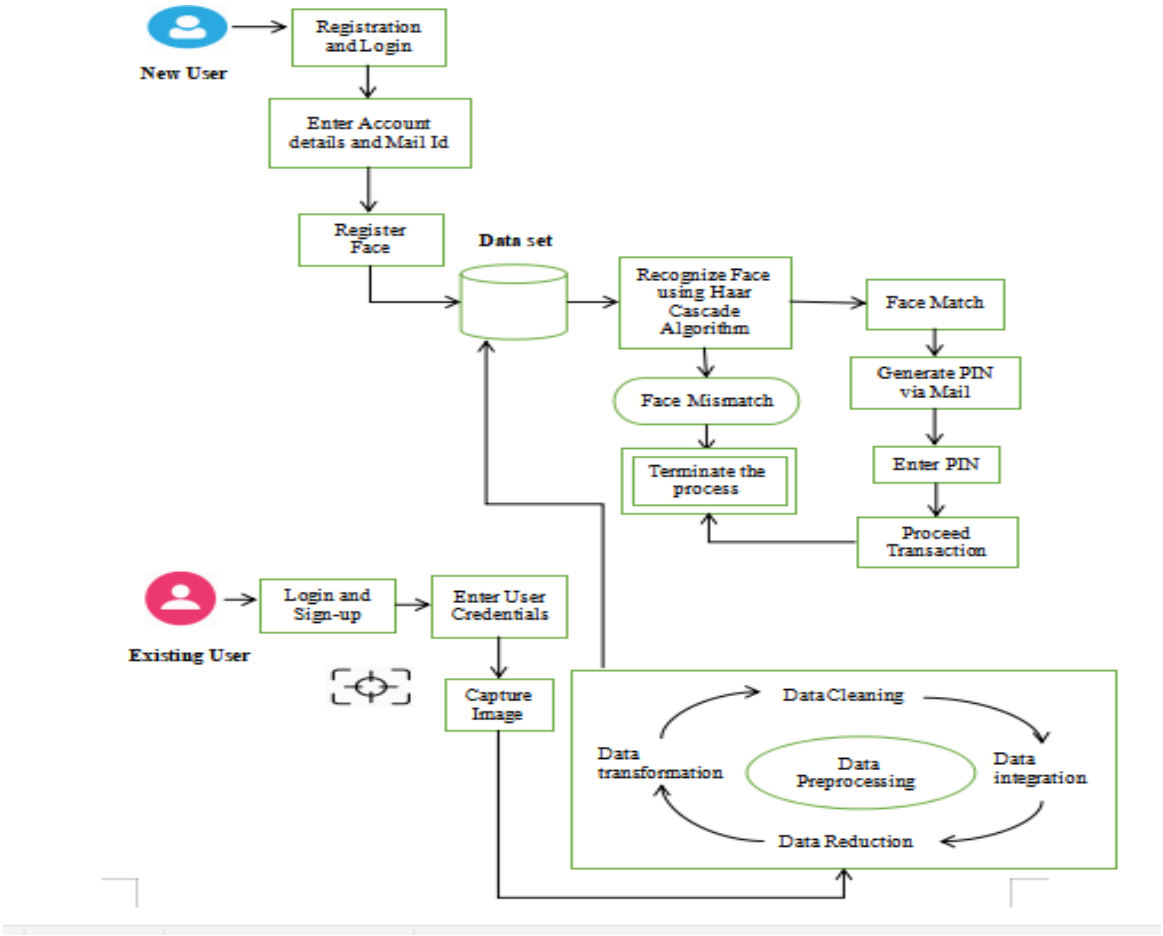

Fig 3.3 Block Diagram

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts

information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

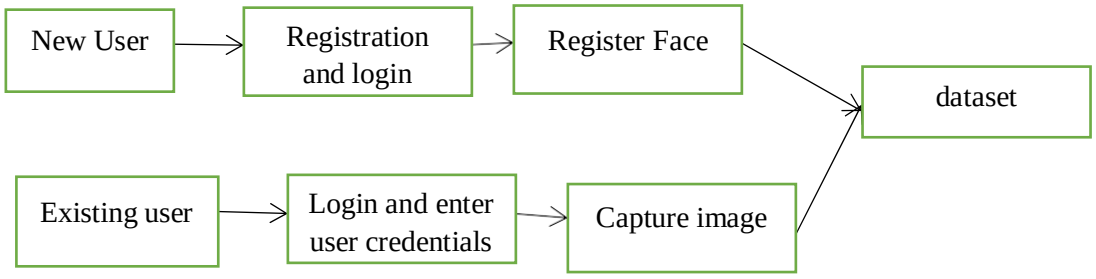

Fig 3.4 Data Flow Diagram 1

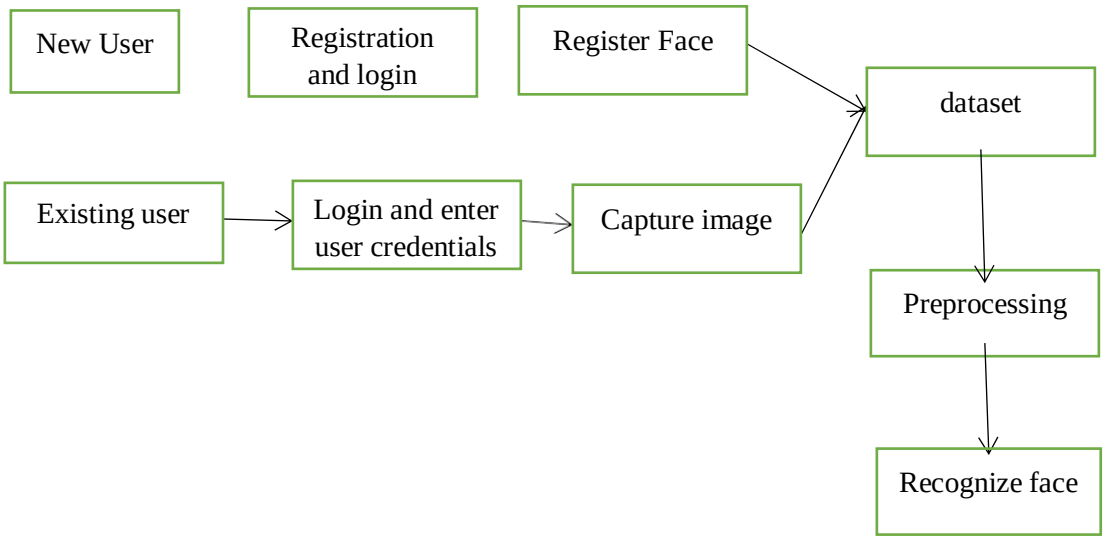

Fig 3.5 Data Flow Diagram 2

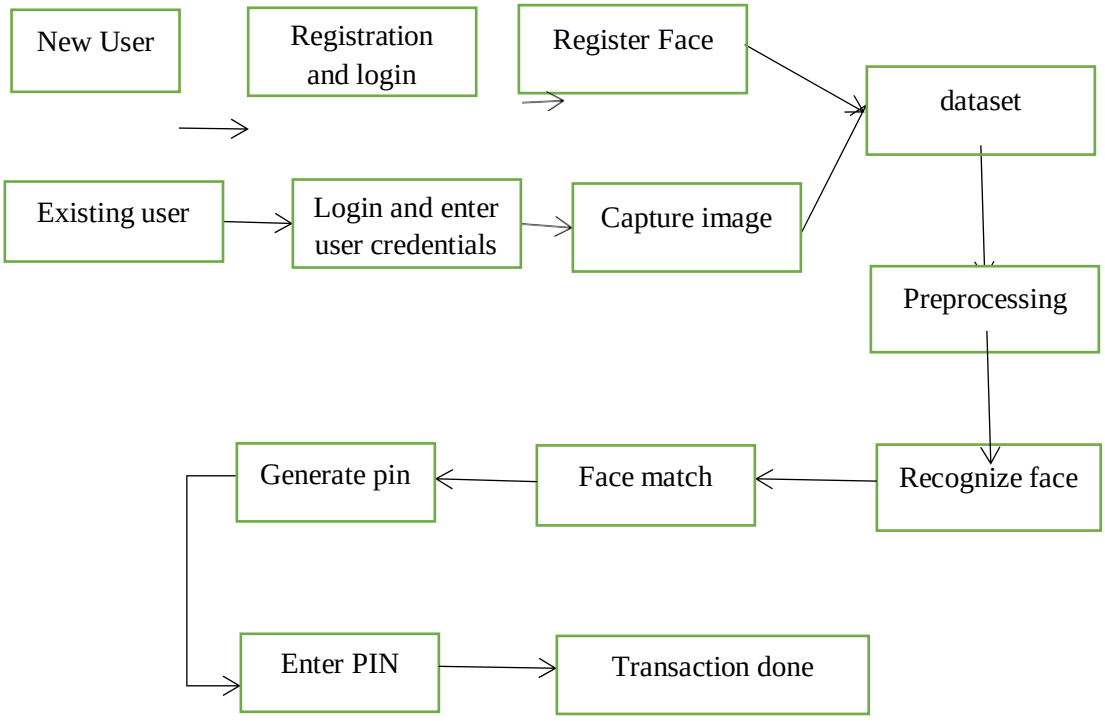

Fig 3.6 Data Flow Diagram 3

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

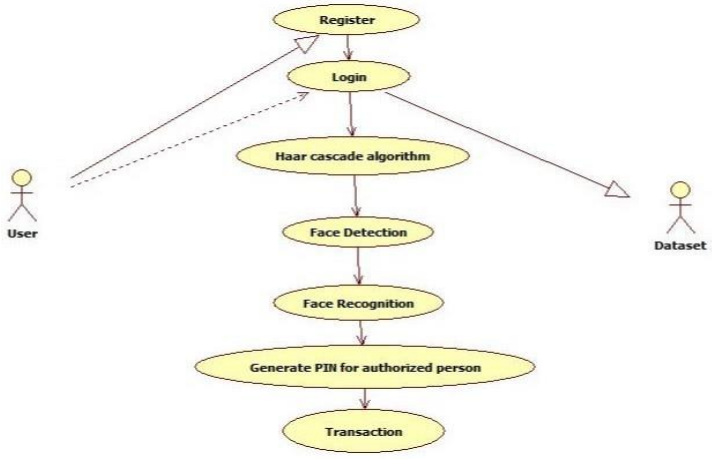

Fig 3.7 Use Case Diagram

CHAPTER 4

SYSTEM REQUIREMENTS

HARDWARE SPECIFICATIONS

- ✓ PROCESSOR: INTEL CORE i5
- ✓ RAM: 8 GB
- ✓ PROCESSOR: 2.4 GHZ
- ✓ MAIN MEMORY: 8GB RAM
- ✓ PROCESSING SPEED: 600 MHZ
- ✓ HARD DISK DRIVE: 1TB
- ✓ KEYBOARD :104 KEYS

SOFTWARE REQUIREMENTS

- ✓ FRONT END: PYTHON
- ✓ IDE: ANACONDA
- ✓ OPERATING SYSTEM: WINDOWS 10

MODULES IMPLEMENTATION

1. Data Preprocessing
2. Feature Extraction
3. Face Recognition

Data Pre-processing

1. It is a technique that is used to convert the raw data into a clean data set.

2. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Feature extraction

1. It is the process of transforming the raw pixel values from an image, to a more meaningful and useful information that can be used in other techniques, such as point matching or machine learning.

Face Recognition

1. Face recognition technology: Ideal for access control, financial transactions and ATM machines. The face key recognition technology performs the following tasks:

a. Locates a moving object within the camera view

b. Determines if the moving object is face

c. Compares live faces with samples from database.

SOFTWARE DESCRIPTION

SQLite

SQLite is not directly comparable to client/server SQL database engines such as MySQL, Oracle, PostgreSQL, or SQL Server since SQLite is trying to solve a different problem.

Client/server SQL database engines strive to implement a shared repository of enterprise data. They emphasize scalability, concurrency, centralization, and control. SQLite strives to provide local data storage for individual applications and devices. SQLite emphasizes economy, efficiency, reliability, independence, and simplicity

SQLite3 can be integrated with Python using sqlite3 module, which was written by Gerhard Haring. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards.

To use sqlite3 module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements.

Python sqlite3 module APIs

Following are important sqlite3 module routines, which can suffice your requirement to work with SQLite database from your Python program. If you are looking for a more sophisticated application, then you can look into Python sqlite3 module's official documentation.

sqlite3.connect(database [,timeout ,other optional arguments])

This API opens a connection to the SQLite database file. You can use ":memory:" to open a database connection to a database that resides in RAM instead of on disk. If database is opened successfully, it returns a connection object.

When a database is accessed by multiple connections, and one of the processes modifies the database, the SQLite database is locked until that transaction is committed. The timeout parameter specifies how long the connection should wait for the lock to go away until raising an exception. The default for the timeout parameter is 5.0 (five seconds).

If the given database name does not exist then this call will create the database. You can specify filename with the required path as well if you want to create a database anywhere else except in the current directory.

connection.cursor([cursorClass])

This routine creates a cursor which will be used throughout of your database programming with Python. This method accepts a single optional parameter cursorClass. If supplied, this must be a custom cursor class that extends sqlite3.

connection.close()

This method closes the database connection. Note that this does not automatically call commit(). If you just close your database connection without calling commit() first, your changes will be lost!

Connect to Database

Following Python code shows how to connect to an existing database. If the database does not exist, then it will be created and finally a database object will be returned.

```
#!/usr/bin/python

import sqlite3

conn = sqlite3.connect('test.db')

print "Opened database successfully";
```

Here, you can also supply database name as the special name :memory: to create a database in RAM. Now, let's run the above program to create our database test.db in the current directory. You can change your path as per your requirement. Keep the above code in sqlite.py file and execute it as shown below. If the database is successfully created, then it will display the following message.

```
$chmod +x sqlite.py

$./sqlite.py

Open database successfully
```

PYTHON LANGUAGE

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Pythons are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

Applications of Python Programming

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy

and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

OPENCV PACKAGE

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation.

And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can

do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

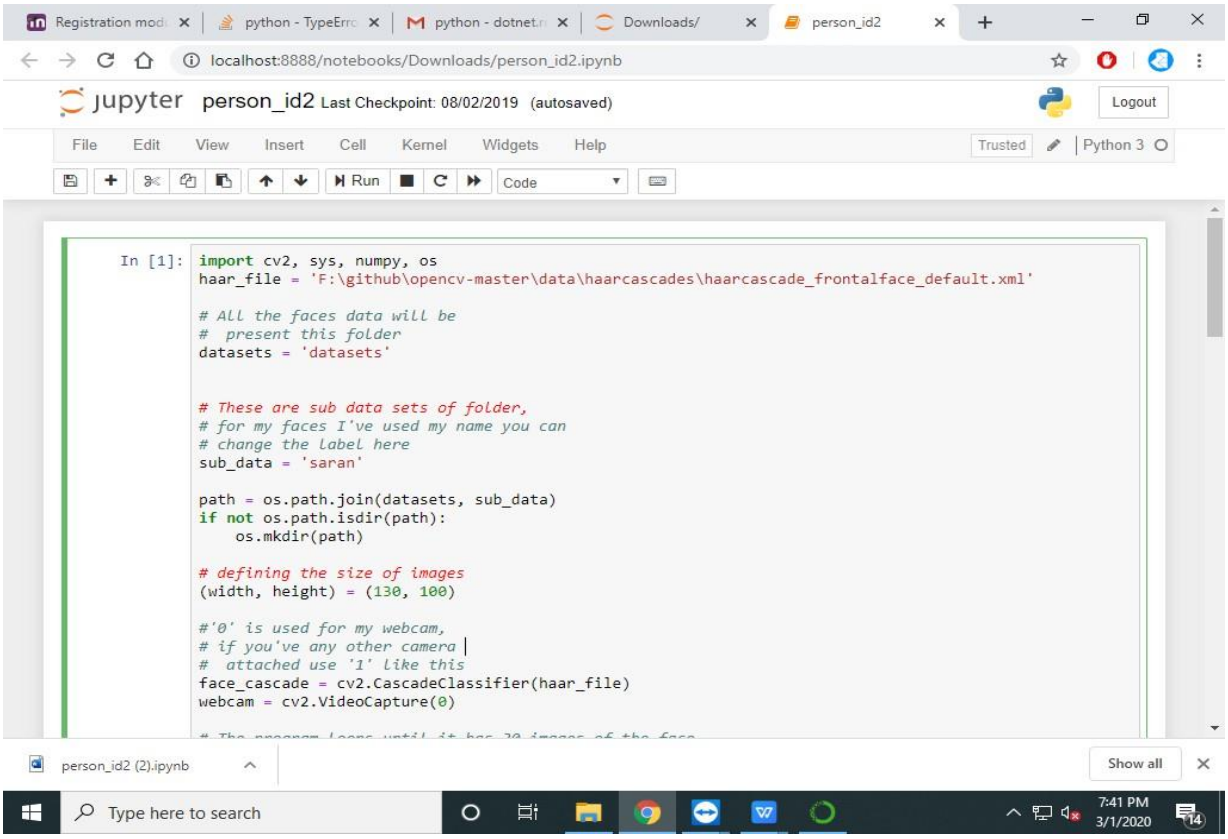

Fig 4.1 Code for Importing dataset

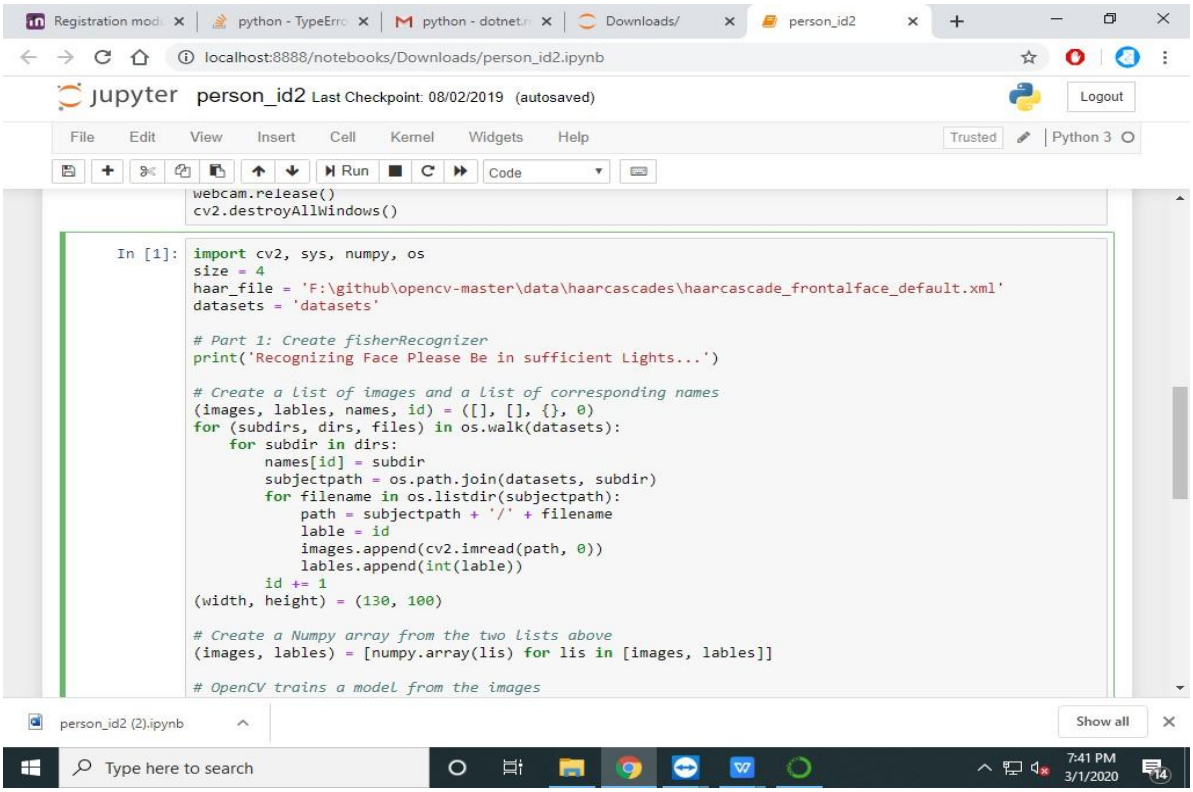

Fig 4.2 Code for Creating Fisher Recognizer

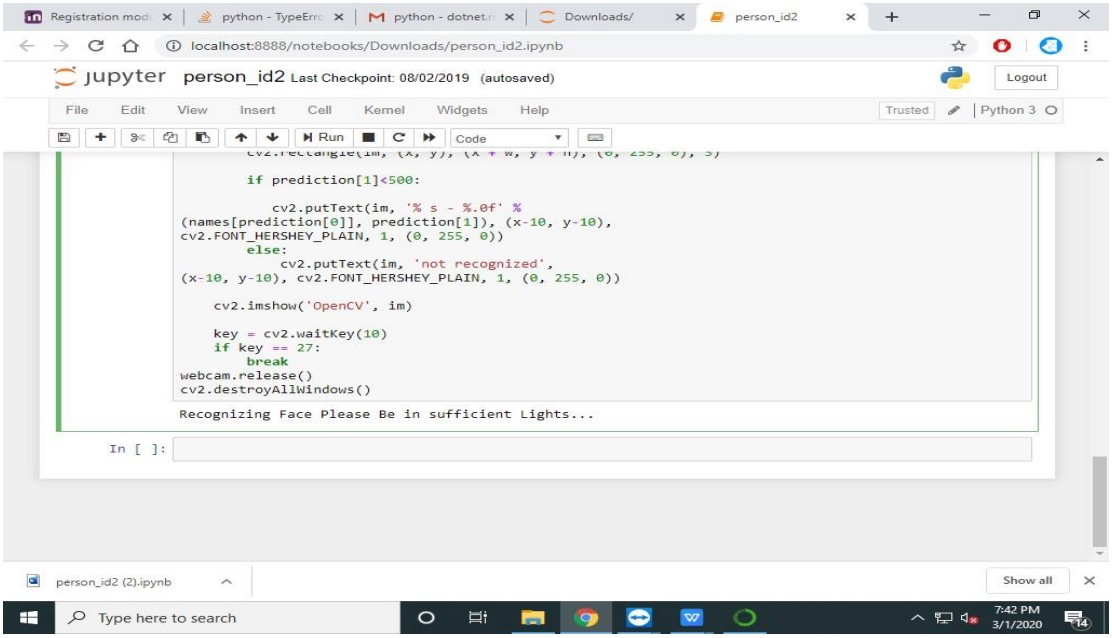

Fig 4.3 Code for Capturing Face

CHAPTER 5

RESULTS AND DISCUSSIONS

In this study, we introduced a facial recognition system to provide a secured and reliable bank transaction. The introduction of the deep for facial authentication had proven to be effective in maximizing security level when performing banking transactions. It is expected that the security level of mobile banking to increase with the employment networks for face authentication.

REGISTRATION

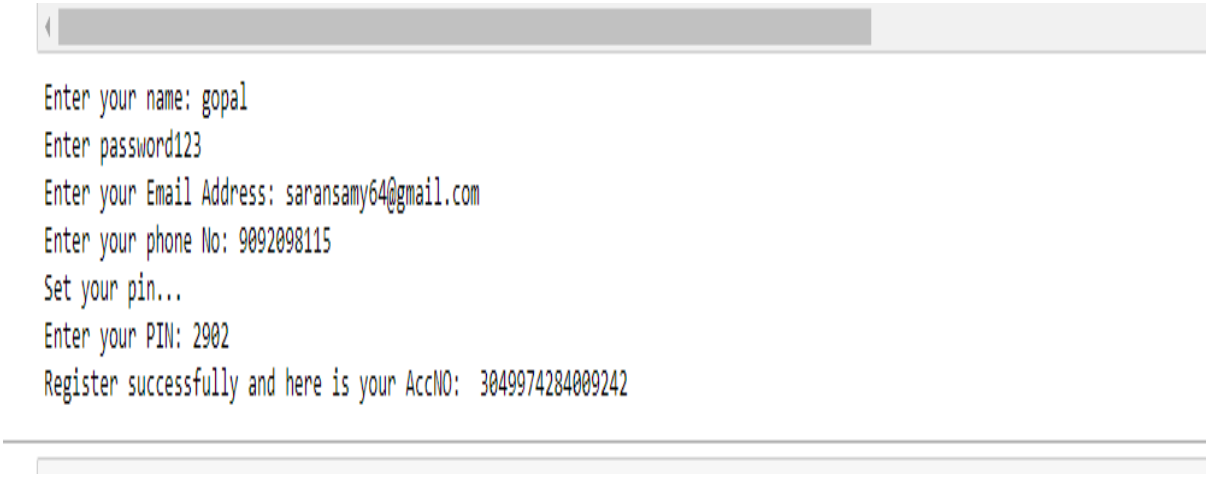

Fig 5.1 User Input for Registration

Fig 5.1 represents the UI to take input from the user for registering a new user. It takes in few information, after putting the pin it will show the registered successfully and it will show the account no of the registered user.

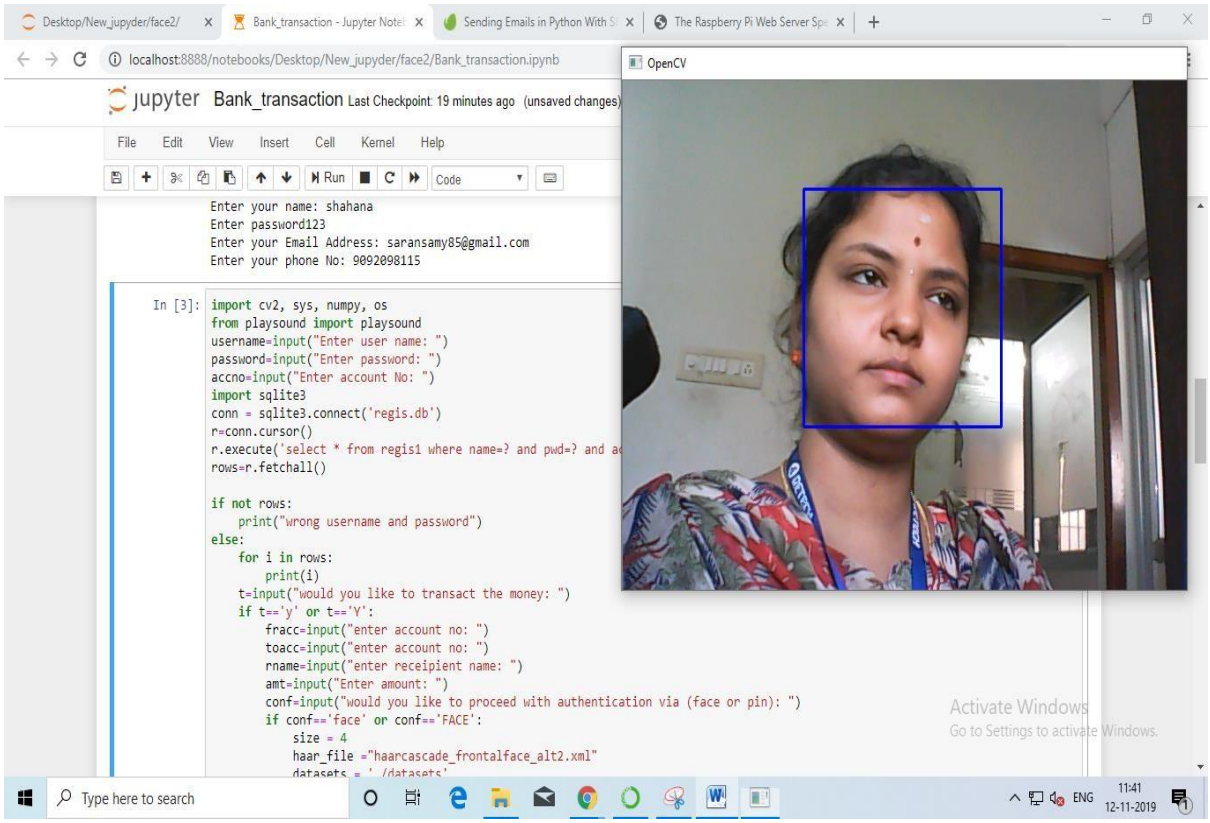

Fig 5.2 Identifying Face

EMAIL ACKNOWLEDGEMENT

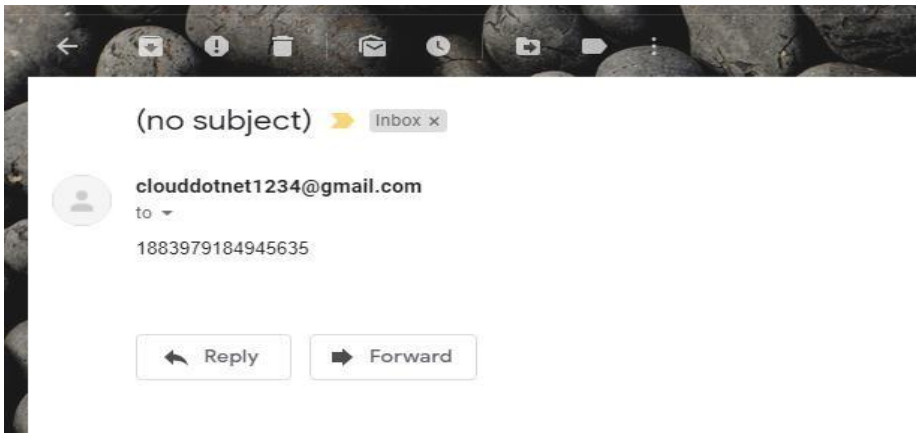

Fig 5.3 Email Acknowledgement

Fig 5.3 shows the email acknowledgement which shows after registering the account no will be send to the linked email id.

LOGIN AND TRANSACTION USING PIN

```
Enter user name: ani
Enter password: 321
Enter account No: 5514028674478257
('ani', '321', 'dotnet.retech@gmail.com', '896574258', '5514028674478257', '1000', '4444')
would you like to transact the money: y
enter account no: 5514028674478257
enter account no: 640
enter receipient name: tharik
Enter amount: 200
would you like to proceed with authentication via (face or pin): pin
confirm your PIN: 4444
Transaction Done!

In [ ]:
```

Fig 5.4 User Input for Transaction

TRANSACTION USING FACE AUTHENTICATION

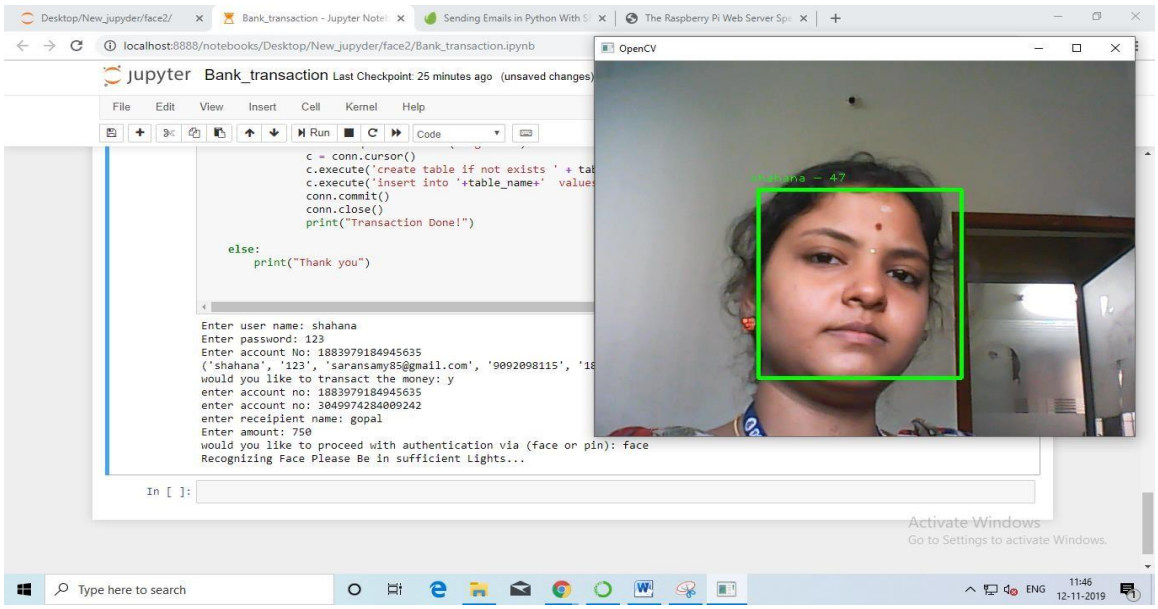

Fig 5.5 Face Detection

Fig 5.6 Transaction Receipt

Fig 5.6 shows the full transaction process where user enters different account details. After entering the details you are asked to do the authentication using face or pin, after finishing the authentication you current balance will show.

CHAPTER 6

CONCLUSION AND FUTURE WORK

CONCLUSION

Realized a reliable, real-time face recognition system on machine learning. According to the new technical era, some advancement has taken place and some techniques of facial recognition have achieved popularity. We are using Haar cascade algorithm for face recognition. Capture module deals with the configuration of video interface and performs the real-time video capture. Face Detection module analyses each captured frame and extracts valid faces from each frame. Face Identification deals with face recognition and verification of the detected face. In Future any fraudulent access by the fake user is eliminated with the help of radio frequency identification card**.**

FUTURE WORK

If more work were to be done on this project the main priority would be to develop better classifiers, choosing a larger object, have more samples and use more advanced feature (not only up right ones). This could be done with a little effort. An alternative approach to improve the recognition robustness of the cascade classifier, is to look into optimizing a implementation of a local key point extraction based technique. The FAST and SUSAN detectors would then be two main candidates, based on a performance chart. Recently, in the 2.4.2 release of OpenCV (released in July 2012), a new key point descriptor called FREAK has been added to the library and is claimed to be very fast and “superior to ORB and SURF descriptors”. The main issue for such approach would be to scale it up and make it able to identify a big set of objects. On the other hand, natural feature tracking (marker-less AR) is more difficult than marker-based tracking (using QR/AR-tags) in most, if not all cases where marker-based techniques are applicable, since such tags are created for recognition (containing clear features). Maybe it was not realistic to think that a local feature-based approach would be a good base for a system able to identify many objects. The issue of what object recognition

systems are buildable is very specific to situation and environment. It would be achievable to construct a real time object recognition system based on local features for a small set of objects, but for a virtually unlimited set (e g. all components of a car) it would be more realistic to use a code-based method or a machine learning approach.

REFERENCES

[1] Facial-Recognition Payment: An Example of Chinese Consumers, Wen Kun Zhang ; Min Jung Kang, IEEE Access, Year: 2019

[2] Secure multifactor authentication payment system using NFC, Anirudhan Adukkathayar ; Gokul S Krishnan ; Rajashree Chinchole, 2015 10th International Conference on Computer Science & Education (ICCSE)\

[3] Biometric Face Recognition Payment System , Surekha. R. Gondkar Saurab. Dr. C. S. Mala International Journal of Engineering Research & Technology NCESC - 2018 Conference Proceedings

[4] Facial Recognition in Banking – Current Applications, Niccolo Mejia,2019 Conference Proceedings

[5] "Face Detection and Recognition for Bank Transaction ", International Journal of Emerging Technologies and Innovative Research , Sudarshan Dumbre ,Shamita Kulkarni ,Devashree Deshpande ,P.V.Mulmule Journal of Emerging Technologies and Innovative Research 2018

[6] Continuous User Identity Verification Using Biometric Traits for Secure Internet Services, Dr.SHAIK ADBUL MUZZER, 2GOSALA SUBHASIN

[7] Skin color based Face detection Method, Devendra Singh Raghuvanshi,Dheeraj Agrawal

[8] Face Detection system based on retinal connected neural network (RCNN), Rowley, Baluja and Kanade

[9] Combining Skin Color based Classifiers and HAAR Feature using VJ Algorithm, N.Gobinathan, Abinaya and Geetha. P

[10] Face Detection and Recognition for Bank Transaction, Sudarshan Dumbre1 , Shamita Kulkarni2 , Devashree Deshpande3 , Prof P.V.Mulmule4

[11] ‘Haxby, J.V., Ungerleider, L.G., Horwitz, B., Maisog, J.M., Rapoport, S.I., and Grady, C.L. (1996). Face encoding and recognition in the human brain. Proc. Nat.Acad. Sci. 93: 922 – 927

APPENDIX

A. SOURCE CODE

```
import cv2, sys, numpy, os

haar_file = 'F:\github\opencv-
master\data\haarcascades\haarcascade_frontalface_default.xml'


# Read in the cascade classifiers for face and eyes

face_cascade = cv2.CascadeClassifier('../DATA / haarcascades /
haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('../DATA / haarcascades /
haarcascade_eye.xml')



# create a function to detect face

def adjusted_detect_face(img):

    face_img = img.copy()

    face_rect = face_cascade.detectMultiScale(face_img,
                                        scaleFactor = 1.2,
                                        minNeighbors = 5)

    for (x, y, w, h) in face_rect:
        cv2.rectangle(face_img, (x, y),
                    (x + w, y + h), (255, 255, 255), 10)\
```

```
    return face_img


# create a function to detect eyes

def detect_eyes(img):

    eye_img = img.copy()
    eye_rect = eye_cascade.detectMultiScale(eye_img,
                                        scaleFactor = 1.2,
                                        minNeighbors = 5)
    for (x, y, w, h) in eye_rect:
        cv2.rectangle(eye_img, (x, y),
                      (x + w, y + h), (255, 255, 255), 10)
    return eye_img

# Reading in the image and creating copies

img = cv2.imread('../sachin.jpg')
img_copy1 = img.copy()
img_copy2 = img.copy()
img_copy3 = img.copy()

# Detecting the face

face = adjusted_detect_face(img_copy)
plt.imshow(face)


# All the faces data will be
```

```
#    present this folder

datasets = 'datasets'



# These are sub data sets of folder,
# for my faces I've used my name you can
# change the label here

sub_data = 'saran'

path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
        os.mkdir(path)

# defining the size of images

(width, height) = (130, 100)

#'0' is used for my webcam,
# if you've any other camera
#  attached use '1' like this
```

```
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

# The program loops until it has 30 images of the face.

count = 1
while count < 30:
    (_, im) = webcam.read()
    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), (255, 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('OpenCV', im)
    key = cv2.waitKey(10)
    if key == 27:
        break
```

```
webcam.release()

cv2.destroyAllWindows()



import cv2, sys, numpy, os

size = 4

haar_file = 'F:\github\opencv-
master\data\haarcascades\haarcascade_frontalface_default.xml'

datasets = 'datasets'



# Part 1: Create fisherRecognizer

print('Recognizing Face Please Be in sufficient Lights...')



# Create a list of images and a list of corresponding names

(images, lables, 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
                lable = id
                images.append(cv2.imread(path, 0))
                lables.append(int(lable))
            id += 1
(width, height) = (130, 100)


# Create a Numpy array from the two lists above

(images, lables) = [numpy.array(lis) for lis in [images, lables]]


# OpenCV trains a model from the images
# NOTE FOR OpenCV2: remove '.face'

model = cv2.face.LBPHFaceRecognizer_create()
model.train(images, lables)


# Part 2: Use fisherRecognizer on camera stream
```

```
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

while True:
    (_, im) = webcam.read()
    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, 0, 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:



            cv2.putText(im, '% s - %.0f' %
(names[prediction[0]], prediction[1]), (x-10, y-10),
cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0))
        else:
```

```
                cv2.putText(im, 'not recognized',
(x-10, y-10), cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0))

    cv2.imshow('OpenCV', im)


    key = cv2.waitKey(10)
    if key == 27:
        break

webcam.release()
cv2.destroyAllWindows()
```