

MULTI-LEVEL SECURE TRANSACTION USING FACIAL AUTHENTICATION (TranSecure)

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

by

Varun Vaibhav (Register No. 38110619)

Tapadyuti Mal (Register No. 38110592)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH – 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in**

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VARUN VAIBHAV (38110619), TAPADYUTI MAL (38110592)** who carried out the project entitled **"MULTI-LEVEL SECURE TRANSACTION USING FACIAL AUTHENTICATION (TranSecure)"** under my supervision from November 2021 to March 2022

**Internal Guide
Dr. P. Asha, M.E., Ph.D.**

Head of the Department

Submitted for Viva-voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **VARUN VAIBHAV (38110619) & TAPADYUTI MAL (38110592)** hereby declare that the Project Report entitled **“MULTI-LEVEL SECURE TRANSACTION USING FACIAL AUTHENTICATION (TranSecure)”** done by us under the guidance of **Dr. P. ASHA, M.E., Ph.D.**, (Internal) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in COMPUTER SCIENCE AND ENGINEERING.

DATE:

PLACE:

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 **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr. S. Vigneshwari M.E., Ph.D. Heads** of the 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 **Dr. P. Asha, M.E., Ph.D.** for her 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** were helpful in many ways for the completion of the project.

ABSTRACT

In this project we are going to develop a robust automated web application for transacting money in higher level security purpose with high facial recognition. First we have to register our personal details with our face. 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, eyebrows, nose, mouth of candidates are extracted from the intensity valleys from the detected face it will consider as datasets now our system can understand who are all authorized and unauthorized. While login our face will be recognized if we authorized will can move to authorized page else our system won't allow to login. For higher security purpose we have used face recognition module. During the transaction our face will be recognized it will allow only the authorized account holder to transact a money, it doesn't allow others to transact a money.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	INTRODUCTION	1
	1.1 DOMAIN DESCRIPTION	1
	1.2 PROBLEM DEFINITION	7
	1.3 PROJECT DESCRIPTION	7
2	LITERATURE SURVEY	8
	2.1 AIM AND SCOPE	13
	2.2 EXISTING SYSTEM	14
3	METHODOLOGY	15
	3.1 PROPOSED SYSTEM	15
	3.2 FEASIBILITY STUDY	16
	3.3 SYSTEM DESIGN	18
	3.3.1 SYSTEM ARCHITECTURE	18
	3.3.2 DATA FLOW DIAGRAMS	18
	3.3.3 UML DIAGRAMS	21
	3.3.4 USE CASE DIAGRAMS	22
	3.3.5 SEQUENCE DIAGRAM	23
	3.3.6 ACTIVITY DIAGRAM	24
	3.3.7 CLASS DIAGRAMS	24
	3.4 SYSTEM REQUIREMENTS	25
	3.4.1 HARDWARE SPECIFICATION	25
	3.4.2 SOFTWARE SPECIFICATION	25
	3.5 MODULES	26

	3.6	WORKING	27
	3.7	SOFTWARE DESCRIPTION	27
	3.8	LANGUAGES USED	33
	3.8.1	SQLite	33
	3.8.2	PYTHON	35
	3.8.3	ABOUT OPENCV PACKAGE	37
	3.9	SOFTWARE TESTING	37
	3.9.1	GENERAL	37
	3.9.2	TEST CASE	38
	3.9.3	TESTING TECHNIQUES	38
4		RESULTS AND DISCUSSION	41
	4.1	RESULTS	41
	4.2	DISCUSSION	46
5		CONCLUSION AND FUTURE WORK	47
	5.1	CONCLUSION	47
	5.2	FUTURE WORK	47
		REFERENCES	48
		APPENDIX	
	A.	SOURCE CODE	49
	B.	PLAGIARISM REPORT	71

LIST OF FIGURES

FIG NO.	NAME OF THE FIGURE	PAGE NO.
3.1	SYSTEM ARCHITECTURE	18
3.2	DFD-0	19
3.3	DFD-1	20
3.4	DFD-2	21
3.5	USE CASE DIAGRAM	23
3.6	SEQUENCE DIAGRAM	23
3.7	ACTIVITY DIAGRAM	24
3.8	CLASS DIAGRAM	25
4.1	HOME PAGE	41
4.2	REGISTRATION PAGE	42
4.3	FACE RECOGNITION (REGISTERING USER)	42
4.4	LOGIN PAGE	43
4.5	FACE RECOGNITION (USER LOGIN)	43
4.6	TRANSACTION PAGE	44
4.7	TRANSACTION DETAILS OF AN USER	44
4.8	DATA SET COLLECTED	45
4.9	REGISTRATION RECORD (DB-SQLite)	45
4.10	TRANSACTION RECORD (DB-SQLite)	46

LIST OF ABBREVIATIONS

EMV cards	-	Europay, MasterCard, VISA cards
KYC	-	Know Your Customer
AI	-	Artificial Intelligence
GPS	-	Global Positioning System
ML	-	Machine Learning
ATC	-	Active Traction Control
QR	-	Quick Response
NFC	-	Near Field Communication
PIN	-	Personal Identification Number
ATM	-	Automated Teller Machine
RFID	-	Radio Frequency Identification
HSV	-	Hue Saturation Value
ROC curve	-	Receiver Operating Characteristic Curve
OTP	-	One Time Password
SET	-	Secure Electronic Transaction
DFD	-	Data Flow Diagram
UML	-	Unified Modeling Language
CMS	-	Content Management System

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).

1.1 DOMAIN DESCRIPTION

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 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.

In contrast, **unsupervised machine learning algorithms** are 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 fall 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 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 behaviour 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.

Advantages of Machine Learning

Machine Learning undoubtedly helps people to work more creatively and efficiently. Basically, you too can delegate quite complex or monotonous work to the computer through Machine Learning – starting with scanning, saving and filing paper documents such as invoices up to organizing and editing images.

In addition to these rather simple tasks, self-learning machines can also perform complex tasks. These include, for example, the recognition of error patterns. This is a major advantage, especially in areas such as the manufacturing industry: the industry relies on continuous and error-free production. While even experts often cannot be sure where and by which correlation a production error in a plant fleet arises, Machine Learning offers the possibility to identify the error early this saves down times and money. Self-learning programs are now also used in the medical field. In the future, after “consuming” huge amounts of data (medical publications, studies, etc.), apps will be able to warn a in case his doctor wants to prescribe a drug that he cannot tolerate. This “knowledge” also means that the app can propose alternative options which for example also take into account the genetic requirements of the respective patient.

Applications of Machine Learning

1. Virtual Personal Assistants

Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice. All you need to do is activate them and ask “What is my schedule for today?”, “What are the flights from Germany to London”, or similar questions. For answering, your personal assistant looks out for the information, recalls your related queries, or send a command to other resources (like phone apps) to collect info. You can even instruct assistants for certain tasks like “Set an alarm for 6 AM next morning”, “Remind me to visit Visa Office day after tomorrow”. Machine learning is an important part of these personal assistants as they collect and refine the information on the basis of your previous involvement with them. Later, this set of data is utilized to render results that are tailored to your preferences. Virtual Assistants are integrated to a variety of platforms. For example: Smart Speakers: Amazon Echo and Google Home Smartphones: Samsung Bixby on Samsung S8 Mobile Apps: Google Allow.

2. Predictions while Commuting

Traffic Predictions: We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS. Machine learning in such scenarios helps to estimate the regions where congestion can be found on the basis of daily experiences.

Online Transportation Networks: When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning. Jeff Schneider, the engineering lead at Uber ATC reveals in an interview that they use ML to define price surge hours by predicting the rider demand. In the entire cycle of the services, ML is playing a major role.

3. Videos Surveillance

Imagine a single person monitoring multiple video cameras! Certainly, a difficult job to do and boring as well. This is why the idea of training computers to do this job makes sense. The video surveillance system nowadays are powered by AI that makes it possible to detect crime before they happen. They track unusual behaviour of people like standing motionless for a long time, stumbling, or napping on benches etc. The system can thus give an alert to human attendants, which can ultimately help to avoid mishaps. And when such activities are reported and counted to be true, they help to improve the surveillance services. This happens with machine learning doing its job at the backend.

4. Social Media Services

From personalizing your news feed to better ads targeting, social media platforms are utilizing machine learning for their own and user benefits. Here are a few examples that you must be noticing, using, and loving in your social media accounts, without realizing that these wonderful features are nothing but the applications of ML. People

You May Know: Machine learning works on a simple concept: understanding with experiences. Facebook continuously notices the friends that you connect with, the profiles that you visit very often, your interests, workplace, or a group that you share with someone etc. On the basis of continuous learning, a list of Facebook users are suggested that you can become friends with. Face Recognition: You upload a picture of you with a friend and Facebook instantly recognizes that friend. Facebook checks the poses and projections in the picture, notice the unique features, and then match them with the people in your friend list. The entire process at the backend is complicated and takes care of the precision factor but seems to be a simple application of ML at the front end. Similar Pins: Machine learning is the core element of Computer Vision, which is a technique to extract useful information from images and videos. Pinterest uses computer vision to identify the objects (or pins) in the images and recommend similar pins accordingly.

5. Email Spam and Malware Filtering

There are a number of spam filtering approaches that email clients use. To ascertain that these spam filters are continuously updated, they are powered by machine learning. When rule-based spam filtering is done, it fails to track the latest tricks adopted by spammers. Multi-Layer Perceptron, C 4.5 Decision Tree Induction are some of the spam filtering techniques that are powered by ML. Over 325, 000 malwares are detected every day and each piece of code is 90–98% similar to its previous versions. The system security programs that are powered by machine learning understand the coding pattern. Therefore, they detect new malware with 2–10% variation easily and offer protection against them.

6. Online Customer Support

A number of websites nowadays offer the option to chat with customer support representative while they are navigating within the site. However, not every website has a live executive to answer your queries. In most of the cases, you talk to a chatbot. These bots tend to extract information from the website and present it to the customers. Meanwhile, the chatbots advances with time. They tend to understand the

user queries better and serve them with better answers, which is possible due to its machine learning algorithms.

7. Search Engine Result Refining

Google and other search engines use machine learning to improve the search results for you. Every time you execute a search, the algorithms at the backend keep a watch at how you respond to the results. If you open the top results and stay on the web page for long, the search engine assumes that the results it displayed were in accordance to the query. Similarly, if you reach the second or third page of the search results but do not open any of the results, the search engine estimates that the results served did not match requirement. This way, the algorithms working at the backend improve the search results.

8. Product Recommendations

You shopped for a product online few days back and then you keep receiving emails for shopping suggestions. If not this, then you might have noticed that the shopping website or the app recommends you some items that somehow matches with your taste. Certainly, this refines the shopping experience but did you know that it's machine learning doing the magic for you? On the basis of your behaviour with the website/app, past purchases, items liked or added to cart, brand preferences etc., the product recommendations are made.

9. Online Fraud Detection

Machine learning is proving its potential to make cyberspace a secure place and tracking monetary frauds online is one of its examples. For example: PayPal is using ML for protection against money laundering. The company uses a set of tools that helps them to compare millions of transactions taking place and distinguish between legitimate or illegitimate transactions taking place between the buyers and sellers.

1.2 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 e.g. 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.

1.3 PROJECT DESCRIPTION

Facial recognition is one of numerous ways banks can decrease friction in their customers' experience and increase efficiency and accessibility. This project makes 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

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.

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.

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.

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.

“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.

Continuous User Identity Verification Using Biometric Traits for Secure Internet Services

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

Skin colour based face detection:**Author: Devendra Singh Raghuvanshi, Dheeraj Agrawal**

Detection of skin color in color images is a very popular and useful technique for face detection. Color is an important feature of human faces. Using skin-color as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features. In the skin color detection process, each pixel was classified as skin or non-skin based on its color components. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model. The first step of face detection is to segment the color image into skin and non-skin region. Different color space has different ranges of pixels which represents skin region and non-skin region.

Face Detection system based on retinal connected neural network (RCNN):**Author: Rowley, Baluja and Kanade**

It first applies a set of neural network-based filters to an image. It uses an arbitrator to combine the outputs. The filters examine each location in the image at several scales, looking for locations that might contain a face. The arbitrator then merges detections from individual filters and eliminates overlapping detections. This method produces good detection rates (77.9% and 90.3%) with an acceptable number of false positives. Depending on the application, the system can be made more or less conservative by varying the arbitration heuristics or thresholds used. We have also applied the same algorithm for the detection of car tires and human eyes.

Combining Skin Color based Classifiers and HAAR Feature using VJ Algorithm**author: N.Gobinathan, Abinaya and Geetha. P**

D. Gobinathan have presented a hybrid means for face recognition in shade images. The well-known HAAR feature-based encounter detector developed by Viola as well as Jones (VJ) that was designed for gray-scale pictures was along with a skin-color filtration system, which supplies complementary facts in shade images. The image

was passed by way of a HAAR Feature based encounter detector, which ended up being adjusted in a way that it ended up being operating in a point in its ROC curve which has a low quantity of missed people but a top number involving false detections. Their recommended method features eliminated a number of these false detections. They have likewise used a color pay out algorithm in order to reduce the results of illumination. Their experimental results about the Bao shade face databases have showed that this proposed method was better than the unique VJ algorithm.

Face Detection and Recognition for Bank Transaction

Authors: Sudarshan Dumbre , Shamita Kulkarni , Devashree Deshpande , Prof P.V.Mulmule

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.

2.1 AIM AND SCOPE

In this project we are going to develop a robust automated web application for transacting money in higher level security purpose with high facial recognition. First we have to register our personal details with our face. 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, eyebrows, nose, mouth of candidates are extracted from the intensity valleys from the detected face it will consider as datasets now our system can understand who are all authorized and unauthorized. While login our face will be recognized if we authorized will can move to authorized page else our system won't allow to login. For higher security purpose we have used face recognition module. During the transaction our face will be recognized it will allow only the authorized account holder to transact a money, it doesn't allow others to transact a money.

2.2 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.

2.2.1 Disadvantages

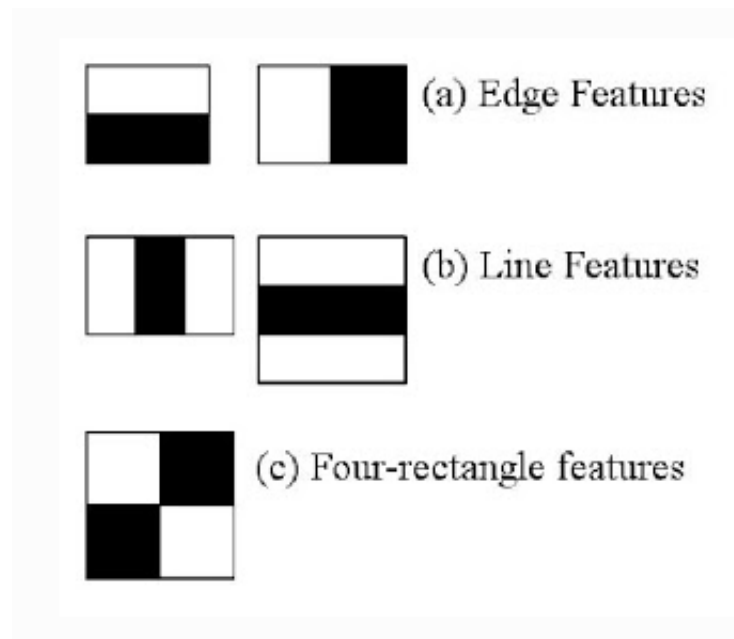
1. User must have credit card
2. It is not cost-effective when the payment is small
3. None of anonymity and it is traceable
4. Network effect – need to install client software (an e-wallet).
5. Cost and complexity for merchants to offer support, contrasted with the comparatively low cost and simplicity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

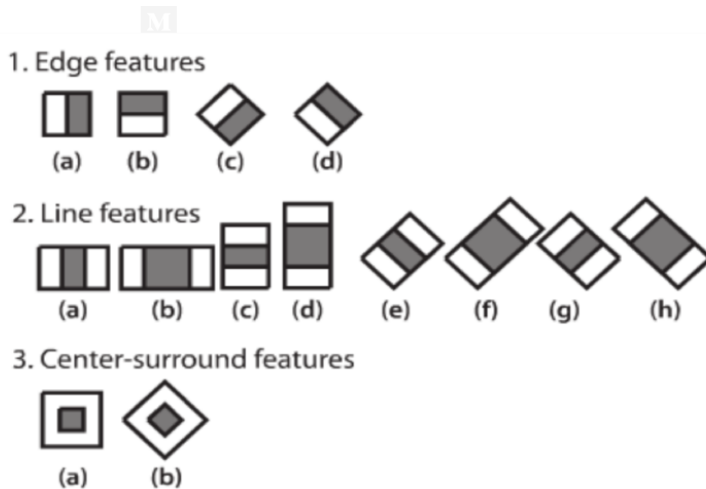
CHAPTER-3

METHODOLOGY

3.1 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.





Face Detection determines the locations and sizes of human faces in arbitrary (digital) images.

In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

3.1.1 ADVANTAGES OF PROPOSED SYSTEM

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.

3.2 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

3.2.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. A feasibility study evaluates the project's potential for success.

3.2.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3 SYSTEM DESIGN

3.3.1 SYSTEM ARCHITECTURE:

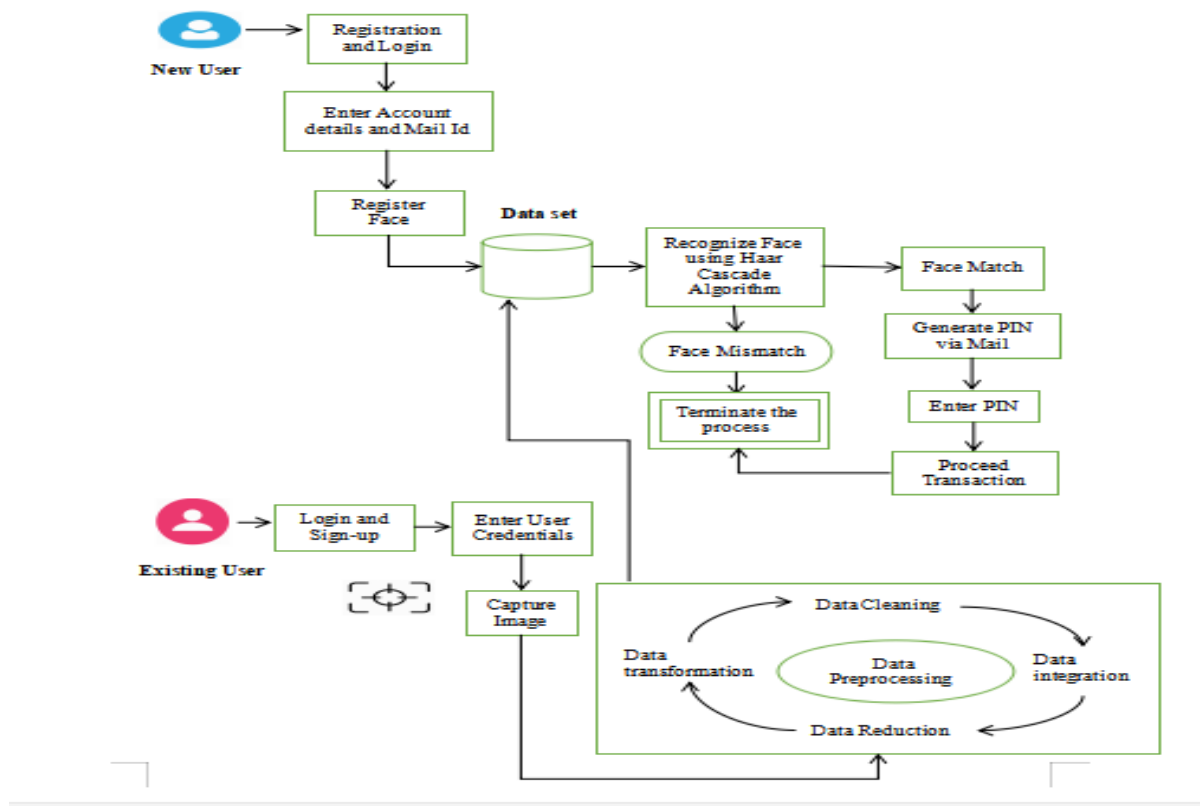


Fig.3.1 System Architecture

3.3.2 DATA FLOW DIAGRAM:

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.

1. 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.

2. 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.
3. 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.

DFD-0

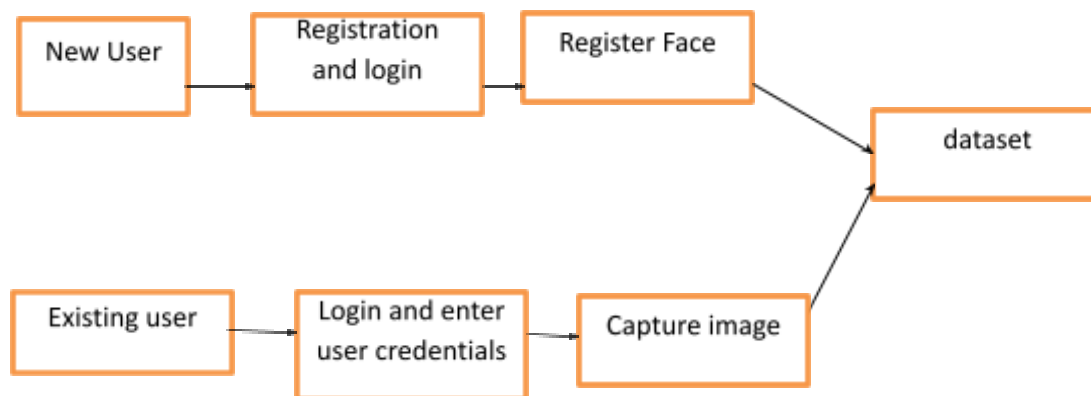


Fig.3.2 DFD-0

DFD-1

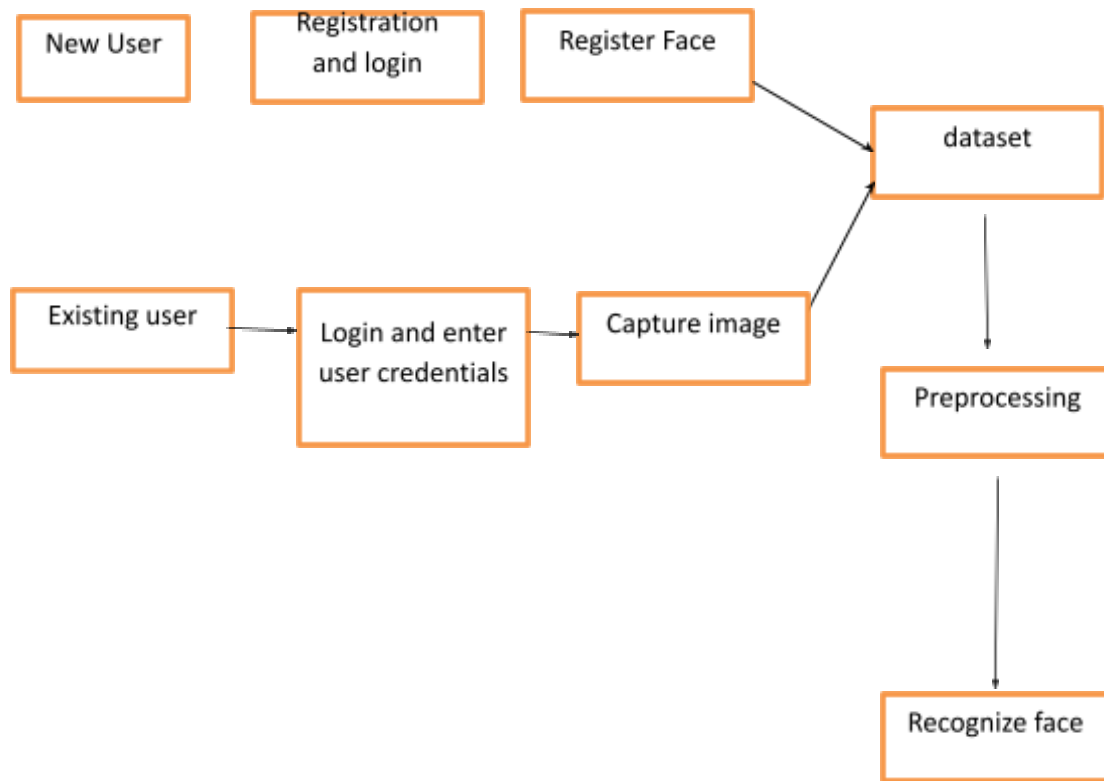


Fig.3.3 DFD-1

DFD-2

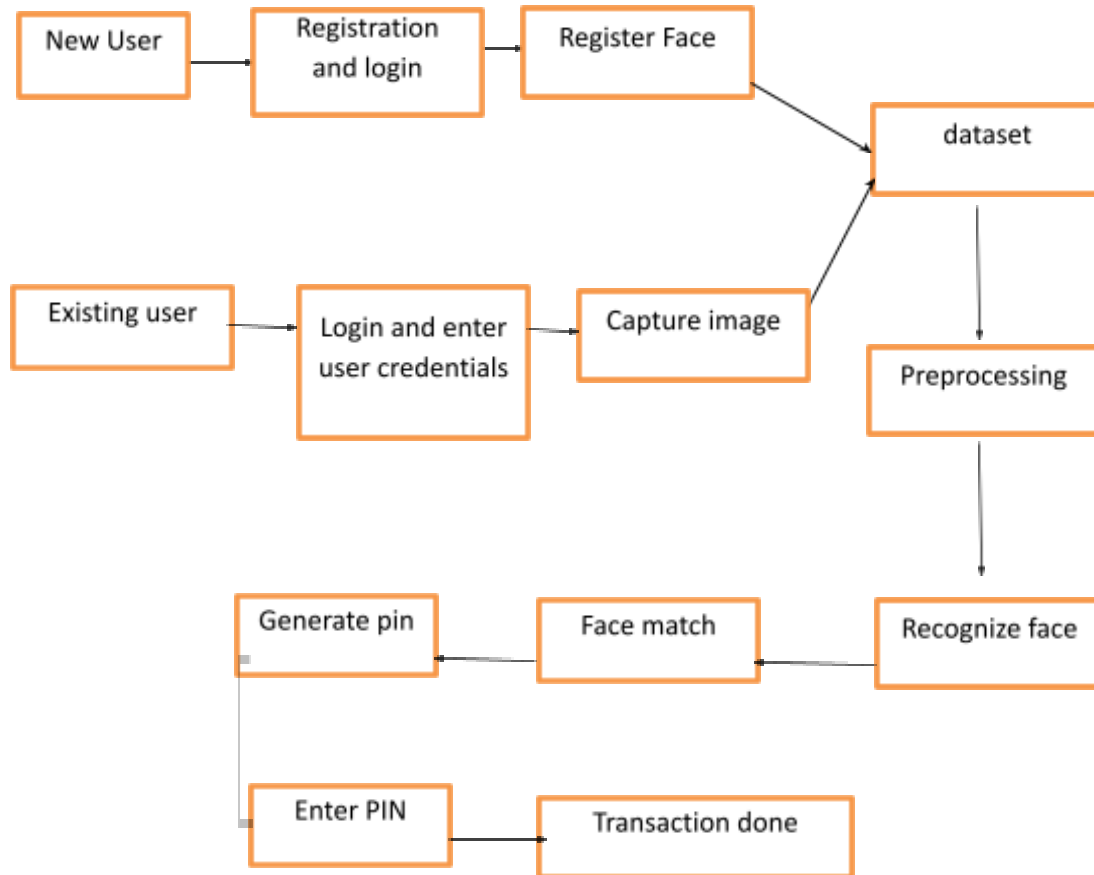


Fig.3.4 DFD-2

3.3.3 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.

3.3.3.1 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.

3.3.4 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.

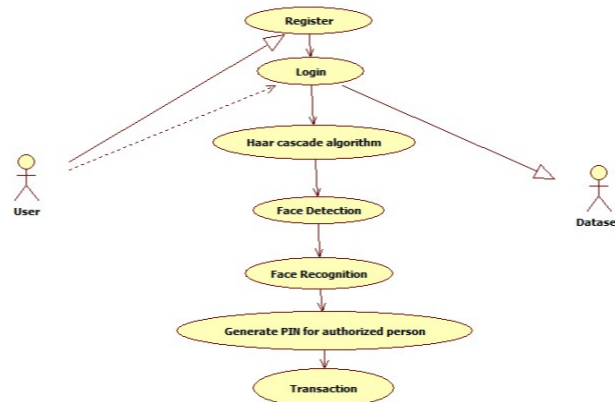


Fig.3.5 Use Case Diagram

3.3.5 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

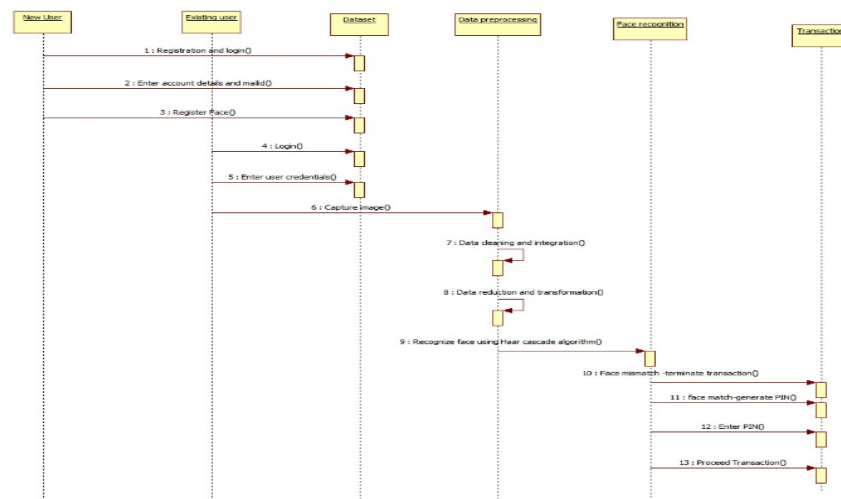


Fig.3.6 Sequence Diagram

3.3.6 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

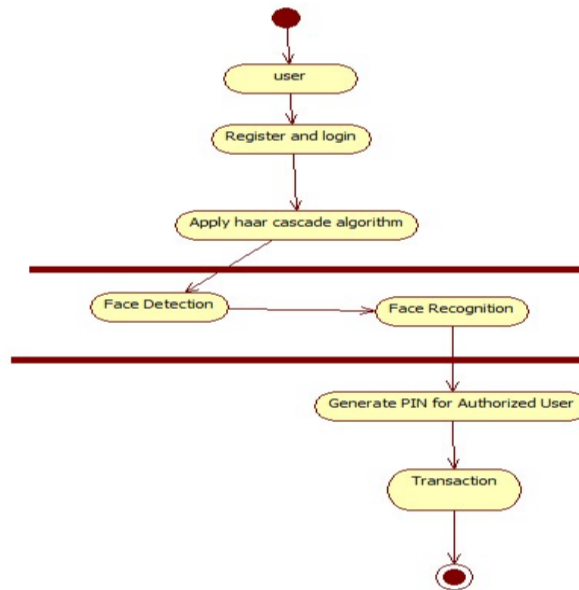


Fig.3.7 Activity Diagram

3.3.7 CLASS DIAGRAM:

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only

used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

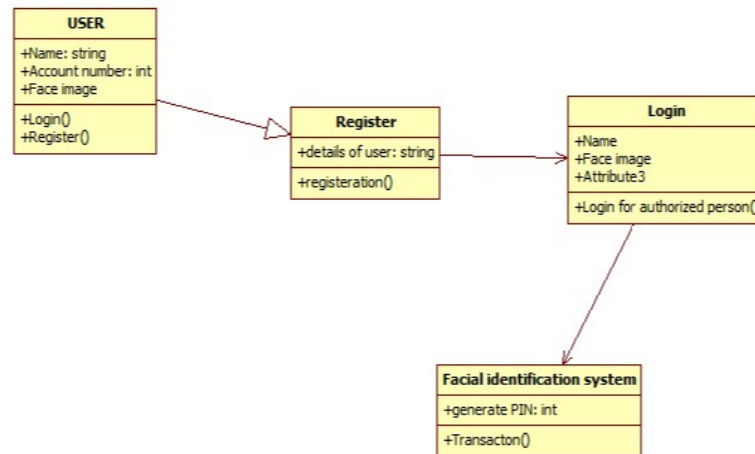


Fig.3.8 Class Diagram

3.4 SYSTEM REQUIREMENTS

3.4.1 Hardware Specification

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

3.4.2 Software Specification

- ✓ FRONT END :PYTHON
- ✓ IDE : ANACONDA

✓ OPERATING SYSTEM :WINDOWS 10

3.5 MODULES

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

Data Preprocessing

- a) It is a technique that is used to convert the raw data into a clean data set.
- b) 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

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

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 the database.

3.6 WORKING:

REGISTER

In this module user have to enter their personal details along with their face. Dataset will be created based on the username.

LOGIN

User have to enter credentials during that period face will be recognized it will allow only authorized user.

TRANSACTION

In this modules user have to enter transaction details like user A/C number, username, amount, Recipient name and Recipient A/C number after this process our system will recognize current transaction user if the user is not authorized our system will not allow to transact.

CHECK BALANCE AND VIEW TRANSACTION DETAILS

It will display user A/C balance. View transaction module will display user transaction details.

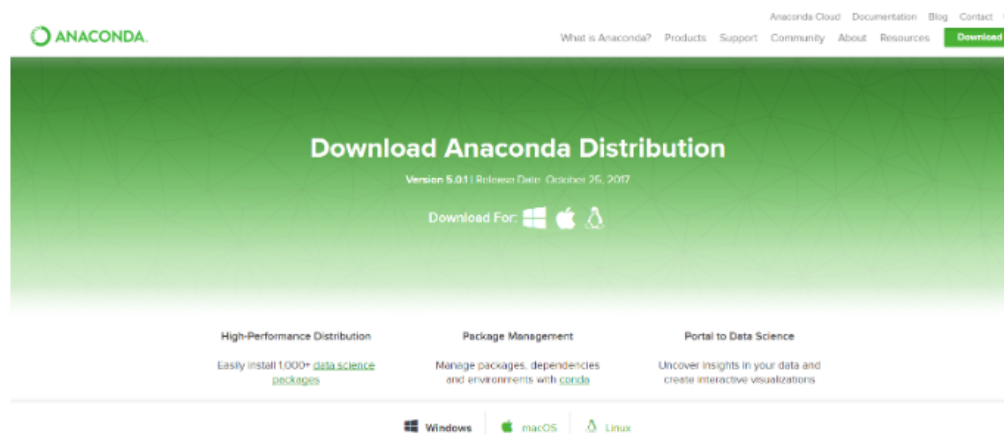
3.7 SOFTWARE DESCRIPTION

- ***Installing Anaconda on Windows***
- **DOWNLOADING AND INSTALLATION STEPS FOR Installing Anaconda on Windows**

Follow the steps below to install the Anaconda distribution of Python on Windows.

Steps:

1. [Download the Anaconda installer.](#)



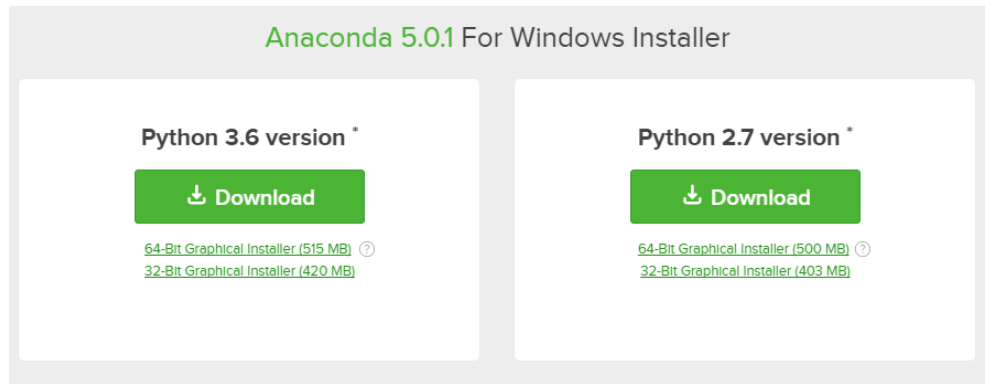
Select Windows

Select Windows where the three operating systems are listed.



Download

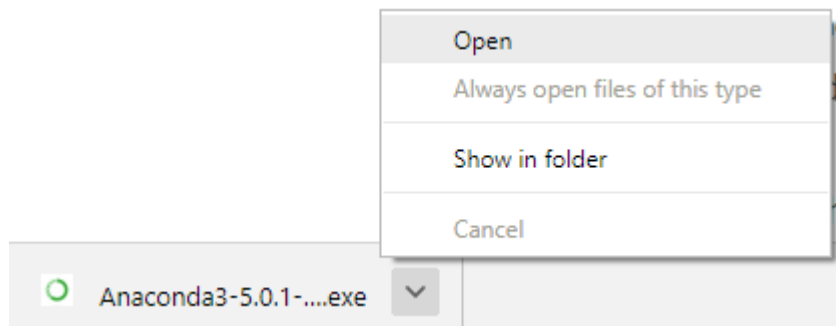
Download the most recent Python 3 release. At the time of writing, the most recent release was the Python 3.6 Version. Python 2.7 is legacy Python. For problem solvers, select the Python 3.6 version. If you are unsure if your computer is running a 64-bit or 32-bit version of Windows, select 64-bit as 64-bit Windows is most common.



2. Double click the installer to launch.

Open and run the installer

Once the download completes, open and run the .exe installer



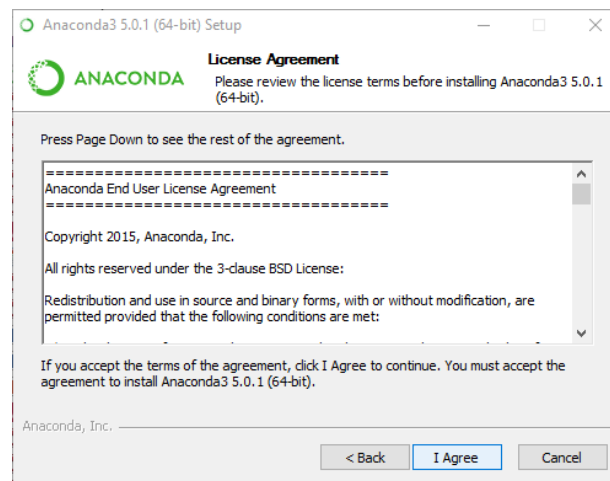
3. Click Next.

At the beginning of the install, you need to click **Next** to confirm the installation.



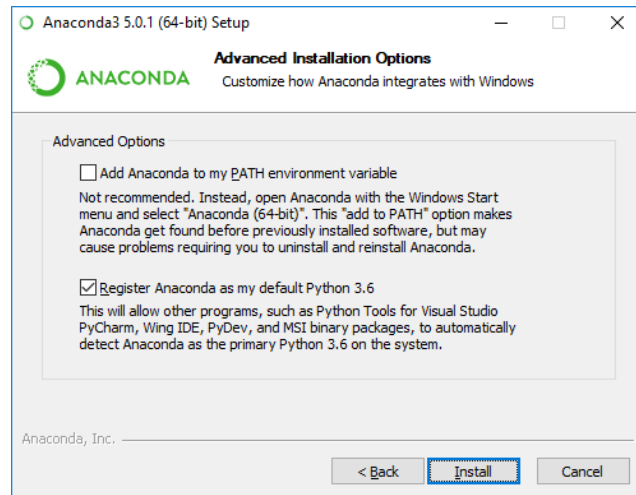
4. Read the licensing terms and click “I Agree”.

Then agree to the license.

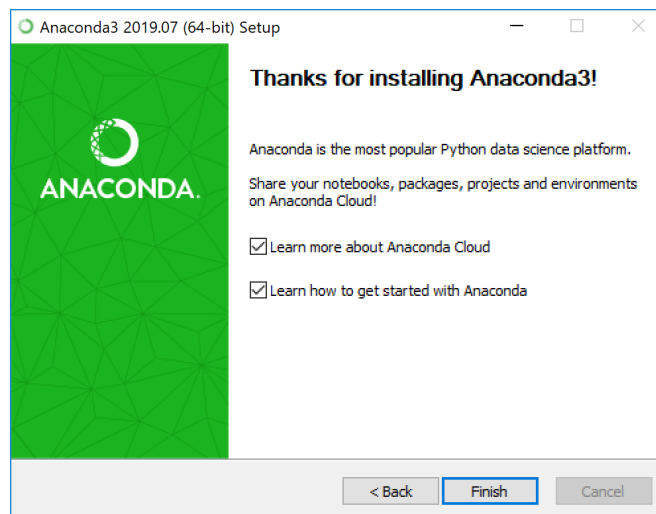


5. Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.

At the Advanced Installation Options screen, I recommend that you **do not check** “Add Anaconda to my PATH environment variable”



6. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:



If you wish to read more about Anaconda Cloud and how to get started with Anaconda, check the boxes “Learn more about Anaconda Cloud” and “Learn how to get started with Anaconda”. Click the Finish button.

- [Orange 3 App](#)
- [Rstudio](#)

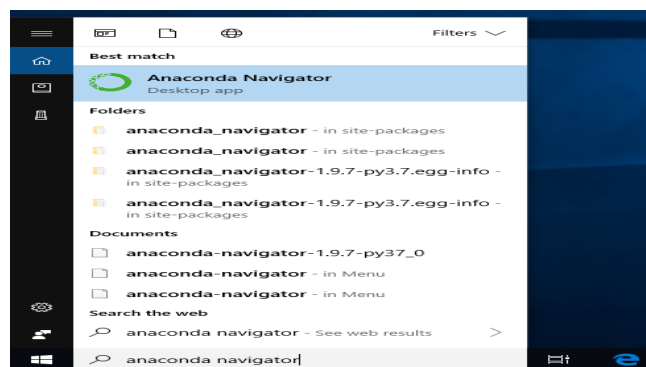
Verifying your installation: You can confirm that Anaconda is installed and working with Anaconda Navigator or conda.

Anaconda Navigator: Anaconda Navigator is a graphical user interface that is automatically installed with Anaconda. Navigator will open if the installation was successful. If Navigator does not open, review our help resources.

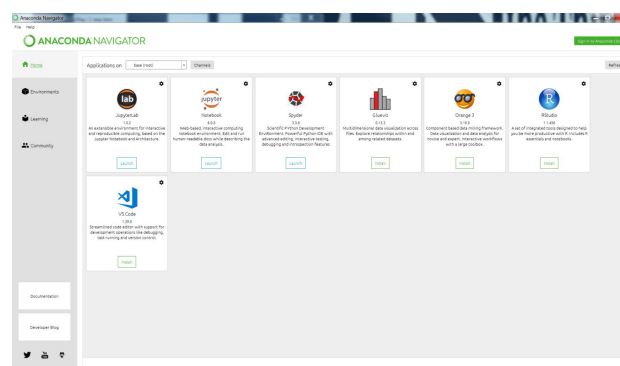
Windows: Click Start, search or select Anaconda Navigator from the menu.

MacOS: Click Launchpad, select Anaconda Navigator. Or, use Cmd+Space to open Spotlight Search and type “Navigator” to open the program.

Linux: See next section.



Getting started with Navigator



The following applications are available by default in Navigator:

- [JupyterLab](#)

- [Jupyter Notebook](#)
- [Spyder](#)
- [VSCode](#)
- [Glueviz](#)

3.8 LANGUAGES USED

3.8.1 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!

3.8.1.1 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

3.8.2 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.

3.8.2.1 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 Python 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.

3.8.2.2 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.

3.8.3 About 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.

3.9 SOFTWARE TESTING

3.9.1 General

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

3.9.2 TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

3.9.3 Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

(I) UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken into account after the individual unit is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

(II) INTEGRATION TESTING

These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

(III) FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

(IV) SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

(V) WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

(VI) BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

(VII) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER-4

RESULTS AND DISCUSSION

4.1 RESULTS

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.

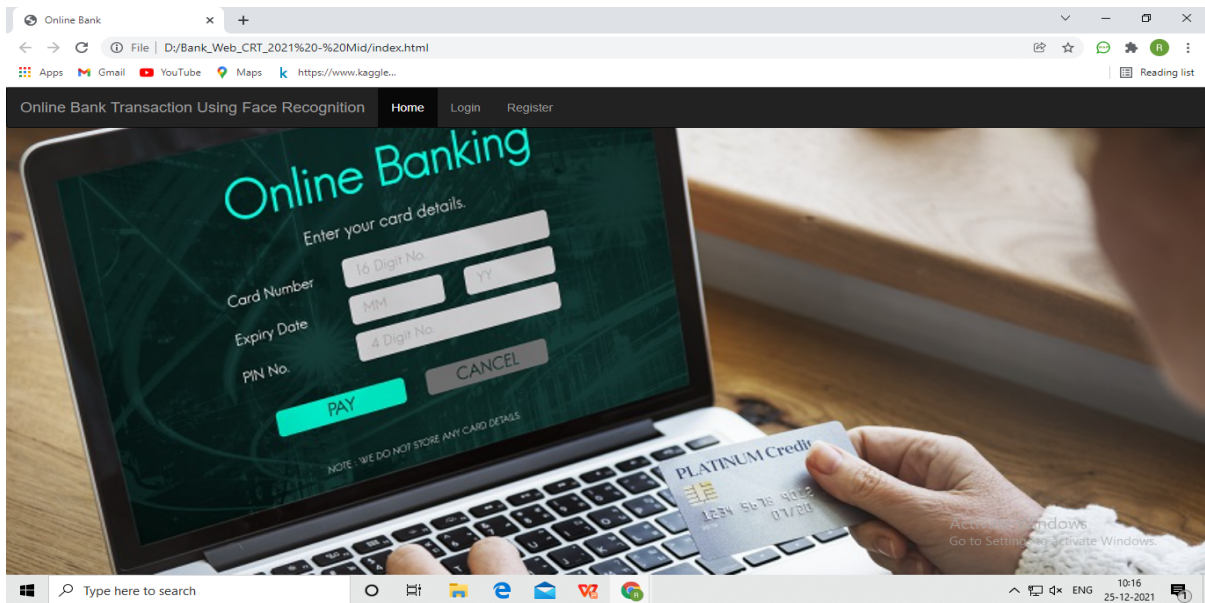


Fig 4.1: Home page

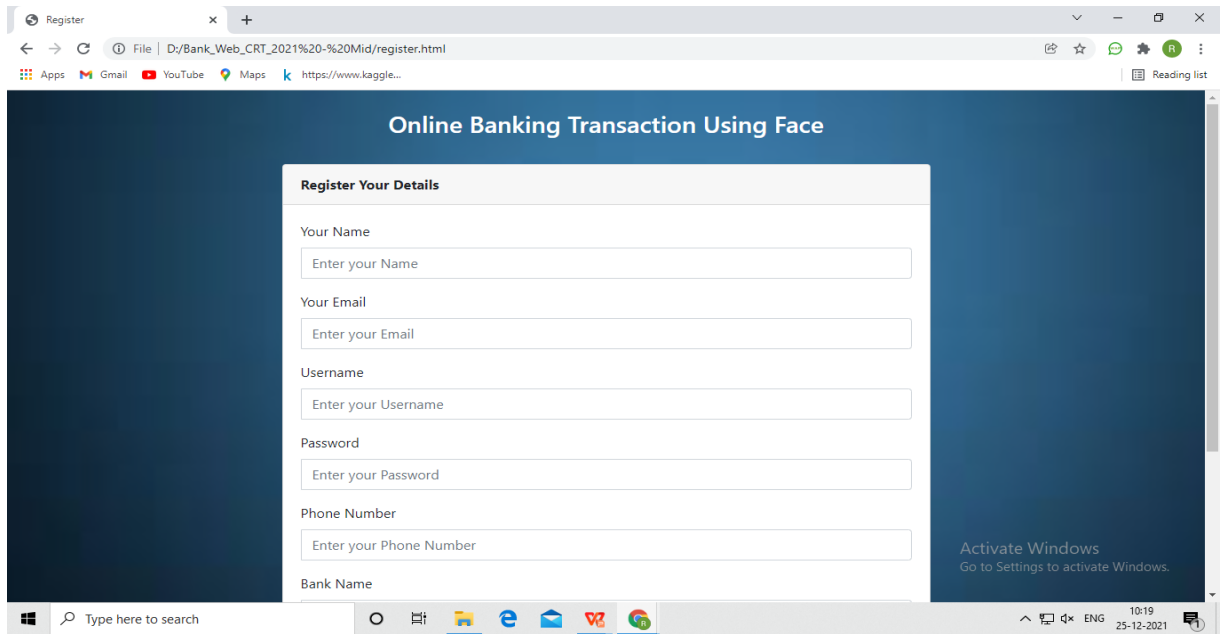


Fig 4.2: Registration Page

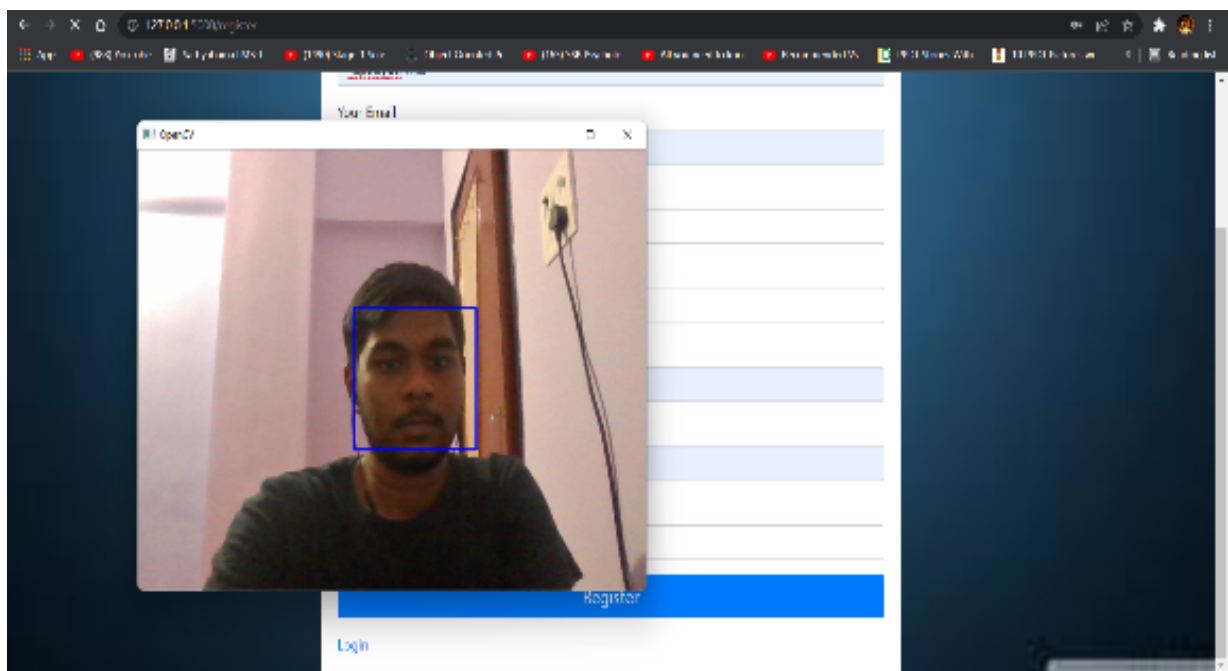


Fig 4.3: Face Recognition (Registration of user)

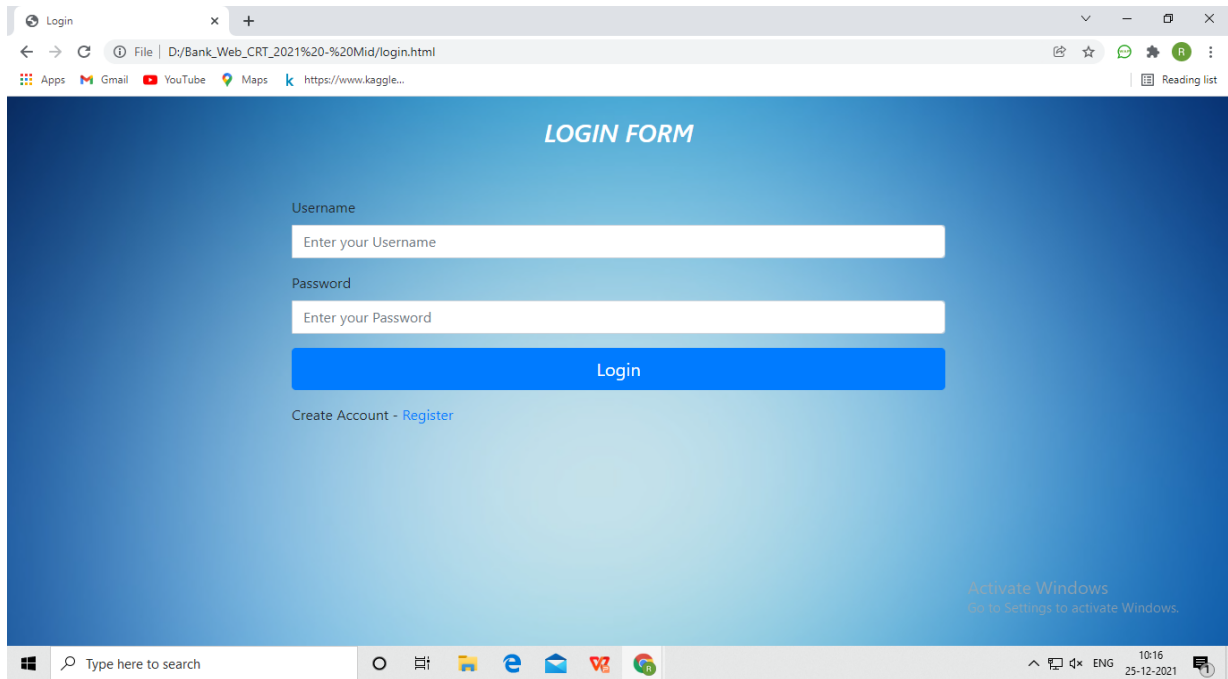


Fig 4.4: Login Page

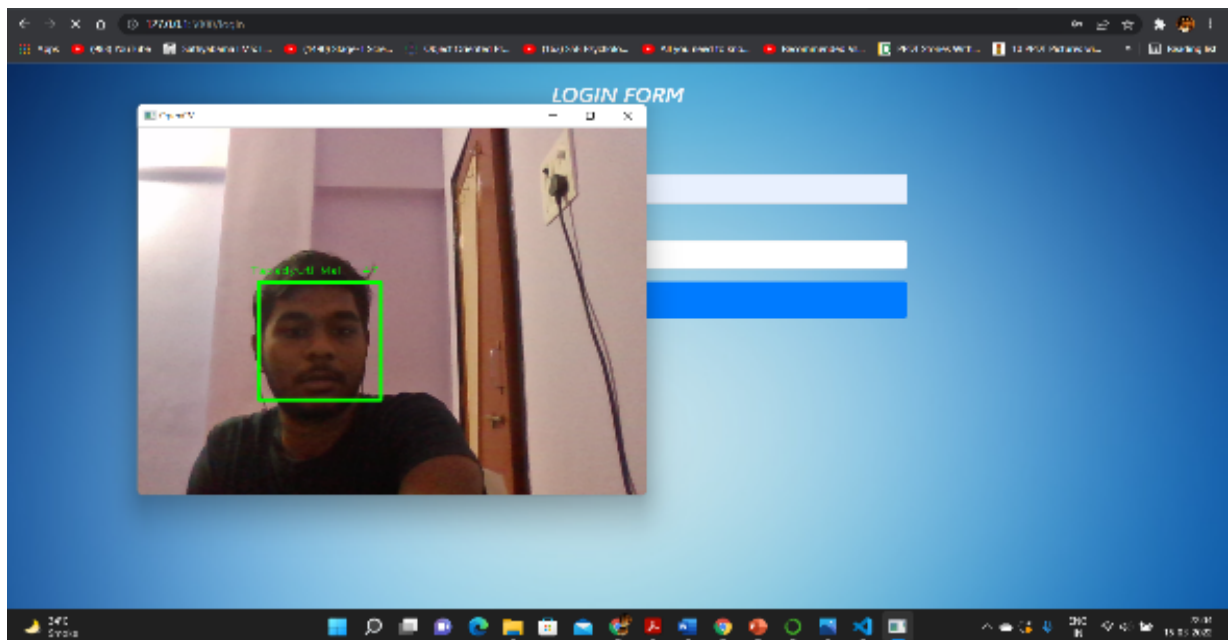


Fig 4.5: Face recognition (Logging in user)

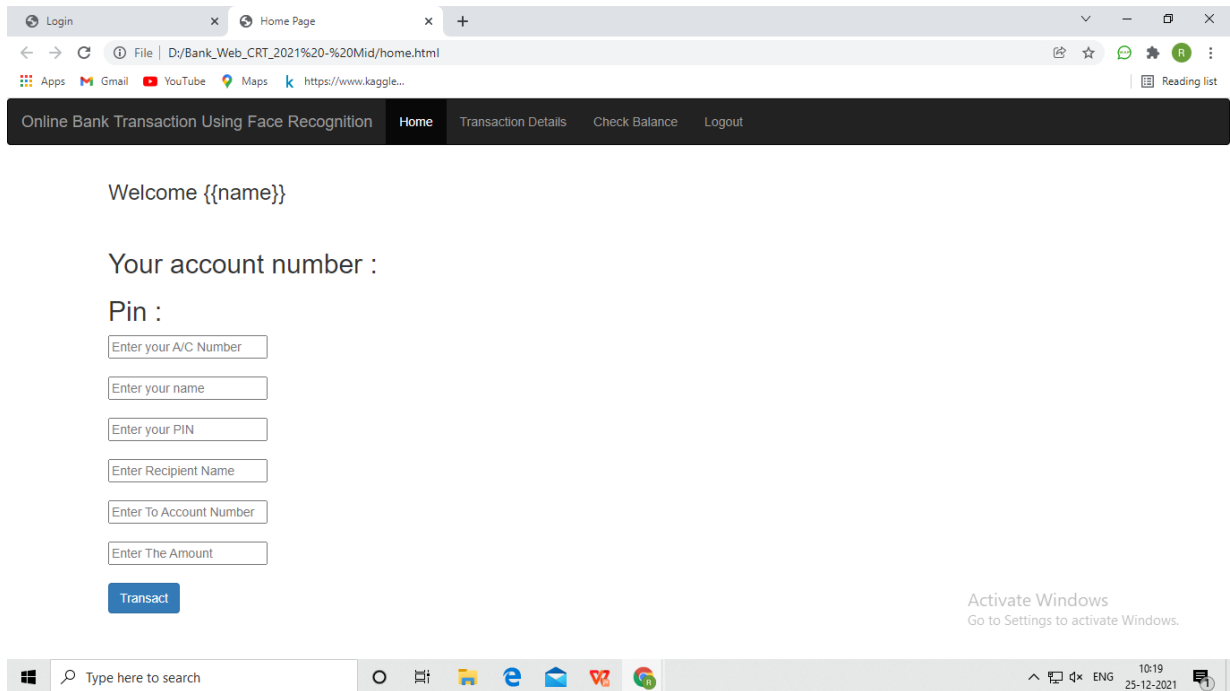


Fig 4.6: Transaction Page

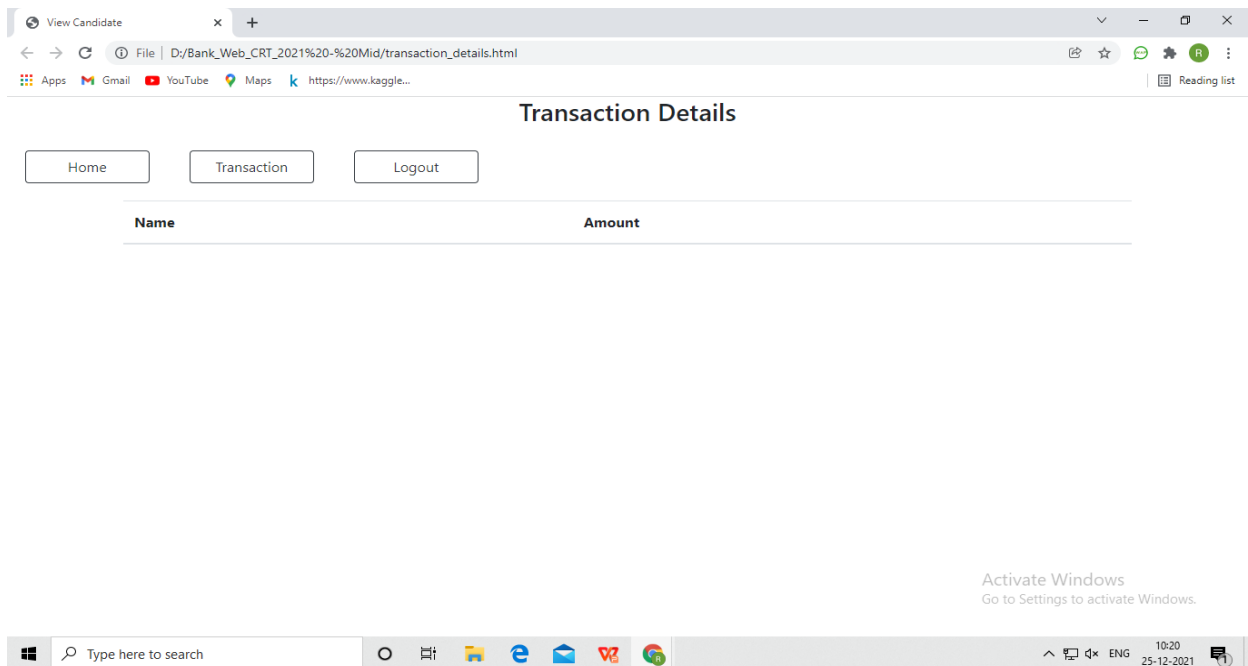


Fig 4.7: Transaction Details page of an user

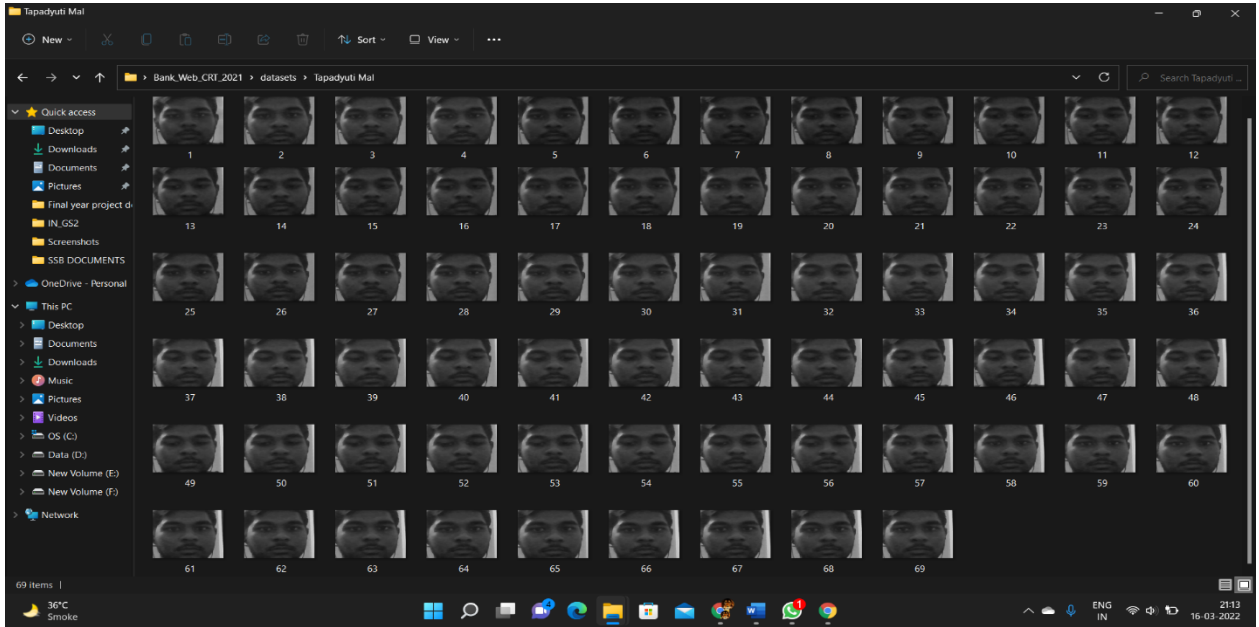


Fig 4.8: Data set collected

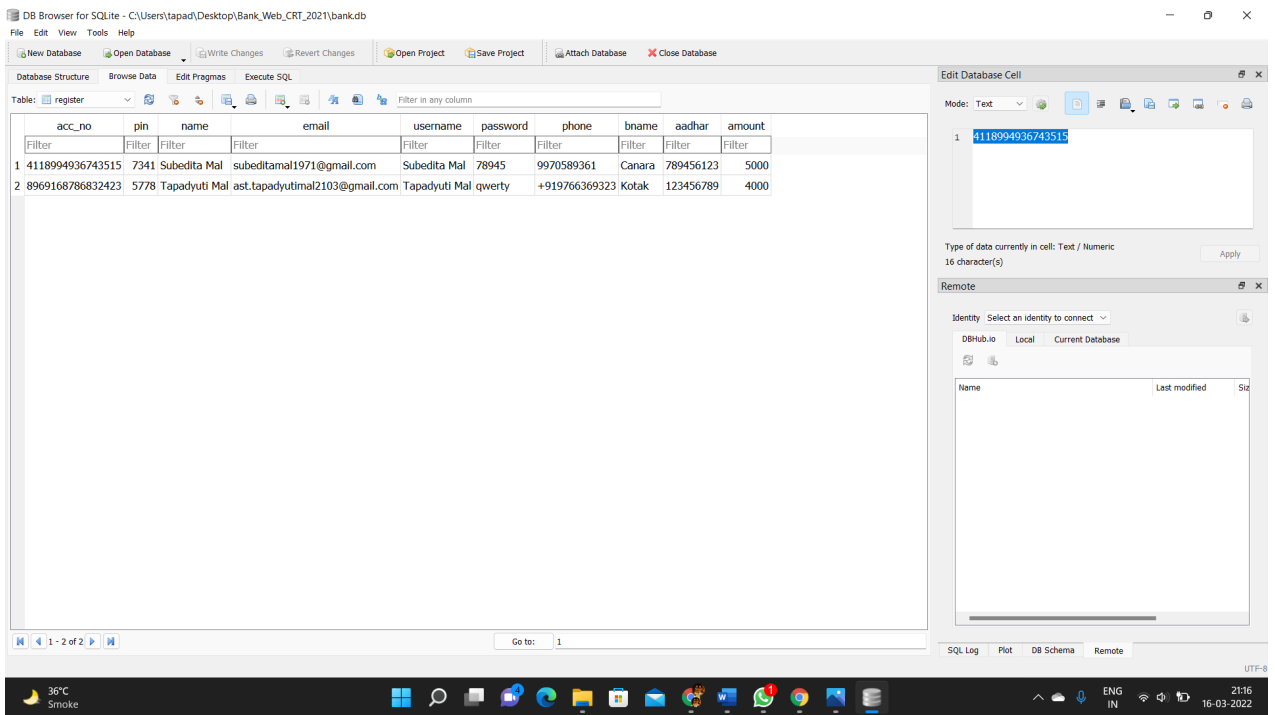


Fig 4.9: Registration Record (DB-Sqlite)

DB Browser for SQLite - C:\Users\tapad\Desktop\Bank_Web_CRT_2021\bank.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: Transaction1

	f_acc	f_name	r_name	to_acc	amount	date
1	2371706403763115	Tapadyuti Mal	Tapadyuti Mal	2371706403763115	1000	19/01/2022 10:12:47
2	9503011494153151	Subedita Mal	Tapadyuti Mal	2371706403763115	1000	19/01/2022 10:15:49
3	2371706403763115	Tapadyuti Mal	Subedita Mal	9503011494153151	2000	19/01/2022 10:19:14
4	2371706403763115	Tapadyuti Mal	Subedita Mal	9503011494153151	1000	19/01/2022 10:30:44
5	2371706403763115	Tapadyuti Mal	Subedita Mal	9503011494153151	1000	19/01/2022 19:13:58
6	2371706403763115	Tapadyuti Mal	Subedita Mal	9503011494153151	1000	29/01/2022 12:43:49
7	2371706403763115	Tapadyuti Mal	Tapadyuti Mal	2371706403763115	500	29/01/2022 12:45:33
8	8812684839104077	Tapadyuti Mal	Varun Vailbhav	8785309294952511	500	07/03/2022 14:26:12
9	8812684839104077	Tapadyuti Mal	Varun Vailbhav	8785309294952511	1000	07/03/2022 14:30:02
10	8812684839104077	Tapadyuti Mal	Tapadyuti Mal	8812684839104077	1000	08/03/2022 07:30:56
11	4118994936743515	Subedita Mal	Tapadyuti Mal	8812684839104077	1000	08/03/2022 07:50:06
12	5206859467656347	Tapadyuti Mal	Varun Vailbhav	2490942360351346	1000	08/03/2022 10:01:00
13	8969168786832423	Tapadyuti Mal	Subedita Mal	4118994936743515	1000	15/03/2022 22:06:11

Go to: 1

Opened 'C:\Users\tapad\Desktop\Bank_Web_CRT_2021\bank.db' from recent file list

36°C Smoke

SQL Log Plot DB Schema Remote

UTF-8

ENG IN 21:16 16-03-2022

Fig 4.10: Transaction Record (DB-Sqlite)

4.2 DISCUSSION

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.

CHAPTER-5

CONCLUSION AND FUTURE WORK

5.1 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.

5.2 FUTURE WORK

In Future any fraudulent access by the fake user is eliminated with the help of radio frequency identification card. Stopping bank robbers might be the first thing that comes to mind when one thinks about physical security in the banking industry. But that's only one thing that concerns banks and other financial institutions; they also have to protect their employees, branches, customers, and servers that contain customers' personal data. Preventing and investigating incidents is easier now, thanks to sophisticated video surveillance networks that are used to conduct real-time monitoring of bank lobbies, ATM machines, vaults and other sensitive branch areas, as well as to gather evidence for post-incident investigations.

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.Mulmule

APPENDIX

A. SOURCE CODE

Home.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>Home Page</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<nav class="navbar navbar-inverse">

  <div class="container-fluid">

    <div class="navbar-header">

      <a class="navbar-brand" href="#">Online Bank Transaction Using Face
Recognition </a>

    </div>

    <ul class="nav navbar-nav">

      <li class="active"><a href="#">Home</a></li>
```

```

<li><a href="{{url_for('transaction_details')}}">Transaction Details</a></li>
<li><a href="{{url_for('check_balance')}}">Check Balance</a></li>
<li><a href="/">Logout</a></li>
</ul>
</div>
</nav>

```

```

<div class="container">
    {% if authorized %}
        <h1>TRANSACTION SUCCESS</h1>
    {% elif unauthorized %}
        <h1>Unauthorized TRANSACTION</h1>
    {% elif Invalid %}
        <h1>Invalid TRANSACTION</h1>
    {% elif invalid_access %}
        <h1>Invalid Access</h1>
    {% endif %}
    {% if name %}
        <h3>Welcome {{name}}</h3><br>
        <h2>Your account number : {{acc_no}} and Pin : {{pin}}</h2>

        <form action="{{url_for('home')}}" method="POST">
            <input type="text" placeholder='Enter your A/C Number'
name='acc_no'><br><br>
            <input type="text" placeholder='Enter your name' name='name'><br><br>

```

```

        <input type="text" placeholder='Enter your PIN' name='pin'><br><br>
        <input type="text" placeholder='Enter Recipient Name'
name='r_name'><br><br>
        <input type="text" placeholder='Enter To Account Number'
name='to_acc'><br><br>
        <input type="text" placeholder='Enter The Amount' name='amount'><br><br>
        <input type="submit" class="btn btn-primary" value="Transact">
</form>

```

```

        {% endif %}
</div>

```

```

</body>
</html>

```

register copy.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

```

```
</head>
<body>
  <form action="{{url_for('register')}}" method="POST">
    <input type="text" placeholder="Enter Username" name='username'>
    <input type="password" placeholder="Enter Password" name='password'>
    <input type="text" placeholder="Enter Valid Email Address" name='email'>
    <input type="text" placeholder="Enter Mobile Number" name='phone'>
    <input type="text" placeholder="Enter Your Aadhar Number" name='aadhar'>
    <input type="submit" value="Register">

  </form>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Login</title>

  <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

  <script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
```

```

<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<style>
  body{
    background-image: url('static/images/2.png');
    background-size: cover;
    background-repeat: no-repeat;
  }
  h3{
    text-align: center;
    color: white;
  }
</style>

</head>

<body>
  {% if error %}
    <h1>you are unauthorized</h1>
  {% elif invalid %}
    <h1>your are invalid user</h1>
  {% endif %}
  <!-- <form action="{{url_for('login')}}" method="POST">
    <input type="text" placeholder="Enter Username" name='username'>
    <input type="password" placeholder="Enter Password" name='password'>
    <input type="submit" value="Login">
  </form> -->

  <br>

```

```

<h3><i>LOGIN FORM</i></h3>

<br><br>

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-8">

            <form class="form-horizontal" method="post"
action="{{url_for('login')}}">

                <div class="form-group">

                    <label for="username" class="cols-sm-2
control-label">Username</label>

                    <div class="cols-sm-10">

                        <div class="input-group">

                            <span class="input-group-addon"><i class="fa
fa-users fa" aria-hidden="true"></i></span>

                            <input type="text" class="form-control"
name="username" id="username" placeholder="Enter your Username" />

                        </div>

                    </div>

                </div>

                <div class="form-group">

                    <label for="password" class="cols-sm-2
control-label">Password</label>

                    <div class="cols-sm-10">

                        <div class="input-group">

                            <span class="input-group-addon"><i class="fa
fa-lock fa-lg" aria-hidden="true"></i></span>

```

```
        <input type="password" class="form-control"
name="password" id="password" placeholder="Enter your Password" />
```

```
    </div>
```

```
    </div>
```

```
</div>
```

```
    <div class="form-group ">
```

```
        <button type="submit" class="btn btn-primary btn-lg
btn-block login-button">Login</button>
```

```
    </div>
```

```
    <div class="login-register">
```

```
        Create Account - <a
href="{{url_for('register')}}">Register</a>
```

```
    </div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

index.html

```
<!DOCTYPE html>
```

```

<html lang="en">

<head>

  <title>Online Bank</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

  <style>

    body{

      background-image: url('static/images/1.png');

      background-repeat: no-repeat;

      background-size: cover;

    }

  </style>

</head>

<body>

<nav class="navbar navbar-inverse">

  <div class="container-fluid">

    <div class="navbar-header">

      <a class="navbar-brand" href="#">Online Bank Transaction Using Face
Recognition </a>

    </div>

```

```

<ul class="nav navbar-nav">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="/login">Login</a></li>
  <li><a href="/register">Register</a></li>
</ul>
</div>
</nav>

```

```

<div class="container">

</div>

```

```

</body>
</html>

```

check_balance.html

```

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtlkvYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/G
pGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxxhU9K/ScQsAP7hUibX39j7fakFP
skvXusvfa0b4Q" crossorigin="anonymous"></script>

```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUa
r5+76PVCmYI" crossorigin="anonymous"></script>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJISAwI
GgFAW/dAiS6JXm" crossorigin="anonymous">
```

```
<style>
```

```
body {
```

```
    font-family: Arial;
```

```
}
```

```
.main{
```

```
    display: grid;
```

```
    grid-template-columns: auto auto;
```

```
}
```

```
.image{
```

```
    height: 100vh;
```

```
    display: flex;
```

```
    align-items: center;
```

```
    justify-content: center;
```

```
}
```

```
.image img{
```

```
    width: 30vw;
```

```
}
```

```

.content{
    display:flex;
    align-items: center;
    justify-content: center;
}
.btn{
    width:10vw;
    margin:20px;

}
</style>
</head>
<body>
    <button type="button" class="btn btn-outline-dark">Home</button>
    <a href="{{url_for('home')}}" type="button" class="btn
btn-outline-dark">Transaction</a>
    <a href="{{url_for('login')}}" type="button" class="btn btn-outline-dark">Logout</a>

    <div class='main'>

    <div class="content">
        <center>
            <h1>Available Balance</h1>
            <br>
            <h4>Your A/C Number - {{acc}}</h4>

```

```

        <h3><p style="color: green;">Balance - {{res}}</p></h3>
    </center>
</div>

<div class="image">
    
</div>
</div>

</body>
</html>

```

app.py

```

from flask import *
from random import randint
from face_detection import *
from face_reg import *
import sqlite3 as sql
from datetime import datetime
app = Flask(__name__)
app.secret_key="panda"
@app.route('/')
def index():
    return render_template("index.html")

```

```

@app.route('/register',methods=['POST','GET'])
def register():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        username = request.form['username']
        password = request.form['password']
        phone = request.form['phone']
        bname = request.form['bname']
        aadhar = request.form['aadhar']
        def random_with_N_digits(n):
            range_start = 10**(n-1)
            range_end = (10**n)-1
            return randint(range_start, range_end)
        a=random_with_N_digits(16)
        pin = random_with_N_digits(4)
        print(pin)
        get(username)
        con = sql.connect('bank.db')
        cur =con.cursor()
        cur.execute('insert into
register(acc_no,pin,name,email,username,password,phone,bname,aadhar,amount)
values(?,?,?,?,?,?,?,?,?,),(a,pin,name,email,username,password,phone,bname,aad
har,5000))
        con.commit()
        return redirect(url_for('login'))

```

```

    return render_template('register.html')

@app.route('/login',methods=['POST','GET'])
def login():
    if request.method == 'POST':
        name = request.form['username']
        print(name)
        password = request.form['password']
        con = sql.connect('bank.db')
        cur = con.cursor()

        cur.execute('select username,password,acc_no,pin from register where
username=? and password=?',(name,password))

        data = list(cur.fetchone())

        p = data[3]
        ano = data[2]

        print(data)
        con.commit()
        ac = face_reg(name)
        print(ac)
        if ac == 'Unauthorized':
            return render_template('login.html',error='error')
        elif ac == 'Authorized':
            session['name'] =name
            session['pin'] = p

```

```

    session['ano'] = ano
    print(type(session['ano']))
    print(type(session['pin']))
    print('auth')

    return render_template('home.html',name=name,acc_no=data[2],pin=data[3] )

elif ac=='Invalid User':
    return render_template('login.html',invalid='invalid')

return render_template('login.html')

```

```
@app.route('/home',methods=['POST','GET'])
```

```
def home():
```

```

    if request.method == 'POST':
        acc_no = request.form['acc_no']
        f_name = request.form['name']
        session['f_name'] = f_name
        r_name = request.form['r_name']
        session['r_name'] = r_name
        to_acc = request.form['to_acc']
        amount = request.form['amount']
        pin = request.form['pin']
        now = datetime.now()
        date = now.strftime("%d/%m/%Y %H:%M:%S")
        ac = face_reg(f_name)
        print("ac = ",ac)
        print(type(acc_no))

```

```

print(type(pin))

# con = sql.connect('bank.db')

# cur = con.cursor()

# res = cur.execute("select acc_no,pin from register where acc_no =?",(acc_no))

# con.commit()

# print("res= ",res)

cd = (str(session['ano'])).strip()

xy = (str(session['pin'])).strip()

ab = acc_no.strip()

lm = pin.strip()

print(type(cd))

if ac == 'Authorized':

    if cd == ab and xy == lm:

        con = sql.connect('bank.db')

        cur1 = con.cursor()

        cur1.execute("insert into Transaction1
(f_acc,f_name,r_name,to_acc,amount,date) values
(?,?,?,?,?,?)",(acc_no,f_name,r_name,to_acc,amount,date))

        cur2 = con.cursor()

        cur2.execute("update register set amount=amount-? where
acc_no=?", (amount,acc_no))

        cur3 = con.cursor()

        cur3.execute("update register set amount=amount+? where
acc_no=?", (amount,to_acc))

        con.commit()

        return render_template('home.html',authorized='authorized')

    else:

```

```

        return render_template('home.html',invalid_access='invalid_access')
    elif ac == 'Unauthorized':
        return render_template('home.html',unauthorized='unauthorized')
    elif ac == 'Invalid User':
        return render_template('home.html',Invalid='Invalid User')
    return render_template('home.html')
else:
    return
render_template('home.html',name=session['name'],acc_no=session['ano'],pin=session['pin'])

```

```

@app.route("/check_balance")

```

```

def check_balance():

```

```

    balance = str(session['ano'])

```

```

    con = sql.connect("bank.db")

```

```

    # con.row_factory = sql.Row

```

```

    cur = con.cursor()

```

```

    cur.execute("select amount from register where acc_no = ?",(balance,))

```

```

    res = cur.fetchone()

```

```

    con.commit()

```

```

    print(res)

```

```

    am = float(res[0])

```

```

    print(balance)

```

```

    return render_template("check_balance.html",res=am,acc=balance)

```

```

@app.route("/transaction_details")

```

```

def transaction_details():
    ac = session['ano']
    con = sql.connect("bank.db")
    cur2 = con.cursor()
    cur2.execute("select f_name,r_name,amount from Transaction1 where to_acc = ?
or f_acc = ?",(ac,ac,))
    res2 = cur2.fetchall()
    con.commit()
    f_name = session['name']
    # r_name = session['r_name']
    det = []
    for i in res2:
        if i[1] == f_name:
            det.append([i[0],i[2],'credited'])
        elif i[0] == f_name:
            det.append([i[1],i[2],'debited'])
    print(i)
    print(det)
    return render_template("transaction_details.html",res = det)

if __name__ == '__main__':
    app.run(debug=True)

# face_detection.py
import cv2, sys, numpy, os

def get(username):

```

```

haar_file = "haarcascade_frontalface_alt2.xml"
datasets = './datasets'
sub_data= username
path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
    os.mkdir(path)
(width, height) = (130, 100)
face_cascade=cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)
count = 1
while count<70:
    (_, 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()
```

facereg.py

```
import cv2, sys, numpy, os

def face_reg(name):

    size = 4

    haar_file ="haarcascade_frontalface_alt2.xml"

    datasets = './datasets'

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

    (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)

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

    print(images)
```

```

model = cv2.face.LBPHFaceRecognizer_create()
model.train(images, lables)
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

su=1
er=1

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))
        prediction = model.predict(face_resize)
        cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 3)
        if prediction[1]<70:
            cv2.putText(im, '% s - %.0f' % (names[prediction[0]], prediction[1]), (x-10,
y-10), cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0))
            su+=1
            if su==30:
                if names[prediction[0]] == name :
                    return 'Authorized'
                else:
                    return 'Invalid User'

```

```
    else:
        cv2.putText(im, 'not recognized', (x-10, y-10),
cv2.FONT_HERSHEY_PLAIN, 1, (0, 255, 0))
        er+=1
        if er==30:
            return 'Unauthorized'
cv2.imshow('OpenCV', im)
key = cv2.waitKey(10)
if key == 27:
    break
webcam.release()
cv2.destroyAllWindows()
```

B. PUBLICATION WITH PLAGIARISM REPORT

TranSecure (Multi-Level Secure Transaction using Facial Authentication)

Varun Vaibhav , Tapadyuti Mal , P. Asha

**UG Student, Department of Computer Science,
Sathyabama Institute of Science and Technology, Chennai, 600119, India**

**Dr. P. Asha, M.E., Ph.d., Department of Computer Science,
Sathyabama Institute of Science and Technology, Chennai, 600119, India**

ABSTRACT: - Security and Assurance are an integral part of any industry. In real time, a person's facial recognition can be done in two stages such as, facial recognition and facial recognition. This paper uses the "Haar-Cascade algorithm" to identify human faces edited with Open CV in Python language and "Binary area pattern algorithm" for face recognition, active feature selection and low variation of good non-fiction features. Haar-based cascade class system uses only 200 of the 6000 features to produce a visibility rate of 85-95%.

INDEX TERMS: - face recognition, Haar – Cascade, Open CV, Authentication system, recognition rate, online payment, security, image verification, credit card.

INTRODUCTION

We have observed an inclination in the use of technology in various sectors of business over the last decade to facilitate and better customers. This is especially true in the banking and finance industries. Face recognition has gained a foothold in touch and text-based authenticity since the advent of digital transformation due to the convenience it provides without compromising transaction security. As a result of the billions of dollars lost by major banks, there has been a call for a change in facial biometric recognition to address the problem. The goal is to validate ownership and only allow the activity to proceed if the account owner's identity is correctly identified. KYC refers to the process of

verifying a customer's ID (Know Your Customer). Because some people are hesitant to use online transactions in today's world of advanced technology, Internet gamblers can easily obtain personal information from users. When it comes to digital payments, security is a top priority. As a result, we propose a strategy to improve online job security by implementing a two-step verification process: OTP authentication followed by Face Recognition.

The plan that we will propose will attempt to reduce the number of attacks during digital payment transactions. Lost or stolen cards, account confiscation, fraudulent cards, fraudulent applications, multiple stamps, and affiliate merchants all pose a

risk to online shopping. In the case of a bankruptcy account, the cardholder unknowingly provides his or her bank account information, which the fraudster uses to create a duplicate card containing that information. A user card is also merged and used by a fraudster in a fraudulent case. When the same work is recorded more than once, multiple stamps are produced. Reseller employees collaborate with fraudsters for affiliate marketers. By capturing and verifying the real-time image of cardholders, the proposed system successfully reduces all of these scams.

LITERATURE SURVEY

1.Chinese Consumers and Facial-Recognition Payment

The introduction and widespread use of face-to-face transaction technology introduced recent challenges. However plastic are faster and convenient, it is simple to misplace a card can be shared and hacked because people use easy passwords and make use of them across multiple accounts and services. QR charges cannot be separated from android/ iOS phones, they may be lost, signals may not be stable, and cells may become depleted.

Face recognition technology, on the other hand, scans and interprets feature vectors without physically being in contact, working, and accuracy. Face-to-face news studies using face authentication technology and face identification payment systems are currently very popular. After lots of studies have been conducted that focus on the operating system in face recognition system. Non-technical research on face-to-face payment features and factors influencing the purpose of consumer use, on the other hand, is uncommon. As a result, the goal of this research was to look

into the factors that influence consumers' willingness to use face recognition systems.

2. Payment System Based on Biometric Face Recognition

Payment cards are increasingly being used in a variety of settings, including shopping, restaurants, lodging, and online payments for hotel reservations, movie tickets, plane and train tickets, and so on. So the issue is that one must carry payment cards with them and keep them safe so that they can be used at all times.

This, too, was lacking in security. Biometric face recognition payments are used for all types of payments in the current work. As a result, it eliminates the need to remember multiple passwords. The face recognition payment system is safe, secure, and simple to use.

When compared to other payment technologies, it is dependable and extremely efficient. The standard design of an online payment system based on facial recognition is proposed. Face detection methods include detecting eigenfaces and Euclidean distance.

3. Facial Recognition in Banking: Current Applications

Face recognition software is useful, with consumer apps like the capability to unlock a personals android/iOS phones with their own face. As machine learning is now a hot topic in the business world in the first decade, the banking sector was at the forefront of AI business acquisition; it stands to reason that face recognition technology will begin to enter the bank. There are a number of companies that offer face identification software to banks and appear to have the required expertise for their C-suite, which we look at when evaluating a company's claims to use AI. Face recognition is one of many ways that

banks can reduce customer conflict and increase efficiency and accessibility. Some experts believe that this is how banks will succeed in the future, as AI and other technologies make many services available without any downtime.

4. Face detection method based on skin colour:

In colour photography, acquiring skin colour is a very popular and useful way to get a face. A person's face is incomplete without colour. There are some advantages to using skin colour as a face tracking feature. Color processing is significantly faster than other facial features. Each pixel was classified as skin or non-skin in the skin colour acquisition process based on its colour components. In situations where colour definition is critical, the HSV colour model is frequently preferred over the RGB colour model. A different colour space has pixels with different widths to represent the skin and non-skin regions.

5. Using the VJ Algorithm to combine skin color-based classifiers and the HAAR feature

In his shadowy images, D. Gobinathan used a variety of facial expressions. Viola and Jones (VJ) developed a hybrid-based HAAR compactor for grayscale images that was compatible with the skin colour filtering system, which provides consistent facts in shadow images. The image was sent in the form of a HAAR Feature-based detector that had been modified so that it eventually operates in its ROC curve with a low number of missed people but a high number of false detection. Their suggested feature features reduce the number of these erroneous discoveries. To minimise light effects, they also employ a color - correction algorithm. Their practical results for the Bao shade face website revealed that

this method outperformed the original Viola Jonas algorithm.

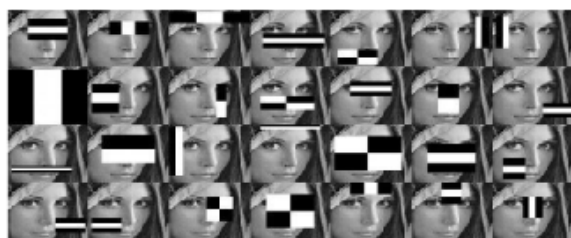
PROPOSED WORK

Verification is defined as the process or act of verifying a user's identity or process. Proof of authenticity computer programming methods are divided into three categories: knowledge-based verification (e.g., password, PIN), administrative-based authentication (e.g., memory card, smart card tokens), and biometric-based authentication (physical or moral). This information-based verification, which is commonly used for mobile devices, is based on unique and private information that is expected to be known only to the user. Object-based authentication is based on the competence of something tangible.



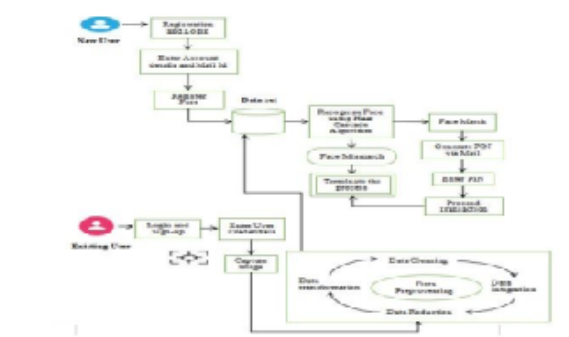
Handwriting, clicks, touch screen, navigation, signature, voice, and behaviour find are examples of behavioural biometrics based on user-readable navigation. All of these methods enable users analysis device interaction and model building, which decides whether or not to authenticate the current user.

The study looked into ongoing AI-based authentication solutions proposed for IoT applications, while the survey provide detailed analysis of the most commonly used methods sensors embedded in smartphones.



Another review paper evaluated book subjects based on the type and size of data collected, classes used for identification, and results discovered. It is worth noting that these papers look into the use of visual biometrics for verification. In this paper, we concentrate on studies that employ behavioural biometrics. We are unable to cover all relevant studies in this paper, and as previously stated, there are numerous research papers on the subject.

SYSTEM ARCHITECTURE



The three main procedures followed in the face recognition program Discovery, Feature Disposal and Comparison.

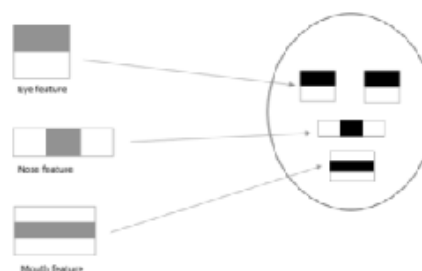
Face detection:

The Viola Jones algorithm is a good method for detecting faces. In general, this algorithm is not limited to facial detection and can be used for a variety of robust

systematic acquisition tasks. The Viola-Jones algorithm is comprised of three key concepts that allow for the development of a real-time face finder: Haar, Image Essentials. Using these features, the system can determine whether or not a person's face is present.

Haar like features:

The Haar cascade classifier detects human faces by using features such as Haar. There are three ways to create Haar-like features. The first type is an edge element, the second is a line element, and the third is a rectangular element. The Haar goal will provide a quick calculation using an important image. It is referred to as Haar features. The algorithm searches the face for a specific haar feature. This acquisition takes a picture, converts it to a 24X24 window, and smears each Haar element pixel by pixel on that window. To train the separator, the algorithm initially requires a large number of good images (facial images) and negative images (facial images).



Types of Haar-like features

Following that, these features are removed. The numerical values are determined by the images used to distinguish one image from another by the one-dimensional element obtained by subtracting the total number of pixels under a white rectangle on the underside of a black rectangle.

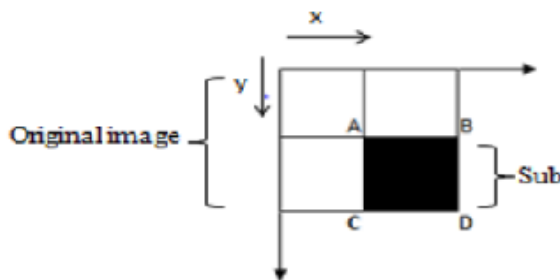
Feature = Σ (pixels in dark) - Σ (pixels in white)

Dark

White

A number of features are counted for each kernel's possible size and location. A 24 hour window yields over 160,000 features. For each aspect calculation, the total number of pixels beneath the white and black rectangles must be determined. To solve this, a key image concept and the adaboost algorithm are used, which reduces 160000 features to 6000.

Integral Image: Rectangular features can be quickly determined using a central image representation known as Complete Image. The complete image includes representations of the smaller units of a given image.



$$\text{Shaded region} = D - (B + C) + A$$

Integral image schematic diagram

As the intersection starts from top left to right down, the image is compacted at the performance of a few pixels. This enables the addition of all pixels within any rectangular point to be calculated using only four values. These pixels correspond to the edges of a rectangle in the input image in the final image.

RESULTS AND DISCUSSION

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.

OUTPUT SCREENSHOTS



Fig 1: Home page



Fig 2: Registration Page

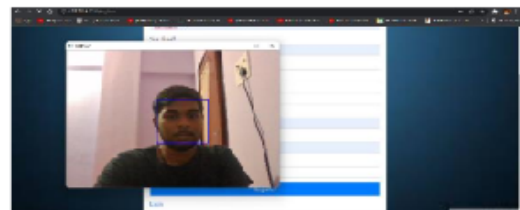


Fig 2.1: Face Recognition (Registration of user)

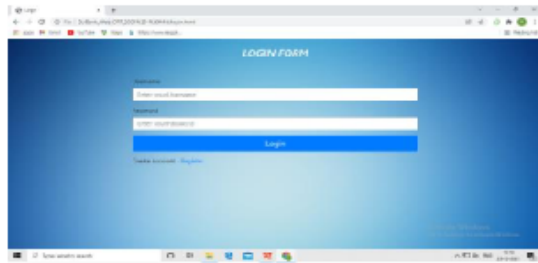


Fig 3: Login Page

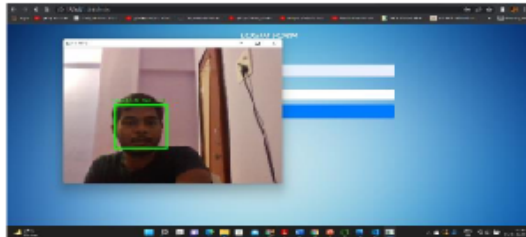


Fig 3.1: Face recognition (Logging in user)

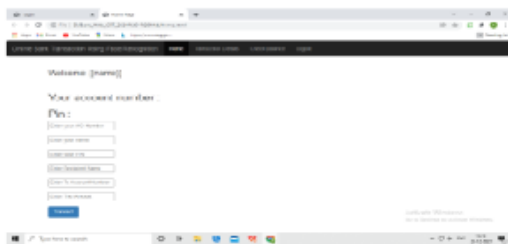


Fig 4: Transaction Page

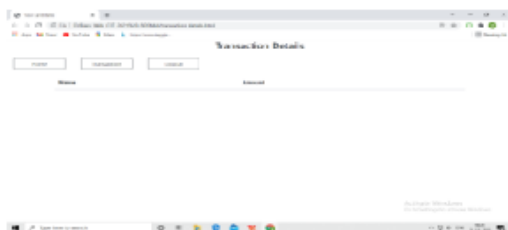


Fig 5: Transaction Details page of an user

CONCLUSION

In this project we are going to develop a robust automated web application for transacting money in higher level security

purpose with high facial recognition. First we have to register our personal details with our face. It will be considered as datasets now our system can understand who are all authorized and unauthorized. While login our face will be recognized if we authorized will can move to authorized page else our system won't allow to login. For higher security purpose we have used face recognition module. During the transaction our face will be recognized it will allow only the authorized account holder to transact a money, it doesn't allow others to transact a money.

REFERENCES

1.Chinese Consumers and Facial-Recognition Payment

Wen Kun Zhang and Min Jung Kang are the authors of this paper, which was published in the journal IEEE Access in 2019.

2. Payment System Based on Biometric Face Recognition

Surekha, R. Gondkar, and Saurab Gondkar are the authors. Dr. C. S. Mala International Journal of Engineering Research & Technology NCFSC - 2018 Conference Proceedings

3. Facial Recognition in Banking: Current Applications

Author: Niccolo Mejia,2019 Conference Proceedings

4. Face detection method based on skin colour:

Authors: Devendra Singh Raghuvanshi, Dheeraj Agrawal

5. Using the VJ Algorithm to combine skin color-based classifiers and the HAAR feature

Authors: N.Gobinathan, Abinaya, and Geetha contributed to this work.

PLAGIARISM REPORT

TranSecure_ResearchPaper

ORIGINALITY REPORT

13%

SIMILARITY INDEX

5%

INTERNET SOURCES

7%

PUBLICATIONS

4%

STUDENT PAPERS
