VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

A Project Report on

DESIGN AND IMPLEMENTATION OF EYE PUPIL MOVEMENT BASED PIN AUTHENTICATION SYSTEM

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by

SUMAN R KULKARNI	1RN17IS099	T KISHORE	1RN17IS103
V HARSHITHA	1RN17IS111	VIBHA S NAVALE	1RN17IS115

Under the Guidance of

Mrs. Kavyashree K

Assistant Professor

Department of ISE

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhana Road, Rajarajeshwari Nagar post, Channasandra, Bengaluru-560098

2020-2021

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhana Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled ***Design and Implementation of Eye Pupil Movement Based PIN Authentication System*** has been successfully completed by **Suman R Kulkarni (1RN17IS099), T Kishore (1RN17IS103), V Harshitha (1RN17IS111) and Vibha S Navale (1RN17IS115)**, bonafide students of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year **2020-2021**. The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mrs. Kavyashree K	**Dr. Suresh L**	**Dr. M K Venkatesha**
Project Guide	Professor and HoD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

We, **SUMAN R KULKARNI [USN:1RN17IS099], T KISHORE [USN: 1RN17IS103], V HARSHITHA [USN: 1RN17IS111], VIBHA S NAVALE [USN: 1RN17IS115]** students of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project work entitled ***Design and Implementation of Eye Pupil Movement Based PIN Authentication System*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belgaum* during academic year 2020-2021.

Place : Bengaluru

Date :

SUMAN R KULKARNI (1RN17IS099)

T KISHORE (1RN17IS103)

V HARSHITHA (1RN17IS111)

VIBHA S NAVALE (1RN17IS115)

ABSTRACT

Design and Implementation of Eye Pupil Movement Based PIN authentication system is carried out because of the vulnerable attacks that might be caused due to an authorised user entering a PIN in public places making it easy for attacks such as shoulder surfing (observation user while typing the password through the keyboard), acoustics keyboard eavesdropping, thermal tracking and screen electromagnetic emanations.

Hence to prevent these issues, eye tracking PIN authentication system is a method to conserve the security of the system and also the natural interaction method which is based on eye movement tracking which provides a promising solution to the system security and usability since PIN authentication with hands-off gaze-based PIN entry techniques leaves no physical footprints behind and therefore offer a more secure password entry option.

The purpose of this work is to enter and identify gaze-based PINs using a smart camera through real-time eye detection and tracking. NI Vision Builder and Lab VIEW are used for eye tracking and for recording eye center location on board the camera real time.

First the detection of the face and eye is done and then after the detection it is used to determine whether the eye is open or closed which is achieved by eye blink detection. The smart camera allows on-board data processing and collection. Non-contact PIN based authentication adds a layer of security to physical PIN entries and is expected to reduce the vulnerability of the authentication process.

ACKNOWLEDGMENT

At the very onset we would like to place our gratefulness to all those people who have helped us in making the Final Year Project a successful one.

Coming up with this topic was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided us in the right direction.

First of all, we would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of Final Year Project.

In this regard, we express sincere gratitude to the Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information Science and Engineering, **Dr. Suresh L**, for having accepted to patronize us in the right direction with all his wisdom.

We place our heartfelt thanks to **Mrs. Kavyashree K**, Assistant Professor, Department of Information Science and Engineering for having guided us and all the staff members of the department of Information Science and Engineering for helping at all times.

We also thank our Project coordinators **Dr. Prakasha S and Mrs. Kusuma S**, Assistant Professors, Department of Information Science and Engineering. We would thank our friends for having supported us with all their strength and might.

Last but not the least, we thank our respective parents for supporting and encouraging us throughout. We have made an honest effort in this Final Year Project.

SUMAN R KULKARNI (1RN17IS099)

T KISHORE (1RN17IS103)

V HARSHITHA (1RN17IS111)

VIBHA S NAVALE (1RN17IS115)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

OCR	Optical Character Recognition
PIN	Personal Identification Numbers
HOG	Histogram of Oriented Gradients
ATM	Automated Teller Machines
LED	Light Emitting Diode
DFD	Data Flow Diagram
HLD	High-Level Design
LLD	Low-Level Design
LLDD	Low-Level Design Document

Chapter 1

INTRODUCTION

1.1 Background

Artificial Intelligence

Artificial Intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals. In computer science AI research is defined as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), autonomously operating cars, intelligent routing etc. Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding.

For most of its history, AI research has been divided into subfields that often fail to communicate with each other. These sub-fields are based on technical considerations, such as particular goals (e.g. "robotics" or "machine learning"), the use of particular tools ("logic" or artificial neural networks), or deep philosophical differences. Subfields have also been based on social factors (particular institutions or the work of particular researchers).

The traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the objects. General intelligence is among the field's long-term goals. Approaches include statistical methods, computational intelligence, and traditional symbolic AI. Many tools are used in AI, including versions of search and mathematical optimization, artificial neural networks, and methods based on statistics, probability and economics.

The AI field draws upon computer science, mathematics, psychology, linguistics, philosophy and many others. The field was founded on the claim that human intelligence "can be so precisely described that a machine can be made to simulate it".

This raises philosophical arguments about the nature of the mind and the ethics of creating artificial beings endowed with human-like intelligence which are issues that have been explored by myth, fiction and philosophy since antiquity. Some people also consider AI to be a danger to humanity if it progresses unabated.

Others believe that AI, unlike previous technological revolutions, will create a risk of mass unemployment. In the twenty-first century, AI techniques have experienced a resurgence following concurrent advances in computer power, large amounts of data, and theoretical understanding. AI techniques have become an essential part of the technology industry, helping to solve many challenging problems in computer science, software engineering and operations research.

In simple terms, AI aims to extend and augment the capacity and efficiency of mankind in tasks of remaking nature and governing the society through intelligent machines, with the final goal of realizing a society where people and machines coexist harmoniously together.

Machine Learning

Machine learning is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed. The name machine learning was coined in 1959 by Arthur Samuel.

Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank, and computer vision.

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making using computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

Introduction to the project

Human-Computer Interaction (HCI) is focused on the joint performance of tasks between humans and computers and how they communicate. There are two components in this exchange of information, Input and Output. Input components identify and sense the user's desired task and information to be communicated to the computer. There are two types of Input methods, Direct and Indirect. The Indirect input refers to the method where the actions of the user will translate into data or commands to be entered into a system. The Indirect input methods have been around since the beginning of Web development and they are still usable. Keyboard and Mouse are prime examples of indirect interaction methods. The direct input refers to the devices which have no intermediary and the movement of the user's body is equal to the input to the system. The Direct input methods have been introduced to facilitate users to have a natural feeling of interaction, for example, touch interaction.

Eye tracking had been introduced as a way of direct interaction which could pave the way for new technologies and devices to be introduced for end-users. It is becoming a popular way of interaction. Gaze coordinates could be used to pinpoint the target selected by the user on the screen, then proceed to do a command more efficiently. We hypothesize that the limitations and disadvantages mentioned for touch screen interfaces can be improved by adding gaze capabilities. It is worth mentioning that gaze is used anyway when people use the touch method and a finger or hand movement follows the user's gaze. Therefore, the targeting can be done via gaze and the selection by touch, which will improve the efficiency and also the screen is not obscured by hand during targeting. The multi modal approach will benefit the users with a user-friendly interface and aims to make the PIN entry secure and to improve the accuracy of target selection and reduce unwanted selections. The focus of this Master thesis is to analyse how the direct method of interaction could be optimized for PIN- entry, i.e., a multimodal approach of enhancing the popular Touch-based interaction using cues and context from gaze signals, so that it would increase accuracy, user-friendliness, efficiency, and security.

The security is needed for preventing and detecting unauthorized use of the computer. Prevention measures help to stop unauthorized users from accessing any part of your computer system. Detection helps in determining whether or not someone attempted to break into your system, if they were successful, and what they may have done.

Usable security is concerned with the study of how security information should be handled in the system, both at the user interface and in the back-end process, without

discarding consideration for resources and costs (Josang & Patton, 2003), according to this principle, a security mechanism should not make accessing are source, or taking some other action, more difficult than it would be if the security mechanism were not present. This means that a security mechanism should add as little as possible to the difficulty of the user's performing some action. Here the perception of "difficult" should account for the abilities, knowledge and mental models of the system users. In essence, for security to be more usable, it has to be less noticeable.

Personal Identification Number - Personal Identification Number (PIN), is a number or alpha-numeric password used in process of authentication a user accessing a system. The personal identification number has been the key for flourishing the exchange of private data between different data-processing centres in computer networks for financial institutions, governments, and enterprises.

PINs may be used to authenticate banking systems with cardholders, governments with citizens, enterprises with employees, and computers with users, among other uses.
common usage, PINs are used in ATM or POS transactions, secure access control, internet transactions or to log into restricted website.

Authentication is the act of confirming the truth of an attribute of a single piece of data claimed true by an entity. In contrast with identification, which refers to the act of stating or otherwise indicating a claim purportedly attesting to a person or things identity, authentication is the process of actually conforming that identity.

Personal identification numbers are widely used for user authentication and security. Password authentication using PINs requires users to physically input the PIN. Upon entering the PIN the validation process is performed. Once the validation process is completed the user can access his/her account based on the result of the validation process.

The use of personal identification numbers (PINs) is a common user authentication method for many applications, such as money management in automatic teller machines (ATMs), approving electronic transactions, unlocking personal devices, and opening doors.

1.2 Existing System and their Drawbacks

Password authentication is degrading as an authentication mechanism due to lack of memorability and security. In user authentication, the process which we have to pass through is user name and password. Generally, password systems are faced by problem of conflicting requirements. First is the fact that passwords should be usable and easy to remember.

The second requirement is for it to be secure. In today's changing world when we are having number of networks and personal account some sort of easy authentication schema need to be provided. The fact that an authorized user must enter the code in open or public places make PIN entry vulnerable to password attacks, such as shoulder surfing as well as thermal tracking.

Shoulder surfing is a type of social engineering technique used to obtain information such as personal identification numbers, passwords and other confidential data by looking over the victims shoulder. This attack can be performed either from the closed edge or from the longer edge. To implement this technique attackers do not require any technical skills, keen observation of victim's surroundings and typing pattern is sufficient. Crowded places are the more likely areas for an attacker to shoulder surf the victim.

The thermal tracking attack is also called as the ruminator and it can be used to retrieve sensitive user information such as passwords and PINs, as well as short strings of text. The thermal attack requires several conditions to be met in order to function properly. First, attackers need to place a specialized camera with thermal recording capabilities near the victim. The camera should also be able to capture the keys of the victim's keyboard. Through this attack the fingers thermal residue on the keyboard keys can be recorded by a hacker who can later use it to reveal the users passwords or any other text he/she has typed.

1.3 Proposed System

The work that has been presented in this report for real-time application for gaze based PIN entry, and eye detection and tracking for PIN identification using a smart camera. PIN authentication with hands-off gaze-based PIN entry techniques, on the other hand, leaves no physical foot prints behind and therefore offer a more secure password entry option. Gaze based authentication refers to finding the eye location across sequential image frames, and tracking eye centre overtime.

The purpose of this work is to enter and identify gaze-based PINs using a smart camera through real-time eye detection and tracking. NI Vision Builder and Lab View are used for eye tracking and for recording eye centre location on board the camera real-time. The smart camera allows on-board data processing and collection. Non-contact PIN based authentication adds a layer of security to physical PIN entries and are expected to reduce the vulnerability of the authentication process.

The aim of this study is to increase the security of pass face algorithm by creating resistance to shoulder surfing attack. Other hand, authentication using PINs requires users to

physically input the PIN, which could be vulnerable to password cracking via shoulder surfing or thermal tracking. PIN authentication with hands-off gaze-based PIN entry techniques, on the other hand, leaves no physical footprints behind and therefore offer a more secure password entry option. Gaze- based authentication refers to finding the eye location across sequential image frames, and tracking eye centre over time. This report presents a real-time application for gaze-based PIN entry, and eye detection and tracking for PIN identification using a smart camera.

1.4 Advantages of Proposed System

Implementing biometric authentication on an enormous scale is still a challenge, the use of personal identification numbers (PINs) is now a common user authentication method for applications such as, money management in automatic teller machines (ATMs), approving electronic transactions, unlocking personal devices, opening doors, etc.

For authentication, a user must be distinctive, quick and accurate for providing secure and widely accepted access. Fulfilment of all these issues is a challenge in itself. Most of the criteria mentioned above depend closely on how people cope up with the emerging technologies and previous works have demonstrated that user interaction with the system has a noteworthy effect on system performance. So, a strong defence system is evidently desirable.

The fact that an authorized user must enter the code in open or public places make PIN entry vulnerable to password attacks, such as shoulder surfing (observation user while typing the password through the keyboard), acoustics keyboard eavesdropping, thermal tracking and screen electromagnetic emanations.

To prevent these security issues, eye tracking is a natural interaction method and security systems based on eye movement tracking provide a promising solution to the system security and usability. The main advantage of this system is that it provides a double layer of security as it first detects and recognises the user's face and if it matches only then the user can enter the PIN using their eyes.

PIN authentication using the eye tracking feature eliminates the various issues of keyboard eavesdropping, shoulder surfing, etc. and secures the system thoroughly. It is relatively easier to use and faster compared to the traditional method of entering passwords or PINs.

Chapter 2

LITERATURE SURVEY

Meaning of Literature Survey

A Literature Survey or Narrative Survey is a type of survey article. A literature survey is a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic.

Different researchers worldwide have recently started experimenting and exploring the domain of authentication systems based on eye movements. Some literature surveys related to such ongoing research have been outlined in this section.

Survey Papers

Biometric authentication has fast evolved to be the default authentication mechanism on smartphones and other mobile devices. Apple's reported statistics show that 89% of its users have a Touch ID enabled smartphone use the Touch ID [1]. There are distinct advantages to using biometrics, including the fact that biometrics are much harder to manipulate and that the burden on the user is very light unlike in password-only authentication where the user is expected to remember long and obfuscated passwords.

With users facing an authentication challenge dozens of times in a single day, there is a distinct need for an approach that is both lightweight in terms of user burden and strong in terms of secureness.

There are two types of attacks that authentication challenges protect against:

The first is against a casual attack, where someone randomly picks up the mobile device and tries to use the device. Current biometric authentication approaches like Touch ID and Face ID are reasonably secure against such casual attacks. However, existing approaches have a bigger vulnerability to targeted attacks. In theory, an attacker can rely on a high-resolution photograph of the user's fingerprint to compromise Touch ID in a matter of minutes. While Face ID is a much newer biometric authentication mechanism, there already have been successful attempts to compromise certain aspects of it.

One of the drawbacks of such morphological biometric solutions is that the biometric template used for the authentication is static and hence any means to get access to that template is sufficient to compromise the authentication process.

Thus, there is considerable motivation to continue to develop newer and safer biometric authentication solutions. There are other biometric solutions such as those that rely on the user's voice where the authentication challenge can be a randomized prompt thus making it difficult to compromise. However, voice biometric solutions have some obvious limitations such as the voice of the user changing because of a cold, etc. Another class of biometrics is one that relies on physiological data of the user rather than simply morphological data. Extreme examples of physiological data include DNA or saliva composition.

While these are more robust in terms of secureness, they have a high cost of implementation both during initial setup, and for every authentication verification. In this paper, we consider a more accessible physiological data for a user – the user's electroencephalogram (EEG) data for a specific action blinking. With EEG growing to be a bonafide input modality in several commercial applications such as healthcare, gaming, and wellness, and the consequent wider availability of EEG headsets off-the-shelf, access to a user's EEG data is easier than it has ever been. At the same time, it is shown that blinks are actions for which the EEG signals are strongly identifiable. In this paper, we learnt that an efficient and accurate blink-based authentication method can be developed using features that capture granular differences in user blinks, as opposed to the central tendency measures or summary statistics.

We show that such a system can either be a cloud-based infrastructure that uses the data of multiple users or it could also operate in an unsupervised manner while only using the concerned user's data. Our work performs on a multi-class classification while combining 3 blinks with a TPR of 92% and an average per-user FPR of 0.7%. The performance for the unsupervised classification yields a TPR of 80% and an average per-user FPR of 2.2%. We plan to extend the future work in two main directions - (i) consider a more diverse set of features to improve the TPR while reducing the FPR of the system (ii) thorough testing of the system for a broader set of users, with multiple trials, and across different environmental conditions and mental states?

Gang Pan∗, Lin Sun, Zhaohui Wu [16] present a real-time liveness detection approach against photograph spoofing in face recognition, by recognizing spontaneous eyeblinks, which is a non-intrusive manner. The approach requires no extra hardware except for a generic web camera. Eyeblink sequences often have a complex underlying structure. We formulate blink detection as inference in an undirected conditional graphical framework, and are able to learn a compact and efficient observation and transition potentials from data.

For purpose of quick and accurate recognition of the blink behavior, eye closity, an easily-computed discriminative measure derived from the adaptive boosting algorithm, is developed, and then smoothly embedded into the conditional model. An extensive set of experiments are presented to show effectiveness of our approach and how it outperforms the cascaded Ada-boost and HMM in task of eyeblink detection.

[2] The accuracy of video-based eye-trackers using systems with high-quality cameras and multiple light sources is less than 2. It has been speculated that gaze estimation accuracy might be improved by using the vestibulo-ocular reflex. This involuntary reflex, results in slow compensatory movements of the eye in the opposing direction of head motion.

We therefore hypothesis that leaving the head to freely move during eye tracking must produce more accurate results than keeping the head fixed, only allowing the eyes to move. The purpose of this study was to create a low-cost eye tracking system that incorporates the vestibulo-ocular reflex in gaze estimation.

Video-oculography under passive illumination requires more processing. The performance of eye-tracking algorithms in video-oculography highly depends on the hardware used and the recording conditions. Video capturing systems attached to the head are notably faster than remote capturing systems. Video-oculography data processing comprises pupil detection and subsequent gaze estimation. Gaze estimation seeks the point of regard (POR, gaze point). Pupil detection algorithms segment images based on color and shape properties.

Color-based algorithms differentiating between the iris and its surroundings may produce false boundary edges. Shape-based algorithms using prior knowledge of the circularity of the iris and the pupil are invariant to translation, scale, and lighting. The optimization of a shape model requires multiple, time-consuming, iterations. Hybrid algorithms reduce the computation time by starting from a rough color-based algorithm. In addition, a shape model can be taken prior to color processing. Gaze estimation algorithms can be either interpolation-based or model-based. Interpolation-based algorithms perform a mapping from the eye to the point of regard either parametrically using polynomials or non-parametrically using neural networks or Support Vector Machines. Model based algorithms create a vector from the eye to the point of regard.

Researchers Wasiq Khan et al [3] proposed a novel pupil estimation method utilising the deep learning-based facial landmark detection and an image processing algorithm to determine the eye centre within an image frame. Reliable extraction of the eye frames within the input

image is one of the major advantages of using Dlib-ml. This eliminates most of the background and irrelevant segments of the image, which helps to identify the target segment using intelligent image processing.

They developed a customised iris kernel using multiple images from various datasets for its generalised representation. Then, the iris kernel is convolved with eye frame in two stages (horizontal and vertical) such that no nested strides are performed by the convolution function. The white paddings surrounding the kernel as well as the eye frame proved very helpful for template matching between the kernel and overlapped eye patches, specifically for the extreme eye positions (e.g., left/right corners).

In addition, utilising a dynamic threshold for identifying the best-matched patch further contributed to the reliability of our method. Compared to several state-of-the-art pupil detection methods, the proposed approach indicated significant improvements in pupil estimation accuracy, specifically with lower-resolution images and minimum error thresholds.

They also introduced a standardized distance metric to measure the relative error in model estimation. This metric can be used regardless of image size and resolution, which is not the case with most of the existing validation metrics used in similar works. In future works, the proposed method will be utilized along with eye-blink detection models to determine eye gaze, in particular for infraduction iris positions. Our method can be useful in various computer vision applications, specifically the one requiring precise pupil and eye centre estimation.

For instance, the eye-related feature extraction in can be replaced with our method to extract the more reliable and micro-level movements within the eyes to distinguish truthful from deceptive behavior. More explicitly, this work is expected to direct several application areas such as human–computer interfaces, gaze estimation, emotion recognition, psychological profiling, fatigue detection, healthcare, visual aid, and automated deception detection.

Researchers Ahmad Aljaafreh, Murad Alaqtash et al[4] in this work developed a low-cost system for saccadic measurements. The system records the raw data of gaze location and dynamic stimuli using a low-resolution webcam. They proposed an algorithmic approach to process and extract saccadic parameters, i.e., time latency, amplitude gain and peak velocity. Experimental results demonstrated that the proposed strategy is a quick, simple and efficient technique for eye tracking and saccade measurement. Future research directions might be utilizing machine learning techniques to recognize and classify normal and pathological gaze patterns. This can be used to assist clinicians and medical physicians in the diagnosis and identification of neurological disorders.

In general, they developed a tool that can be used by clinicians and medical physicians for the diagnosis and identification of neurological disorders. Hence, we use this study of their research on eye tracker and saccade system for a reference of eye tracking using pygaze or eye gazing technology.

A Siripitakchi et al. [5] proposed a new concept CAPTCHA (which is the common method to differentiate between human and machine) based eye movement characteristics. Biometrics is a method that uses a unique pattern obtained from physiological or behavioral features to identify and verify individuals. There are many features such as fingerprint, face recognition, palm print, hand geometry, iris, and voice recognition. Furthermore, an eye movement characteristic is one of the new technologies applied for biometrics. Several researches studied about gaze detection and eye movement for identifying and verifying humans. In this study, the reliability of biometric system with eye movement characteristic is integrated to CAPTCHA for tackling its weakness.

Thus, he proposes a new biometric-based CAPTCHA system called EYE-CAPTCHA using gaze detection and eye movement. Verification by CAPTCHA is commonly used to separate between human and machine. It can protect the website or the system from some attacks by automated programs. Since the existing CAPTCHAs have some limitations, here the new CAPTCHA, which is called EYE CAPTCHA is developed under the technologies of gaze detection and eye movement. The objective of design this CAPTCHA is associated in terms of safety and can be used interchangeably with old CAPTCHA. He says Eye movement is one characteristic that can be used for biometric authentication. There are several research related to the use of eye movement for identifying people.

Researchers Z. Li et al.[7] described iType concept that use eye gaze technology enhancing its typing piracy. They devised effective techniques to address a series of design challenges, covering accuracy, latency and mobility several aspects. They have consolidated their devised techniques and implemented iType on the iOS platform.

In iType, the keyboard consists of multiple buttons and each button represents unique character(s) (number or letter). To type a password, the user looks at the corresponding buttons sequentially, and iType essentially solves a decoding puzzle: it reads the user's gaze, infers the buttons being looked at, and assembles the password. The iType typing is secure primarily due to the fact that the eye gaze is difficult to eavesdrop. Even an adversary in front of the user could decode the eye gaze, the gaze itself conveys no meaningful information, unless it matches with the keyboard layout, which however can be user-defined and changed. Experiments show iType

achieves high typing accuracy within reasonable short latency in variant environments. A user study further verifies the efficacy of the iType design. In the future, it can be planned to conduct a larger-scale field study to obtain more efficient algorithms for the same study.

Z Li faced few challenges such as:

1) Low accuracy of mobile gaze tracking. iType is designed atop the gaze tracking technique, which relies on the gaze tracker trained in advance. As the relative position between front camera and user's eyes may vary over time during the real usage, it could make existing gaze tracking solutions inherently inaccurate and unreliable when they are used on mobile devices. Precisely inferring typed characters based on low-accuracy gaze tracking results is difficult.

2) Lack of true text-entry value in error correction. The assembled password may contain typing errors, which will be rejected by the underlying application, and the user has to type it again to correct the errors. Due to the privacy concern,the password plain text is not displayed on the screen, and the correctness of each recognized character is thus unknown to the user. If iType verifies to the application only when an intact password is obtained, it may significantly impair the typing efficiency and prolong the typing delay.

3) Noises from device motions. As front camera continuously tracks the user's face, device motions during typing may blur the captured frames and degrade the quality of gaze tracking. The gaze tracking accuracy further deteriorates, which will in turn degrade the iType performance.

He overcame the above challenges by contributing following:

1) Although the individual gaze tracking results (points) are unreliable, their statistics can give good approximations. They then looked at a group of gaze tracking points and leverage their collective behavior to approximate the user's true gaze position. Based on that they proposed a technique to confirm the typing of each character using a minimal number of gaze tracking points. It ensures both the accuracy and delay for typing individual characters.

2) When assembled, password contains typing errors, iType requires another round of user's input. He observed that rearranging the layout of buttons shuffles the vicinity of true gaze positions, which thus provides opportunities to migrate typing errors and jointly decode user's input cross multiple rounds even the typing channel has a low signal-to-noise ratio.

3) The quality of the frames generated from the front camera is impacted by device motions. iType leverages accelerometer readings to select frames with the best expected quality to enhance the typing performance.

The main study we learnt from Z Li's research is:

1) Gaze tracking accuracy on mobile devices- The input of gaze tracking is a stream of frames from front camera that captures user's eyes. The core module, gaze tracker, then calculates (x, y) coordinates of the user's gaze on the screen plane for each frame.

2) Gaze engine- Gaze engine converts the live video frames to a stream of gaze point coordinates. After iType starts, video frames are extracted from front camera and they are attached with accelerometer readings. Due to the processing delay, only a subset of frames can be utilized to produce gaze tracking points.

His study produced some useful experiments such as:

- Impact of input password sequences-no obvious influence on the accuracy and the per character latency increases from 2.2s to 3.0s on average.

- Impact of device-to-eye distances. The device-to-eye distance is usually around 25cm to 35cm when people use mobile devices. the performance exhibits no remarkable differences within 35cm, but it drops significantly beyond this distance.
- Thus, the device-to-eye distance within a common range like 25cm to 35cm, has a minimal impact on the typing performance.
- Impact of head movements- We also find when head moves back, the performance can be recovered at a certain degree, but this experiment still inspires us to further enhance iType by better handling head movements in the future.

Chunning Meng and Xuepeng Zhao [6] propose using convolutional neural network (CNN) for analysis of webcam-based eye movement . The CNN is applied to estimate eye movement by training eye features collected from five points: inner corner, outer corner, center of upper eyelid, center of lower eyelid, and center of iris. Alexandra Papoutsaki et al. propose WebGazer library for webcam eye tracking using user interactions.

They release the eye tracking library named WebGazer, in which a webcam is used to find gaze locations on a screen in real-time. WebGazer consists of two parts, a detector using eye detection gauge and a gaze estimator using analysis information of interactions between user and screen. He used the library integrated into the website for gaze interactions in real-time. As he mentioned in the previous section, CAPTCHA and Eye Movement plays a major role in providing appropriate human verification. In this study, he proposed CAPTCHA based on gaze detection and eye movement that can replace the existing CAPTCHA. This proposed CAPTCHA belongs to the type of puzzle-based CAPTCHAs.

Krafka et al. [7] developed an end-to-end eye tracking solution targeting mobile devices, and a large-scale dataset with almost 2.5 million frames was introduced to train the CNN models, which can achieve a significant reduction in errors. Due to the use of deep and multimodal networks and large-scale training dataset, these methods can be applicable in the unconstrained daily-life setting with arbitrary head poses.

However, the estimation precision of these methods is not good enough for eye-based human-computer interaction. If a distance of 70cm is kept from the person to the computer screen, the error is around 7.7cm for Zhang's method. The error of GazeCapture around 1.5cm is also larger than the distance between two adjacent app icons on mobile phones.

Precise gaze estimation is a powerful guarantee for the technology of eye movement analysis. Although deep convolutional nets have achieved breakthroughs for gaze estimation in low-cost and daily-life scenarios, the precision is still unsatisfactory. It should be noted that gaze point or orientation is not necessary for eye movement analysis in various domains such as nystagmus or dyslexia diagnoses, eye based lie detection, fatigue detection and eye-based activity recognition. A mapping function should exist between a length of eye movement videos and one type of activity or state. Time-varying eye movement information can be used for analyzing eye movement without considering gaze and related calibration. The advantages of abandoning gaze estimation based model can be summarized as follows:

(1) The error of gazing mapping calibration can be avoided.

(2) More features related to eye movement (not just the center of iris which is unreliable due to eyelid occlusion) can be employed to compensate the error of iris center detection. For example, the open width of eye is a good supplement for low quality videos.

(3) Relative movement information is easier to be detected in inter-frame of low quality videos, which can be emphasized rather than the rough absolute position.

In almost all previous research, the basic eye-movement types such as saccades are considered the basic elements of eye movement. However, Heiko, Salvucci et al. and Goldberg et al. [19], argued that these types are meaningless in physiology and they are not necessary for eye-movement analysis. Therefore, we suggest abandoning the use of basic eye-movement types. The feasibility of this idea has been assessed.

Webcam based eye movement record method becomes more natural and noninvasive compared with the method in, where results of electrooculography become insignificant due to vestibule-ocular reflex.

They proposed a novel method to analyze eye movement under webcam. Different from the feature-based gaze tracking method, the feature points are used to obtain eye movement signals instead of estimating the mapping function. Compared with, the specific improvements can be summarized as follows:

(1) The CNN is used to extract the eye feature points, where the detection results can become more robust and precise.

(2) Eye movement features are extracted by the CNN rather than artificial extraction.

(3) More cues are added to analyze the eye movement patterns, which include relative displacement of iris center and variation of open width. The experimental results show that the proposed method obtains promising results. The advantages of lower invasive, lower cost and easier operation in webcam-based method are the key component for the popularization of the applications based on eye movement.

Researchers continued to proceed their study which we are using for a reference here in 2 steps:

1- Eye Feature Point Detection

Eye movement information can be described by eye feature point in sequential eye images. In this paper, five feature points are specifically defined and calibrated based on a unified standard. In a typical figure of an eye there are five points Consider in particular, points A and B are inner and outer corners of the eye, respectively, and points C and D represent the centers of upper and lower eyelids, respectively. These four points are calibrated by the PB-points method introduced in our early works. The location of the center of the iris (point E) is difficulty to be identified, since the video images captured by webcam are not good enough and eyelid occlusion is ubiquitous.

In this paper, point E is defined as barycentre of the area surrounded by upper, lower eyelids and the visible iris outer edge. To detect the eye feature point, points-CNN is designedInstead of the Sigmoid and Tanh functions, the ReLU (Rectified Linear Units) activation function is used to produce the output feature of the convolutional layer with reduced computational complexity. Two max-pooling layers are used, similar to the LeNet network architecture.

2- Eye movement Analysis

In traditional eye movement analysis methods, two steps are necessary. Firstly, basic eye-movement types such as saccades, fixations, and blinks should be detected. Secondly, analysis modeling is established based on the detected types.

The relationship between basic eye-movement types and human activity is very close. However, it is almost impossible to explicitly define a sort of basic eye-movement type.

In addition, these types do not make any sense in physiology, which are not necessary. Furthermore, it is impossible to detect fast saccades (duration less than 30ms), where larger errors would be produced by basic eye- movement types detection algorithms for webcam. Therefore, it is preferable to directly use the original time-varying eye movement signals (for example the displacement of iris center) without detecting the basic eye-movement types. The feasibility of this strategy has been validated in our previous work.

In this paper, a novel eye movement analysis model is proposed. To verify the feasibility, a webcam-based visual activity dataset is collected and constructed. The proposed CNN-based model outperforms other state-of-the-art methods in three office activities recognition tasks. Although the accuracy of eye tracking is limited by the quality of webcam, it is revealed that the proposed webcam-based eye movement analysis method can be successfully employed to recognize human visual activities without gaze estimation and detection of the basic eye-movement types.

Moreover, it can be shown that accurate recognition results can be achieved by using CNN, which can extract more representative internal feature from original time-varying eye movement signals. In particular, this model provides a new way for eye movement analysis by detecting the feature point and classifying the original time-varying eye movement signals using CNN.

Researchers M. Khamis et al. [9] introduced multimodal authentication combining gaze and touch on mobile devices. GazeTouchPass is significantly more secure than single modal systems, particularly against side attacks due to having to quickly switch focus between phone and eyes. Its usability compares favorably to state-of-the-art schemes. His theory concludes that the use of multiple modalities can greatly enhance the security of authentication systems against advanced as well as basic threat models, while maintaining high usability.

In general, the appearance-based methods do not require calibration of cameras and geometry data. The image content is used as input for estimating the underlying function for gaze points or gaze orientation. be more flexible, it is very sensitive to head movements, where the accuracy is limited. More recently, deep learning has become a promising tool for computer vision applications, by achieving remarkable performance gains. By using CNN, superior performance of gaze estimation can be obtained to learn representations from huge amounts of data.

Zhang et al.[10] proposed a method for in-the-wild appearance-based gaze estimation using multimodal CNN, where a dataset that contains 213,659 images is collected.

Researcher R Revathy [11] says, The personal identification number (PIN) is one of the day-to-day user authentication technique used in diverse situations, such as in with drawing cash from an automatic teller machine (ATM), approving an electronic transaction, unlocking a mobile device, and even opening a door.

In computer security, shoulder surfing refers to using direct inspection techniques, such as peeping over someone's shoulder, to acquire information. Since the identical PIN is usually selected by a user for diverse purposes and applied frequently, a compromise of the PIN might cause the user a serious risk. Shoulder surfing is frequently used to acquire passwords, PIN security codes, and related data.

Shoulder surfing can also be done at a long distance using binoculars or other vision-magnify devices. Inexpensive, miniature closed circuit television cameras can be hidden in ceilings, walls or fixtures to perceive data entry. Most of the known shoulder-surfing resistant PIN-entry methods apply the fact that the capacity of Short-term memory and the actual-time processing performance of a human are very restricted. The users are susceptible to malevolent people nearby or spyware inside because they can grasp the key input, especially confidential input such as a password, in mobile environs.

To stop shoulder surfing, it is recommended to shield paperwork or the keypad from view by using one's body or cupping one's hand. To deal with this problem, which is between the customer and the system, cryptographic prevention approach is hardly relevant because users are restricted in their capacity to process information. Among them, the PIN entry technique introduced was effective because of its clarity and instinctive: in every round, a structured numeric keypad is colored at odd; half of the keys are in black and another half in white, which is called the BW method.

A customer who knows the accurate PIN digit can enter the color by pressing the distinct color key below. The primary BW method is targeted to withstand a human shoulder surfing attack. Our proposal called covert attention shoulder surfing indeed can crack the well known PIN entry technique formerly estimated to be secure against shoulder surfing.

From the study conducted by R Revathy [11], it is concluded how the normal PIN entries as well as passwords can be perceived easily. The PIN entry can be perceived by close by ad-

versaries, more effectually in a crowded place. the PIN entry was elegant because of its simplicity and accessibility. Hence she describes how this can be avoided by introducing a new approach of utilizing the PIN. Although her study was about introducing concepts of BW in ensuring the security of PIN. It surely encourages to build some new studies of utilizing PIN entry. Hence, we make use of studies from few papers for our further utilization. They considered the challenges of securing users' passwords on the internet and shown some equivalent effort in this field. Also we have discussed how to secure users' passwords which is being thieved by adversaries in the above paper.

Human adversaries are more strong than awaited when shoulder surfing. The advanced system reduced the difficulties of shoulder surfing or eves dropping by introducing the different PIN entry methods. The covert attention shoulder surfing suggested in this paper is to expertise the first experienced counter-attack of humans against the system, formerly estimated to be secure. Here what we have well read from the delicacy of the BW method is that obtaining both securities and usability is honestly demanding and subject to incorrect plan due to the absence of formal remedy. Also, they introduced a new PIN-entry technique that has proposed security against human shoulder-surfing attacks. This is feasible by successfully enlarging the part of memory needed by a shoulder surfer.

Pawel Kasprowski and Katarzyna Harezlak [12] present results in the competition of eye movement verification and identification. Among five participants, the best result is shown in terms of recognition rate, memory effect, and data dependency.

Dat Tien Nguyen et al. proposed a method for head pose estimation in smart TV by using a new gaze detection method based on Face ROI extraction. With their method, the boundary of face and the shoulder line are detected and then combined to estimate the head pose. For the experimental results, from the original CAPTCHA can be replaced by the EYE CAPTCHA. However, he explains there are some issues such as how to improve the system friendlier to users to be concerned.

Researcher D. Rozado [13] has explored videoculography gaze tracking method while inputting password in a skillful manner such that shoulder surfing can be avoided. He proves before that eye movements can be consciously controlled by humans to the extent of performing sequences of predefined movement patterns, or "gaze gestures" that can be used for human-computer interaction purposes in desktop computers. By learning how the human gestures can interact we utilized this study to learn about the eye positioning and considering the eye blinking movement gesture to consider to lock a PIN.

He proposes that gaze gestures can be an effective input paradigm to interact with handheld electronic devices. He proves it through a pilot user study how gaze gestures can be used to interact with a smartphone, how they are assimilated by potential users, and how the Needleman-Wunsch algorithm can effectively discriminate intentional gaze gestures from otherwise typical gaze activity performed during standard interaction with a small smartphone screen.

Researchers M. Martin et al. [14] described an eye tracking study of proposed Image Pass system, which can be considered as a graphical authentication system that is based on recognition presented the first eye-tracking study of a graphical authentication mechanism. His analysis offered insight into the initial perception and behaviour of users during interaction with an object-based graphical authentication mechanism.

With some design changes the ImagePass concept was shown as potentially feasible as enrolment to the system was completed relatively fast, in roughly a minute. There are also potential differences in how male and female users observe the system, which, however, should be further researched before any specified claims are made.

Researchers M. Brooks et al. [15] have discussed how there can be security breach with normal PINs and passwords which is the reason why biometrics can reduce the vulnerabilities but doesn't clearly mention how biometrics can completely prevent this.

A wide variety of physical and behavioral characteristics have been successfully used for biometric identification. Of these, fingerprint, iris, and face recognition have received the most attention recently, and there have been large-scale deployments, such as UIDAI's biometric identification for India's 1.2 billion citizens However, there are significant barriers to widespread adoption of biometric authentication technology, including usability and accessibility. The ways in which people interact with a biometric system can shape the overall security and performance of the system.

To his knowledge, the potential of eye movement as a biometric identification technique was first demonstrated by Kasprowski and Ober [17]. Research has focused on decreasing the recognition error rate by developing salient features of gaze data and evaluating classification algorithms for distinguishing individuals. Although recognition error rates for this technology remain high, recent progress has been promising. The experiment was conducted in presence of 22 participants and measurement metrics included authentication time, security, acceptability and usability of proposed system in comparison to its traditional PIN based counterpart.

Most participants selected the proposed system as the preferable approach to the traditional PIN based system. system. There is growing interest in eye movement biometric identification. The IEEE Fifth International Conference on Biometrics: Theory, Applications, and Systems (BTAS 2012) featured a competition on eye movement verification and identification. With programs such as UIDAI increasingly applying biometric technology at a massive scale, the importance of understanding human interaction with these systems is clear. After the fixation, eyes move rapidly to another gaze point – another fixation. This rapid movement is termed a saccade. Saccades differ in longitude, yet always are very fast.

According to a survey conducted by Hansen [18], the taxonomy of the existing gaze estimation consists of feature-based and appearance-based methods. The feature-based method is the most popular gaze estimation method, which requires to extract gaze related local features. Feature-based methods can be divided into corneal reflection and shape-based methods, where the categorization is based on the adoption of external light sources.

Corneal-reflection methods that use multiple cameras and multiple infrared lights have been successfully applied such as Tobi and SMI eye tracker. The precision of this method is satisfactory since the pupil center and the glint can be easily extracted to calibrate the errors caused by head movements. However, complex calibration, high cost and ill-health caused by IR lights are unavoidable.

Moreover, IR light based systems are not reliable when used in outdoor conditions. Shape-based methods cannot obtain estimation with high accuracy and require high image quality with precisely extracted pupil, iris and eyelid edges. However, both the pupil and glint are often unavailable in videos captured by webcam.

In general, the appearance-based methods do not require calibration of cameras and geometry data. The image content is used as input for estimating the underlying function for gaze points or gaze orientation. be more flexible, it is very sensitive to head movements, where the accuracy is limited. More recently, deep learning has become a promising tool for computer vision applications, by achieving remarkable performance gains.

By using CNN, superior performance of gaze estimation can be obtained to learn representations from huge amounts of data. Zhang et al. [11] proposed a method for in-the-wild appearance-based gaze estimation using multimodal CNN, where a dataset that contains 213,659 images is collected.

Researchers V. Paul and M. Jones[19] brings together new algorithms and insights to construct a framework for robust and extremely rapid object detection. They demonstrated on and in part motivated by, the task of face detection presented an approach for object detection which minimizes computation time while achieving high detection accuracy.

Toward this end they constructed a frontal face detection system which achieves detection and false positive rates which are equivalent to the best published results .This face detection system is most clearly distinguished from previous approaches in its ability to detect faces extremely rapidly.

Operating on 384 by 288 pixel images, faces are detected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates.

There are three main contributions of their object detection framework. We will introduce each of these ideas briefly below:

The first contribution of this paper is a new image representation called an integral image that allows for very fast feature evaluation.

The second contribution of this paper is a method for constructing a classifier by selecting a small number of important features using AdaBoost. Within any image sub-window the total number of Harr-like features is very large, far larger than the number of pixels. In order to ensure fast classification, the learning process must exclude a large majority of the available features, and focus on a small set of critical features.

The third major contribution of this paper is a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image. The notion behind focus of attention approaches is that it is often possible to rapidly determine where in an image an object might occur. More complex processing is reserved only for these promising regions. The key measure of such an approach is the “false negative” rate of the attentional process. It must be the case that all, or almost all, object instances are selected by the attentional filter.

This approach constructed a face detection system which is approximately 15 faster than any previous approach. brings together new algorithms, representations, and insights which are quite generic and may well have broader application in computer vision and image processing.

He presents a set of detailed experiments on a difficult face detection dataset which has been widely studied. This dataset includes faces under a very wide range of conditions including: illumination, scale, pose, and camera variation. Experiments on such a large and complex dataset are difficult and time consuming. Nevertheless, systems which work under these conditions are unlikely to be brittle or limited to a single set of conditions. More importantly conclusions drawn from their dataset are unlikely to be experimental artifacts.

Heiko, Salvucci et al. and Goldberg et al. [20], argued that these types are meaningless in physiology and they are not necessary for eye-movement analysis. Therefore, we suggest abandoning the use of basic eye-movement types. The feasibility of this idea has been assessed. Webcam based eye movement record method becomes more natural and noninvasive compared with the method in, where results of electrooculography become insignificant due to vestibule-ocular reflex. They proposed a novel method to analyze eye movement under webcam. Different from the feature-based gaze tracking method, the feature points are used to obtain eye movement signals instead of estimating the mapping function.

The work of Papageorgiou et al [21]. His detection system does not work directly with image intensities. Like other authors he used a set of features which are reminiscent of Haar Basis functions (though we will also use related filters which are more complex than Haar filters). In order to compute these features very rapidly at many scales he introduced the integral image representation for images. The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Harr-like features can be computed at any scale or location in constant time.

Chapter 3

ANALYSIS

3.1 Problem Identification

Today, the Internet has entered our day-to-day life and all the services have been moved online. Beyond reading the news, looking for information, and other threat free tasks, we have also become accustomed to other risk-related work, such as paying using credit cards, checking/composing emails, online banking, and so on. While we appreciate its benefits, we are placing ourselves at risk.

Human factors are often considered the weakest link in a computer security system. There are three major areas where Human-Computer Interaction is important and those are authentication, security operations, and developing secure systems. Here we focus on the authentication problem.

These techniques are not safe because they are viewed by malicious observers who use surveillance techniques such as shoulder-surfing (observation user while typing the password through the keyboard) to capture user authentication data. Also, there are security problems due to poor interactions between systems and users.

Passwords that are hard to guess or break are often hard to remember. Studies have shown that since users can only remember a limited number of passwords, they tend to write them down or will use the same passwords for different accounts. To address the problems with traditional username and password authentication, alternative authentication methods, such as biometrics have been used.

But biometric systems also include keypad based systems which are easier to hack and voice controlled passwords which are user specific. And even head controlled or gesture controlled systems where physical movement is required. The authentication scheme that uses text as well as graphics has lacked the security factor. and the existing authentication schemes are not immune to shoulder surfing and brute force attack.

The use of Personal Identification Numbers (PINs) is a common user authentication method for many applications, such as money management in automatic teller machines (ATMs), approving electronic transactions, unlocking personal devices, and opening doors. Authentication is always a challenge even when using PIN authentication, such as in financial systems and gateway management. According to European ATM Security, fraud attacks on ATMs increased by 26% in 2016 compared to that of 2015.

The fact that an authorized user must enter the code in open or public places makes PIN entry vulnerable to password attacks, such as shoulder surfing as well as thermal tracking. Therefore a combination of both the PIN System and the Biometric System is both safer and also difficult to hack.

As a result, the researchers proposed a three-layered security framework to secure PIN numbers, where users can enter the password by blinking the eye at the suitable symbols in the appropriate order and thus the user is invulnerable to shoulder surfing. Eye blinking is a natural interaction method and security systems based on eye blink tracking provide a promising solution to the system security and usability. The aim of this paper is to review techniques or solutions to dealing with eye blink in security systems.

3.2 Objectives

To develop a model for Eye Movement Based PIN entry system using a smart camera through real time eye detection and tracking. The objective of this work is to increase the security of the password authentication system by entering the password through eye movement.

Eye tracking is the process of detecting the eye location across a video frame. The motion of the eye relative to the head may also be more interesting. Eye tracking is important for development and research areas such as visual systems, psychological analysis, cognitive science and product design.

An eye tracking system is an integration of a set of devices and associated programs for measuring eye positions and movement, and correlating the results to the same eye across images acquired sequentially over time. One of the security requirements for general terminal authentication systems is to be easy, fast and secure as people face authentication mechanisms every day and must authenticate themselves using conventional knowledge based approaches like passwords.

The methods for entering passwords can be made safe enough using latest methods such as eye tracking. It means to make use of your eyes which will not leave prints like when we enter password by hands, which can be retrieved through silica gel, so there’s no point of safe entry of password. So, the eye tracking system can be used for safer options which have many methods in it, here we choose a method like blinking of eye for password authentication, which will not leave any prints behind.

The way in which the system works is to design the system and to train the system for detecting face region and eye region within the image that is captured by the camera that is

mounted on the laptop. The eye movement tracking should occur continuously i.e., the continuous image capturing from the camera has to take place.

Detecting of the face and eye region in the image will be done continuously to track the eye movements, through these detections the gaze ratio of the eye can be calculated, which helps in selecting the keyboard and the blinking ratio of the eye will be calculated, which will be useful in selecting and updating the letter from the keyboard that has been selected and displayed by the system through calculating the gaze ratio of eye.

Through this process the password has to be updated to the system, then the validation process has to be done between the password that has been stored in the system and the password entered or updated by the user. Then the locker has to be made open for authenticated users i.e., the synchronization between the system and locker has to be achieved.

Some of the important things that we will achieve through this is Confidentiality as the System will protect the data i.e., the password and any other sensitive data so that only authorized person can access the data, Access Security as Password that is entered by the user will not be visible at any point even while password is being entered so that the system ensures that it is safeguarded against deliberates and intensive fault occurring internally or externally. and Integrity as System will ensure the data here it is password is accurate and Reliability as the System will perform the password authentication process without any failure when the user makes a request for the transaction and maintainability as, if any faults or failure occurs in the system it will be fixed easily.

Flexibility as Software can be modified to different environments and configurations and scalability as the number of users requesting the transactions would increase in peak holidays and at this time system will expand its processing capabilities. And finally, portability as software can be easily transferred from its current hardware or software environment to another.

3.3 Methodology

Image Processing is a technique to enhance raw images received from cameras/sensors placed on space probes, aircrafts and satellites or pictures taken in normal day-today life for various applications. An Image is a rectangular graphical object. Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data.

The operations that come under image processing are image enhancement operations such as sharpening, blurring, brightening, edge enhancement etc. Image processing is any form of

signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image.

Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible.

Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations such as sharpening, blurring, brightening, edge enhancement. Traffic density of lanes is calculated using image processing which is done of images of lanes that are captured using a digital camera. We have chosen image processing for calculation of traffic density as cameras are very much cheaper than other devices such as sensors.

Image Acquisition:

Generally an image is a two-dimensional function f(x,y) (here x and y are plane coordinates). The amplitude of the image at any point say f is called intensity of the image. It is also called the gray level of image at that point. We need to convert these x and y values to finite discrete values to form a digital image. The image of the retina is taken for processing and to do the processing of the lanes to find out the number of cars present in a lane. We need to convert the analog image to digital image to process it through a digital computer. Each digital image is composed of finite elements and each finite element is called a pixel.

Image scaling occurs in all digital photos at some stage whether this be in Bayer demo slicing or in photo enlargement. It happens anytime you resize your image from one pixel grid to another. Image resizing is necessary when you need to increase or decrease the total number of pixels. Even if the same image resize is performed, the result can vary significantly depending on the algorithm.

Images are resized because of a number of reasons but one of them is very important in our project. Every camera has its resolution, so when a system is designed for some camera specifications it will not run correctly for any other camera depending on specification similarities. So, it is necessary to make the resolution constant for the application and hence perform image resizing.

Humans perceive colors through wavelength-sensitive sensory cells called cones. There are three different varieties of cones, each has a different sensitivity to electromagnetic radiation (light) of different wavelengths. One cone is mainly sensitive to green light, one to red light,

and one to blue light. By emitting a restricted combination of these three colors (red, green and blue), and hence stimulating the three types of cones at will, we can generate almost any detectable color. This is the reason behind why color images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such color images as stored in an RGB format.

In grayscale images, however, we do not differentiate how much we emit of different colors, we emit the same amount in every channel. We will be able to differentiate the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels. When converting an RGB image to grayscale, we have to consider the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One of the approaches is to take the average of the contribution from each channel: (R+B+G)/3. However, since the perceived brightness is often dominated by the green component, a different, more "human-oriented", method is to consider a weighted average, e.g.: 0.3R + 0.59G + 0.11B

About Python

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. Python is distributed under an OSI approved open source license that makes it free to use, even for commercial products.

Introduction to OpenCV

Open CV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD licensed product, OpenCV makes it easy for businesses to utilize and modify the code. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. Open CV is written natively in C++ and has a templated interface that works seamlessly with STL containers. Here we are using OpenCV for processing the image of the eye.

3.4 System Requirements Specification

In this section we discuss in a brief way about all the requirements we need for our project, we use software and hardware components for our project.

3.4.1 Software Requirements Specification

Operating System : Windows 7/8/10

Software : OpenCV

RAM : Minimum 4GB

Graphic card : 2GB

Drivers : 2.0 USB WEBCAM drivers

3.4.2 Hardware Requirements

Processor : Dual-Core intel i5 processor Laptop

Speed : Min 1.3 Ghz

RAM : 4 GB

Hard Disk : 80 GB

Monitor : LED

3.4.3 Functional Requirements

Functional requirement defines a function of the system or its component, where a function is described as a specification of behaviour between outputs and inputs.

1. Face and Eye detection: System will detect the face and eye of a person from the input i.e., the image captured by the camera.

2. Feature extraction: System will extract the eye region then the eye pupil region and this region is marked and the eye pupil center will be located.

3. Eye tracking: System will track the eye pupil movement and mark the eye pupil center on the coordinate system.

4. PIN identification: System will identify the PIN based on the point that is located on the coordinate system and the updated password will be sent for further authentication process.

5. Authentication: Once the system receives the password the authentication process will be done to verify whether the password entered is valid or not. So, that the unauthorized user cannot access the account.

6. Authorization: If the entered password is valid then the system allows the locker to open so that the user can further do his transactions else the locker remains locked.

Chapter 4

SYSTEM DESIGN

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of system theory to product development

4.1 System Architecture

The System Architecture diagram depicts the overall structure of the software application or model that is to be created or already created architectural diagram. It uses information flow characteristics and maps them into the program structure. The system architecture is shown in the Figure 4.1 below.

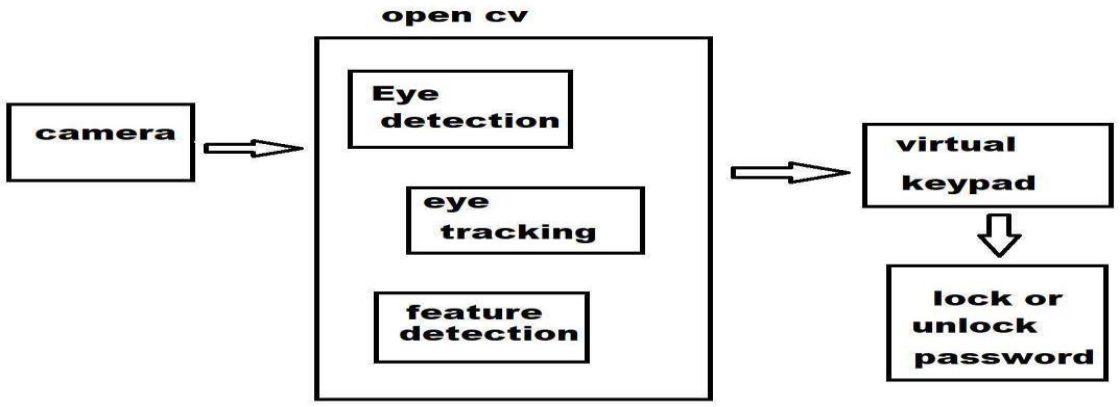

Figure 4.1 System Architecture

1. Camera

Web camera is used to capture the continuous images i.e., the video of the person in front of the camera. The captured image acts as input to Open CV. This captured image is sent Eye detection module.

2. Open CV

The image from the camera is fed into Open CV. This image is sent to eye detection module in Open CV where the face and eye region in the image would be captured and the respective window location is sent to feature detection module, here the co-ordinates of the eye region is will be the output.

Lastly in the eye tracking module the eye movements will be tracked to get the gaze ration and the eye blinks will be detected to get the blinking ratio. Based on these two ratios the password would be updated.

4.2 Detailed Design

Here the focus is on deciding which modules are needed for the system, the specifications of these modules low and how the module should be interconnected. It includes the high-level design and low-level design.

4.2.1 High-Level Design

High-Level Design (HLD) explains the top-level architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

The HLD uses possibly nontechnical to mildly technical terms that should be understandable to the administrators of the system. In contrast, low-level design further exposes the logical detailed design of each of these elements for programmers. The High-Level Design is shown in the Figure 4.2 below

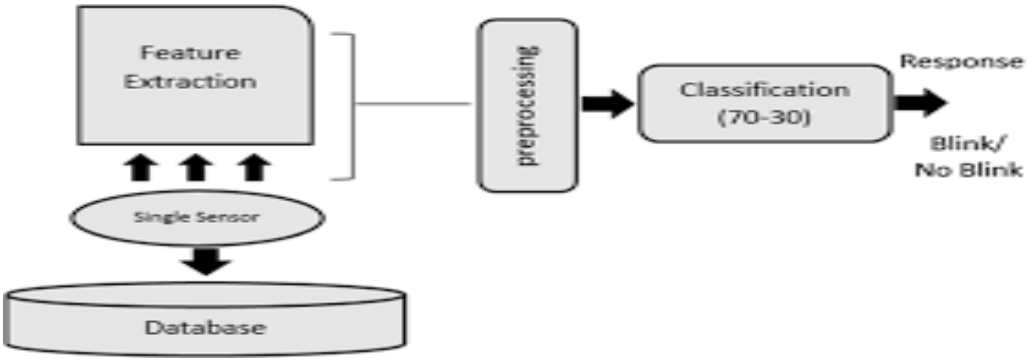

Figure 4.2 High-level design

4.2.2 Low-Level Design

Low-Level Design (LLD) is a component-level design process that follows a step-by-step refinement process. It is also termed as detailed design. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. Post-build, each component is specified in detail

The LLD phase is the stage where the actual software components are designed. During the detailed phase the logical and functional design is done and the design of application structure is developed during the high-level design phase. The goal of LLD or a Low-Level Design Document (LLDD) is to give the internal logical design of the actual program code. Low-level design is created based on the high- level design. It describes the modules so that the programmer can directly code the program from the document.

A good low-level design document makes the program easy to develop when proper analysis is utilized to create a low-level design document. The code can then be developed directly from the low-level design document with minimal debugging and testing. Other advantages include lower cost and easier maintenance. The low-level design is shown in the Figure 4.3 below.

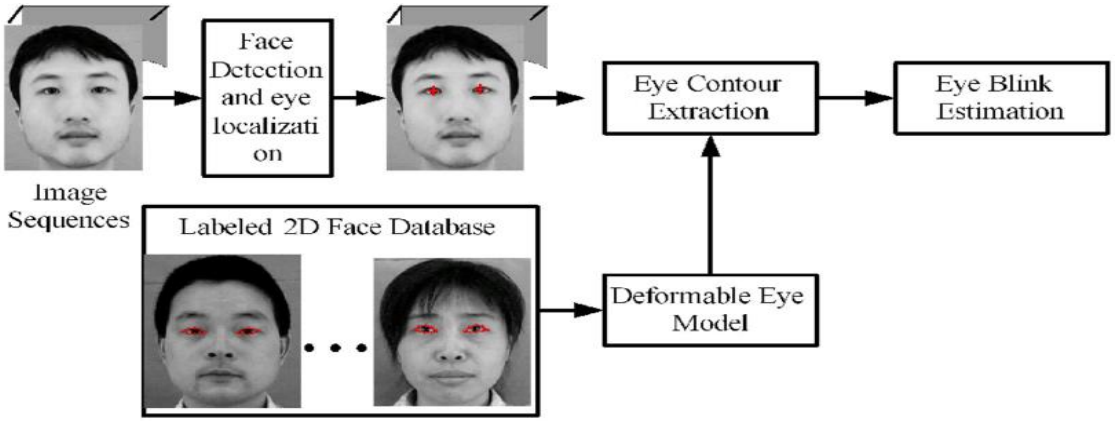

Figure 4.3 Low-level design

4.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The DFD may be used to perform a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily two levels in the data flow diagram, which are: 0-level DFD and 1-level DFD.

4.3.1 Level – 0 Data Flow Diagram

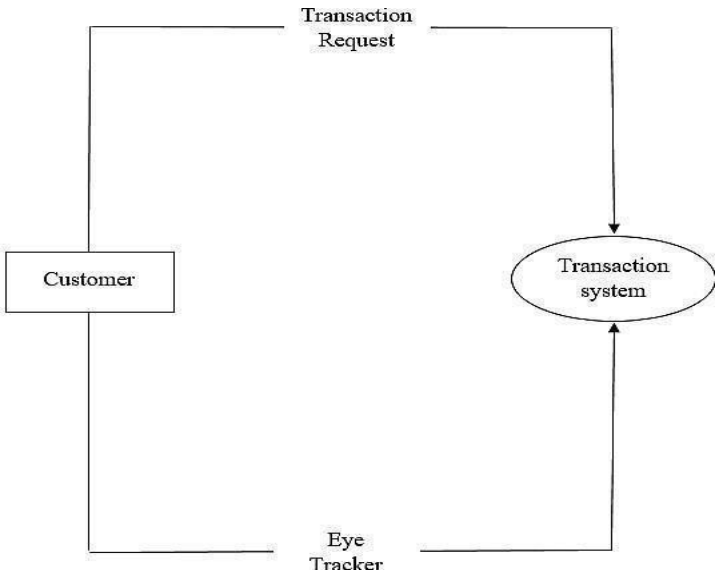

Figure 4.4 Level – 0 Data Flow Diagram

The DFD Zero level is shown in the above Figure 4.4. Customer will start the process by requesting the transaction system to make the transaction. Then the eye tracker will track the eye pupil movement of the consumer requesting for the transaction and locate the pupil center on the coordinate system and based on this the password would be updated. This updated pin would be sent to the transaction system for further authentication process.

4.3.2 Level – 1 Data Flow Diagram

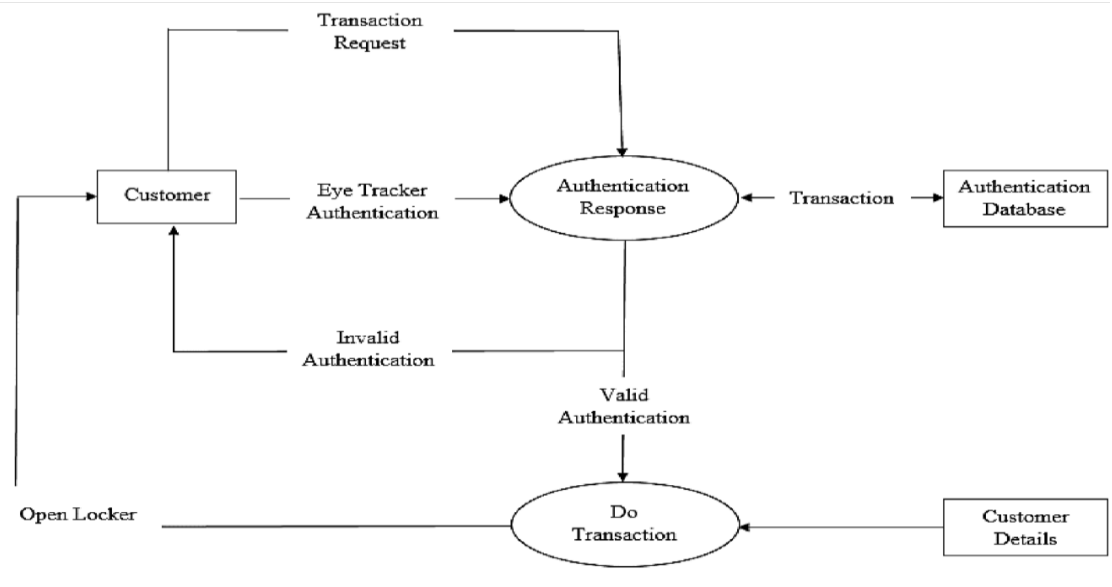

Figure 4.5 Level – 1 Data Flow Diagram

The DFD First level is shown in the above Figure 4.5. Transaction system after receiving the password will send the password for the authentication response system for further authentication process then the eye tracker authentication will request the authentication response to make the authentication.

The Authentication system will make the transaction with the Authentication database where the password corresponding to the customer would be stored. If the received password would get matched with the stored password, then the system allows the customer to do transactions i.e., system activates the relay to open the locker. If the authentication process fails then the signal is sent to customer as invalid password.

4.4 Flowchart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

The system implementation will be explained with the help of the below Figure 4.6.

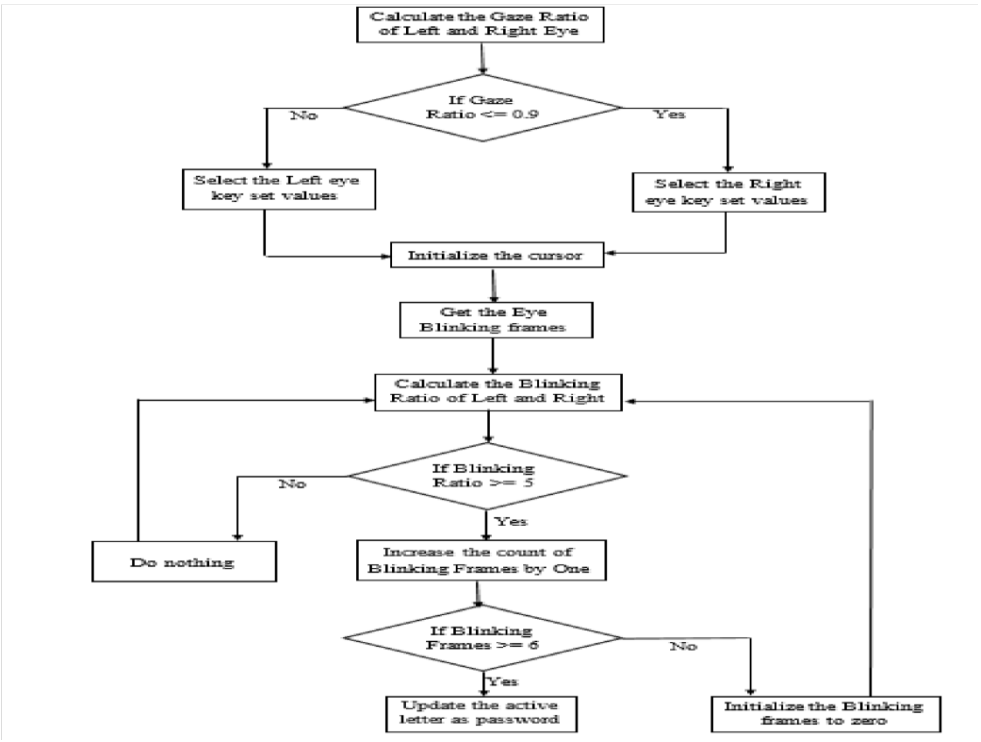

Figure 4.6 Flowchart for proposed system

Chapter 5

IMPLEMENTATION

5.1 Overview of System Implementation

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

The purpose of this system is to enter and identify gaze-based PINs using a smart camera through real-time eye detection and tracking. NI Vision Builder and Open CV are used for eye tracking and for recording eye center location on board the camera real-time. The smart camera allows on-board data processing and collection. Non-contact PIN based authentication adds a layer of security to physical PIN entries and are expected to reduce the vulnerability of the authentication process. The proposed system is easy to implement with real time systems through a portable device with a low cost and more secured features**.**

Block Diagram

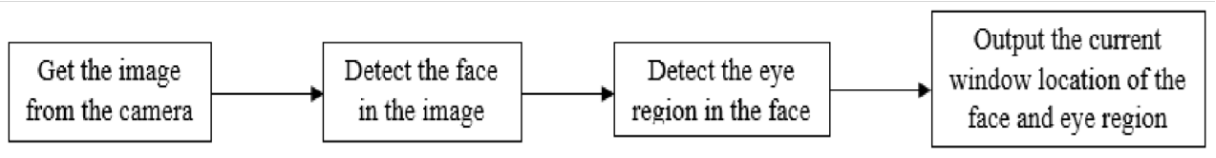

Figure 5.1 Block diagram of Eye detection module

5.2 Algorithms

5.2.1 Eye detection

Eye detection module is used to detect the eye region in the given image. Haar cascade algorithm is used to achieve the task.

Haar cascade algorithm

Haar cascade algorithm is the machine learning object detection algorithm used to identify objects in an image or video based on the concept of features.

Haar cascade algorithm has four steps:

1. Haar Feature Selection
2. Creating integral image
3. Ada boost training
4. Cascading Classifiers

1. Haar Feature selection:

Haar like features are the digital image features used in object recognition. Haar feature selection is a cascade classifier. Initially the algorithm needs to train with lots of positive (images of face) and negative (images without face) images to train the classifier. Then the feature is extracted from it. For this, Haar features shown in Figure 5.2:

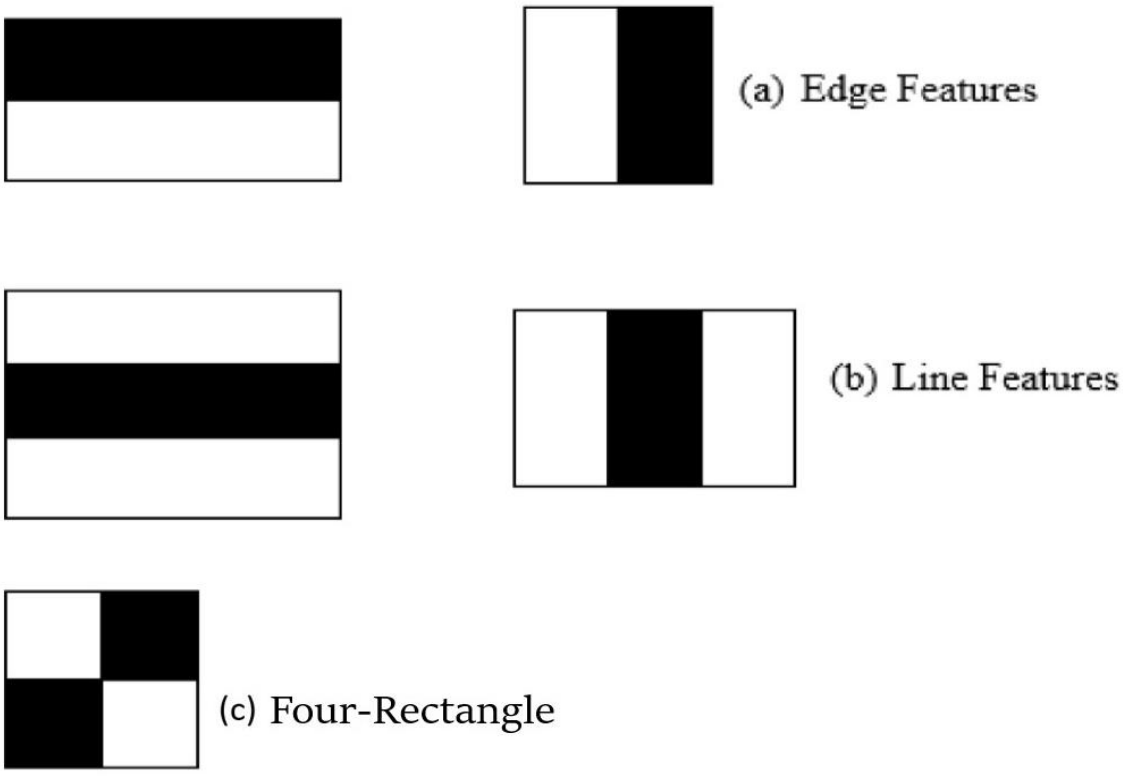

Figure 5.2 Three different Haar features

Each feature is a single value obtained by subtracting sum of pixels under white rectangle from the sum of pixels under black rectangles.

The eye region in the face is detected by using the edge feature detection since the eye region is darker than the other region that is nose and checks.

2. Creating Integral Images:

Integral images are those images in which the pixel value at any (x, y) location is the sum of the all pixel values present before the current pixel. Its use can be understood by the following example Figures 5.3, 5.4, 5.5, 5.6 and 5.7:

5	4	3	8	3
3	9	1	2	6
9	6	0	5	7
7	3	6	5	9
1	2	2	8	3

5	9	12	20	23
8	21	25	35	44
17	36	40	55	71
24	46	56	76	101
25	49	61	89	117

Figure 5.3 Image on the left and the integral image on the right

5	4	3	8	3
3	9	1	2	6
9	6	0	5	7
7	3	6	5	9
1	2	2	8	3

Figure 5.4 5X5 representation of the image

Calculate the average intensity over the area highlighted:

5	4	3	8	3
3	9	1	2	6
9	6	0	5	7
7	3	6	5	9
1	2	2	8	3

Figure 5.5: Region for addition

Normally you'd do the following: 9 + 1 + 2 + 6 + 0 + 5 + 3 + 6 + 5 = 37, 37 /9 = 4.11

This requires a total of 9 operations. Doing the same for 100 such operations would require: 100 * 9 = 900 operations.

Now, first make integral image of the preceding image:

5	9	12	20	23
8	21	25	35	44
17	36	40	55	71
24	46	56	76	101
25	49	61	89	117

Figure 5.6 Integral image for the preceding image making the image requires total 56 operations

Again, focus on the highlighted portion:

5	9	12	20	23
8	21	25	35	44
17	36	40	55	71
24	46	56	76	101
25	49	61	89	117

Figure 5.7 Integral image with highlighted portion

To calculate the average intensity:

(76 – 20) – (24 – 5) = 37, 37 / 9 = 4.11

This required a total of 4 operations.

To do this for 100 such operations, we would require: 56 + 100 * 4 = **456 operations**.

For just a hundred operations over a 5×5 matrix, using an integral image requires about 50% less computations. Imagine the difference it makes for images and other such operations.

Creation of an integral image changes other sum difference operations by almost *O(1)* time complexity, thereby decreasing the number of calculations.

It simplifies the calculation of the sum of pixels—no matter how large the number of pixels—to an operation involving just four pixels. Nice, isn't it? It makes things superfast.

3. Ada boost Training:

This process selects only those features known to improve the predictive power of the model, reducing dimensionality and potentiality improving execution time as irrelevant features need not be computed.

During this window of the specific size is moved over the image and for each sub section of the image the haar features are calculated. The difference is then compared to a learned threshold that separates non-object from objects.

4. Cascade Classifier:

It consists of collection of storage, where each storage is an ensemble of week learners. The week learners are simple classifiers called decision stumps. Each stage is trained using a technique called boosting.

Boosting provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

While the window slides, it detects whether the region is positive or negative. If the positive region is detected, then it considers that the object is found and passes it on to the next stage. If the negative region is detected, then the sliding window considers the next smaller region of the image.

After the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

5.2.2 Feature Detection

The window location from the above module is considered to detect the key facial structure of the in face and locate the facial structures with the specific (x, y) co-ordinates values. Then the co-ordinate values of the left and right eyes are considered and the polygon is drawn over the eye region.

Facial Landmark detector is used to achieve the above process.

Facial Landmark Algorithm

• Input the window location where the face and eye region is found.

• Detect the key facial structures in the image.

• Locate the key facial structures with specific (x, y) coordinates.

- Start with 1 for first (x, y) co-ordinate.
- End with 68 for the last (x, y) co-ordinate.

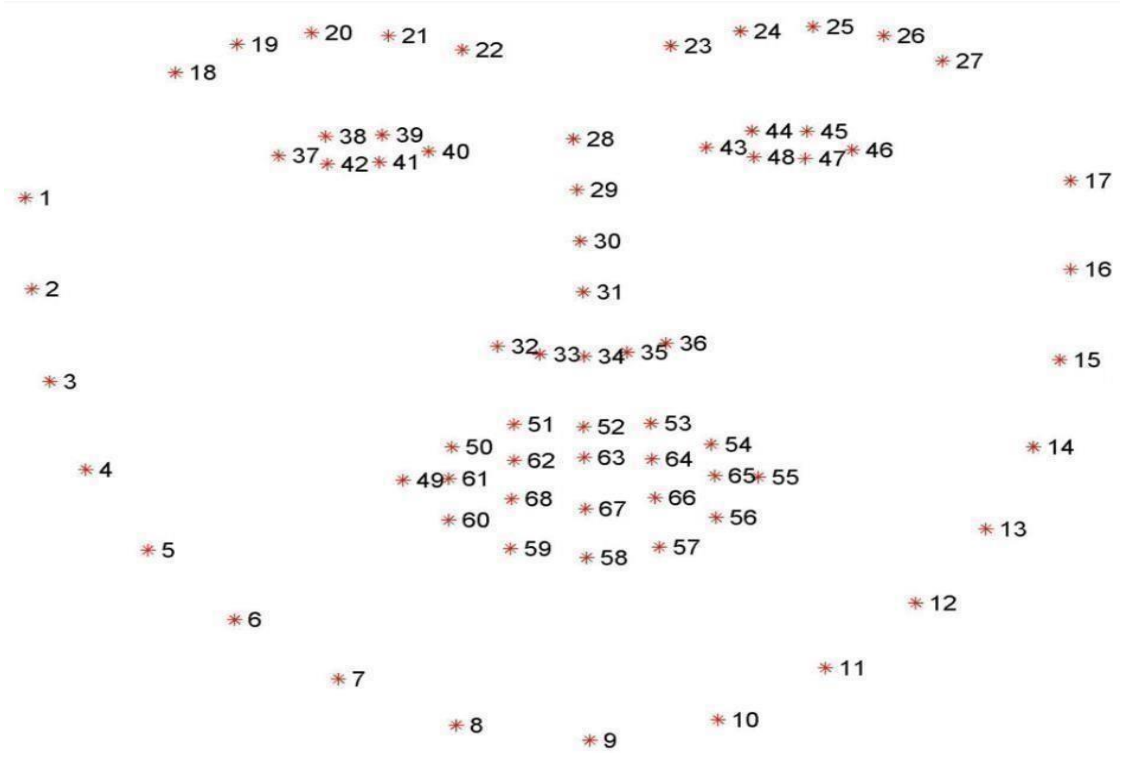

Figure 5.8 68 Facial landmark points

To draw the polygon over the eye region:

Get the co-ordinates value of the left and right eye.

```
eye_region=np.array([(facial_landmarks.part(eye_points[36]).x,
facial_landmarks.part(eye_points[36]).y),
(facial_landmarks.part(eye_points[37]).x,facial_landmarks.part(eye_points[37]).y),
(facial_landmarks.part(eye_points[38]).x,facial_landmarks.part(eye_points[38]).y),
(facial_landmarks.part(eye_points[39]).x,facial_landmarks.part(eye_points[39]).y),
(facial_landmarks.part(eye_points[40]).x,facial_landmarks.part(eye_points[40]).y),
(facial_landmarks.part(eye_points[41]).x,facial_landmarks.part(eye_points[41]).y)],np.int32)
```

Table 5.1 Coordinates for left and right eye

	x	y		x	y
36	403	321	42	495	320
37	415	313	43	508	311
38	430	313	44	521	311
39	443	326	45	533	316
40	430	326	46	523	321
41	415	326	47	509	322
Left eye values				Right eye values	

Based on the above pixel values the polygon is drawn over the eye region.

Figure 5.9 Polygon drawn over the eye region

5.2.3 Eye Tracking

In this module, continuously the eye movement is tracked to obtain the Gaze Ratio and based on the gaze ratio the respective keyboard will be displayed. Then the eye blinking ratio will be calculated to update the respective digit as the PIN.

Algorithm: To calculate the gaze ratio

1. Input the pixel values of the eye region.
2. Get only the eye region.
3. Divide each eye region into left and right part.
4. Convert the eye image into grayscale.
5. Get the number of white pixels on both sides i.e., on the left side and right side of each eye.
6. Calculate the Gaze Ratio:

   Gaze Ratio of left eye = Number of white pixels on right side/Number of white pixels on left side

   Gaze Ratio of right eye = Number of white pixels on right side/Number of white pixels on left side

   Gaze Ratio = Gaze Ratio of left eye + Gaze Ratio of righteye/ 2
7. If Gaze Ratio <= 0.9 then select the right keyboard

   Else then select the left keyboard

Working

The table values are taken as the input here. The corresponding eye region is converted into gray scale which is shown in the below Figure 5.10.

Figure 5.10 Gray Scale Image

Figure 5.11 The menu keyboard

Then the midpoint between 37th and 38th co-ordinate is calculated and the midpoint of 40th and 41th co-ordinate is calculated and those midpoints are joined to divide the eye region into two parts.

Figure 5.12 Vertical line drawn over the image to divide the eye region

The gazeration of both left and right eye will be calculated and then the average of both the left and right eye is taken to obtain the Gaze ratio.

Calculation of the gaze ratio is shown below:

Gaze Ratio of the left eye:

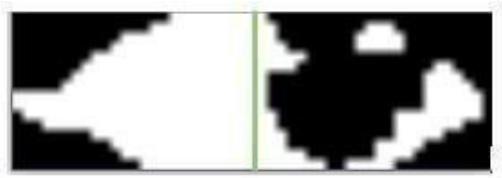

Figure 5.13 Gaze of the left eye

Number of white pixels on left side = 7187

Number of white pixels on right side = 2701

Gaze ratio of Left Eye = Number of white pixels on right side / Number of white pixels on left side

=2701/ 7187

= 0.376

Gaze Ratio of the right eye:

Figure 5.14 Gaze of the right eye

Number of white pixels on left side = 4759

Number of white pixels on right side = 398

Gaze ratio of right Eye = Number of white pixels on right side/ Number of white pixels on left side

= 398/4759

= 0.0836

Gaze Ratio = Gaze Ratio of left eye + Gaze Ratio of right eye/ 2

= 0.376 + 0.0836/ 2

= 0 .2297

The gaze ratio obtained above is less than 0.9. Similarly, the gaze ratio is obtained for 15 frames continuously. If all the obtained gaze ratio of the 15 frames is less than 0.9 then the right keyboard is selected. If the obtained Gaze ratio is more than then the left keyboard is selected.

Since the obtained value is less than 0.9 the right keyboard is selected.

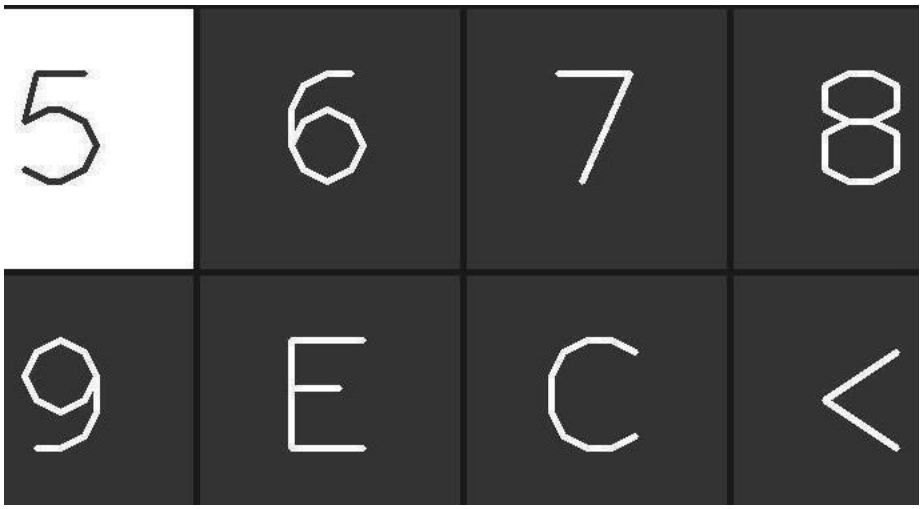

Figure 5.15 Right Keyboard

When the keyboard is displayed the eye blinking is calculated to update the PIN.

Algorithm: To calculate the Blinking Ratio

1. Input is the co-ordinate values of the eye region.
2. Obtain the horizontal and vertical of left eye:
    i. Calculate the midpoint of 37th, 38th co- ordinate and 40th, 41stcoordinate.
    ii. Join the points to the vertical line.
    iii. Join the 36th and 39th point to get the horizontal line.

3. Obtain the horizontal and vertical of righteye:

iv. Join the 42th and 47th point to get the horizontal line.

v. Join the points to the vertical line.

vi. Join the42th and 47th point to get the horizontal line.

4. Calculate the Blinking Ratio:

i. Blinking Ratio of left eye:

Blinking ratio of left eye = length of the horizontal line/ length of the vertical line

ii. Blinking Ratio of right eye:

Blinking ratio of right eye = length of the horizontal line/ length of the vertical line

iii. Blinking ratio = Blinking ratio of left eye + blinking ratio of right eye) / 2

5. Initialize the Blinking frames to zero

6. If Blinking ratio >=5

Increase the blinking frames value by one

Else

Do nothing

7. If Blinking frames == 6, Then update the digit as PIN.

Working

Calculating the midpoints to obtain the horizontal and vertical line.

Consider the left eye co-ordinate points as shown in above Table 5.1.

For obtaining the vertical line

Midpoint of 37th, 38th co-ordinate = ((415 + 430) / 2), ((313 + 313) / 2)

= 422.5, 313

Midpoint of 40th, 41st co-ordinate = ((430 + 415) / 2), ((326 + 326) / 2)

= 422.5, 326

Join the above points i.e., (422.5,313) and (422.5,326) to get the vertical line .

For obtaining the horizontal line, join the 36th and 37th co-ordinate i.e., (403,321) and (443,323) to get the horizontal line.

The Figure 5.16 shown below will show how the horizontal and vertical line is drawn on the eye.

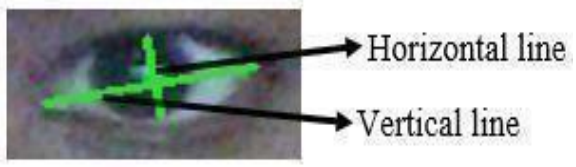

Figure 5.16 The horizontal and vertical line drawn on the eye

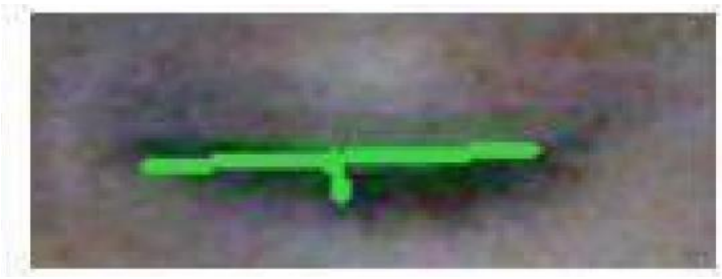

Figure 5.17 Horizontal and vertical line when eye is blinked

The calculation of the blinking ratio of the above Figure 5.17 is shown below.

Calculating the blinking ratio:

Blinking Ratio of left eye:

Length of the horizontal line: 48.0936

Length of the vertical line: 8.0

Blinking ratio of left eye = (48.0936 / 8.0)

= 6.0117

Blinking Ratio of right eye:

Length of the horizontal line: 49.09175

Length of the vertical line: 10.0498

Blinking ratio of right eye = (49.09175 / 10.0498)

= 4.8848

Blinking ratio = (Blinking ratio of left eye + blinking ration of right eye) / 2

= (6.0117 + 4.8848) / 2

= 5.448

Here the blinking ratio is greater than 5. This blinking ratio computation is done continuously for 6 frames if in all the frames the blinking ratio obtained is greater than 5 then the digit pointed by the cursor will be updated as the PIN, else the blinking frames will be made zero and the process repeats.

5.3 Code Snippets

5.3.1 Capturing Images from the Camera

The images are captured from the Camera using OpenCV which is used for image processing.

```
def fun():
    # Import OpenCV2 for image processing
    import cv2
     import time
```

```
vid_cam = cv2.VideoCapture(0)
# Detect object in video stream using Haarcascade Frontal Face
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# For each person, one face id
face_id = 2
# for multiple person different ids
count = 0
while(True):
    _, image_frame = vid_cam.read()
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)
    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    # Loops for each faces
    for (x,y,w,h) in faces:
        # Crop the image frame into rectangle
        cv2.rectangle(image_frame, (x,y), (x+w,y+h), (255,0,0), 2)
        # Increment sample face image
        count += 1
        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User."+ str(face_id) + '.' + str(count) + ".jpg",
gray[y:y+h,x:x+w])
        # Display the video frame, with bounded rectangle on the person's face
        cv2.imshow('Frame', image_frame)
    # To stop taking video, press 'q' for at least 100ms
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
    # If image taken reach 100, stop taking video
    elif count>50:
        break
# Stop video
vid_cam.release()
# Close all started windows
cv2.destroyAllWindows()
##    os.system("python face_datasets.py")
```

5.3.2 Face Recognition

Face recognition can be obtained through Haar Cascade method which detects the face.

```
def fun1():
    import cv2
    # Import os for file path
    ##    import cv2, os
    # Import numpy for matrix calculation
    import numpy as np
    # Import Python Image Library (PIL)
    from PIL import Image
    # Create Local Binary Patterns Histograms for face recognization
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    # Using prebuilt frontal face training model, for face detection
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
    # Create method to get the images and label data
    def getImagesAndLabels(path):
        # Get all file path
        imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
        # Initialize empty face sample
        faceSamples=[]
        # Initialize empty id
        ids = []
        # Loop all the file path
        for imagePath in imagePaths:
            # Get the image and convert it to grayscale
            PIL_img = Image.open(imagePath).convert('L')
            # PIL image to numpy array
            img_numpy = np.array(PIL_img,'uint8')
            # Get the image id
            id = int(os.path.split(imagePath)[-1].split(".")[1])
            print(id)
            # Get the face from the training images
            faces = detector.detectMultiScale(img_numpy)
```

```
        # Loop for each face, append to their respective ID
          for (x,y,w,h) in faces:
              # Add the image to face samples
              faceSamples.append(img_numpy[y:y+h,x:x+w])
              # Add the ID to IDs
              ids.append(id)
      # Pass the face array and IDs array
      return faceSamples,ids
  # Get the faces and IDs
  faces,ids = getImagesAndLabels('dataset')
  # Train the model using the faces and IDs
  recognizer.train(faces, np.array(ids))
  # Save the model into trainer.yml
  recognizer.write('trainer/trainer.yml')
  ##    os.system("python training.py")
```

5.3.3 Eye Detection

Eye detection is used to detect the eye region in the given image. Haar cascade algorithm is used to achieve the task.

```
def eye():
    cap = cv2.VideoCapture(0)
    detector = dlib.get_frontal_face_detector()
    predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
    # Keyboard settings
    ##keyboard = np.zeros((600, 1000, 3), np.uint8)
    keyboard = np.zeros((1000, 800, 3), np.uint8)
    keys_set_1 = {0: "1", 1: "2", 2: "3",
                  3: "4",4: "5", 5: "6",
                  6: "7", 7: "9",8: "3",
                  9: "8", 10: "9",11:"0",12:"2"}
```

To get the coordinates of the eye we use Facial Landmark Algorithm which locate the facial structures with the specific (x, y) co-ordinates values. Then the co-ordinate values of the left and right eyes are considered and the polygon is drawn over the eye region.

```
def eyes_contour_points(facial_landmarks):
        left_eye = []
        right_eye = []
        for n in range(36, 42):
            x = facial_landmarks.part(n).x
            y = facial_landmarks.part(n).y
            left_eye.append([x, y])
        for n in range(42, 48):
            x = facial_landmarks.part(n).x
            y = facial_landmarks.part(n).y
            right_eye.append([x, y])
        left_eye = np.array(left_eye, np.int32)
        right_eye = np.array(right_eye, np.int32)
        return left_eye, right_eye
```

5.3.4 Eye Gaze Ratio and Blinking Ratio

Eye movement is tracked to obtain the Gaze Ratio and based on the gaze ratio the respective keyboard will be displayed. Then the eye blinking ratio will be calculated to update the respective letter as the password.

```
def get_blinking_ratio(eye_points, facial_landmarks):
        left_point=(facial_landmarks.part(eye_points[0]).x,
facial_landmarks.part(eye_points[0]).y)
    right_point=(facial_landmarks.part(eye_points[3]).x,
facial_landmarks.part(eye_points[3]).y)
    center_top=midpoint(facial_landmarks.part(eye_points[1]),
facial_landmarks.part(eye_points[2]))
    center_bottom=midpoint(facial_landmarks.part(eye_points[5]),
facial_landmarks.part(eye_points[4]))
      ##    hor_line = cv2.line(frame, left_point, right_point, (0, 255, 0), 2)
      ##    ver_line = cv2.line(frame, center_top, center_bottom, (0, 255, 0), 2)
   hor_line_length= hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
   ver_line_length=hypot((center_top[0] - center_bottom[0]), (center_top[1]-
center_bottom[1]))
        ratio = hor_line_length / ver_line_length
        return ratio
```

Chapter 6

TESTING

6.1 Introduction

Software testing is a detailed analysis conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

• Meets the requirements that guided its design and development,

• Responds correctly to all kinds of inputs,

• Performs its functions within an acceptable time,

• It is sufficiently usable,

• Can be installed and run in its intended environments, and

• Achieves the general result its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

Here in this chapter we focus on various testing techniques like unit testing, integration testing, system testing, validation testing and user acceptance testing.

6.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software unit software applications. It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test as specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

TEST CASE - 1

In the following test case, the image of the face is captured and detected.

Table 6.1 Unit testing for detecting face region

Name of the test:	Checking the image for face
Item/Feature to be tested:	Image being captured
Sample Input:	Image
Expected Output:	Face region
Actual Output:	Face region detected
Remark:	Test passed

TEST CASE - 2

In the following test case, the image of the eye is captured and detected.

Table 6.2 Unit testing for detecting eye region

Name of the test:	Eye region detection
Item/Feature to be tested:	Image being captured
Sample Input:	Output image from test 1
Expected Output:	Eye region
Actual Output:	Eye region detected
Remark:	Test passed

6.3 Integration Testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

We can see two test cases for Integration Testing in Table 6.3 and 6.4. We get the coordinate values of the eye region and the values of gaze ratio that are calculated, respectively. Both the integration testing test case pass as the respective values are output.

TEST CASE 3

Table 6.3 Integration testing for coordinating values of Eye region

Name of the test:	Coordinate values of eye region
Item/Feature to be tested:	Image being captured
Sample Input:	Output image from test 2
Expected Output:	Coordinate Values of Eye region
Actual Output:	Matrix of coordinate values (i.e., the pixel values) of both left and right eye region
Remark:	Test passed

TEST CASE 4

Table 6.4 Integration testing for selection of keypad through gaze ratio

Name of the test:	Selection of keyboard through calculation of gaze ratio
Item/Feature to be tested:	Set of continuous image captured
Sample Input:	Image
Expected Output:	Values of Gaze Ratio
Actual Output:	Continuous values of gaze ratio calculated for 15 frames to select the keypad
Remark:	Test passed

6.4 Validation Testing

Validation tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Validation testing is centered on the following items:

- Valid Input: Identified classes of valid input must be accepted.
- Invalid Input: Identified classes of invalid input must be rejected.
- Functions: Identified functions must be exercised.
- Output: Identified classes of application outputs must be exercised.
- Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

In addition, systematic coverage pertaining to identifying business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before Validation testing is complete, additional tests are identified and the effective value of current tests is determined.

TEST CASE - 5

Table 6.5 Validation testing for selection of keyboard through gaze ratio

Name of the test:	Selection of letter
Item/Feature to be tested:	Set of continuous image captured
Sample Input:	Image
Expected Output:	Values of Blinking ratio
Actual Output:	Continuous values of blinking ratio calculated for 6 frames to select the activated digit in keypad
Remark:	Test passed

Chapter 7

DISCUSSION OF RESULTS

In this chapter, we briefly discuss on the results obtained in our project in various cases. We attach pictures of the processing each case to explain in an enhanced way for better understanding.

When we first run our project, the first window that opens is shown in Figure 7.1 which has “WELCOME” and the title of the project. To proceed, we click on the “Click Here” button.

Once the “Click Here” button is pressed, we go to the main page of the project as shown below in Figure 7.2.

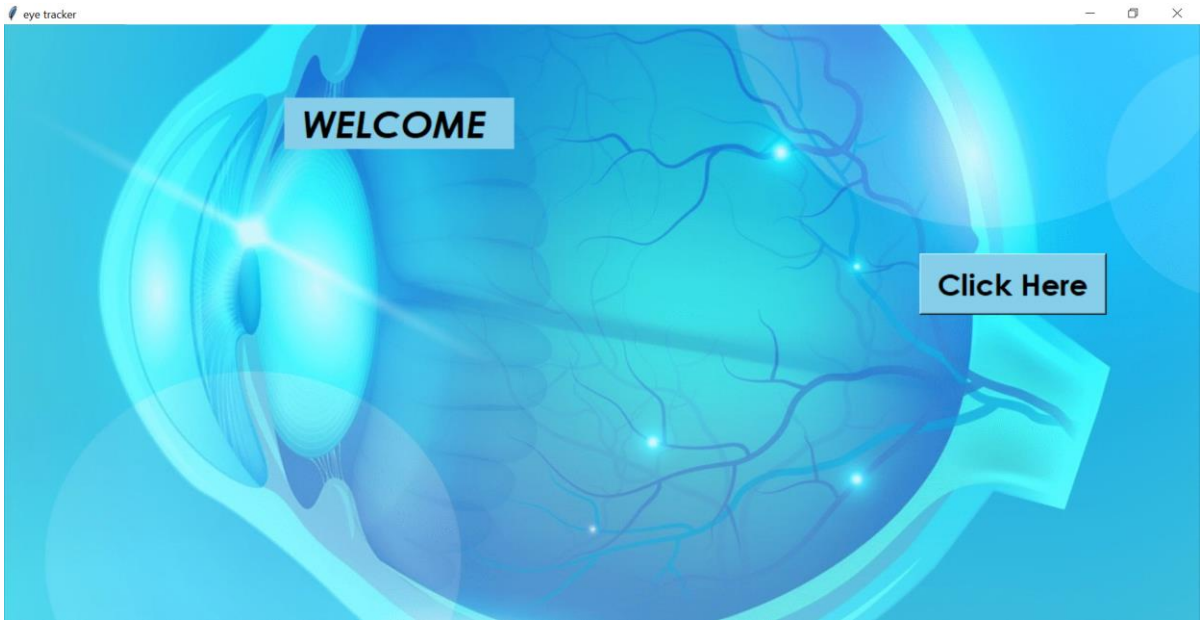

Figure 7.1 Welcome Window

The Figure 7.2 displays the various options such as “Face Capturing” which is used to detect the user’s face and capture images. The “Training Face ID” option trains the model for the captured images and assigns IDs for every image. “Face recognition and password detection” is the main part of our project. It is used to recognize the face and if successful, the eye is detected and the PIN can be entered by the user.

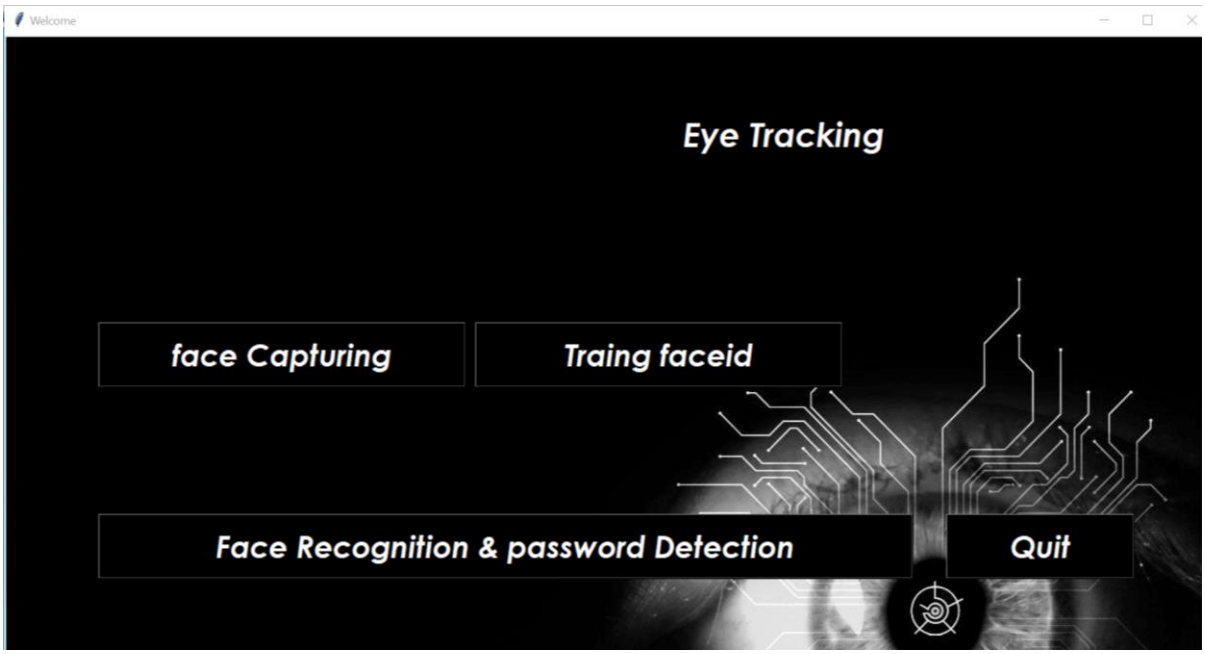

Figure 7.2 Main Window

The 3 options are illustrated below in Figures 7.3, 7.4 and 7.5.

When the first button i.e. “Face Capturing” is pressed, the video frame opens and clicks images of the user.

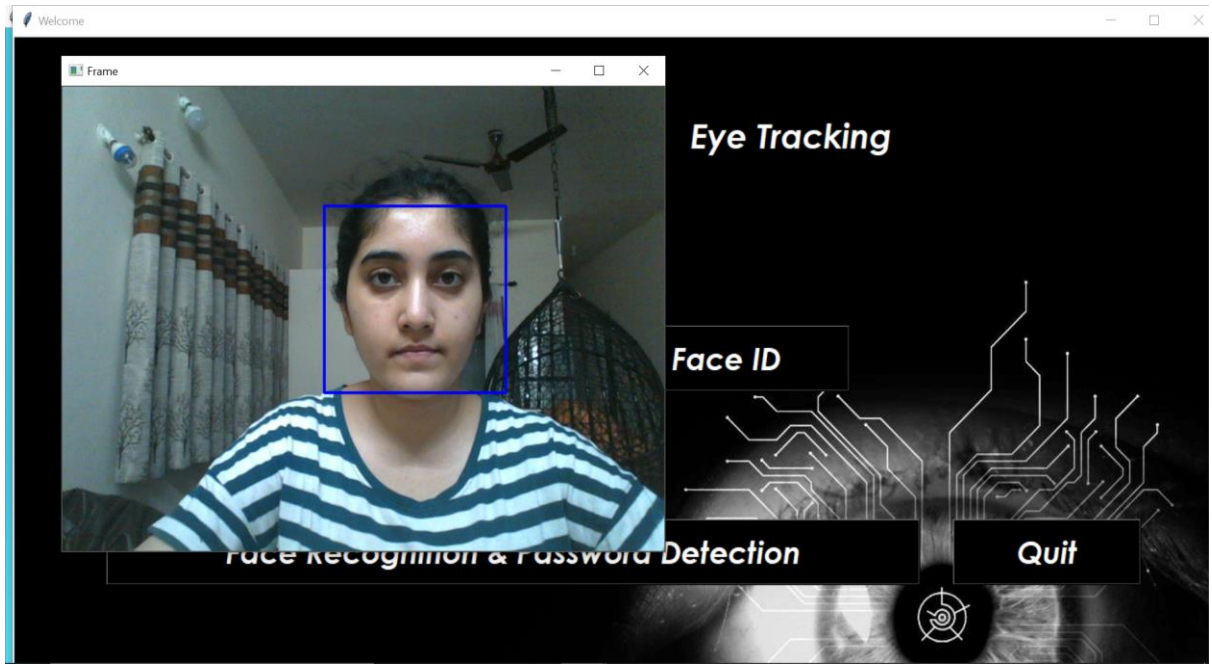

Figure 7.3 Face Capturing

After the face capturing is done, then the "Training Face ID" button is pressed and this trains the captured frames and an ID is given to every frame/image. When seen in the datasets folder, the IDs are given to them as shown in Figure 7.4.

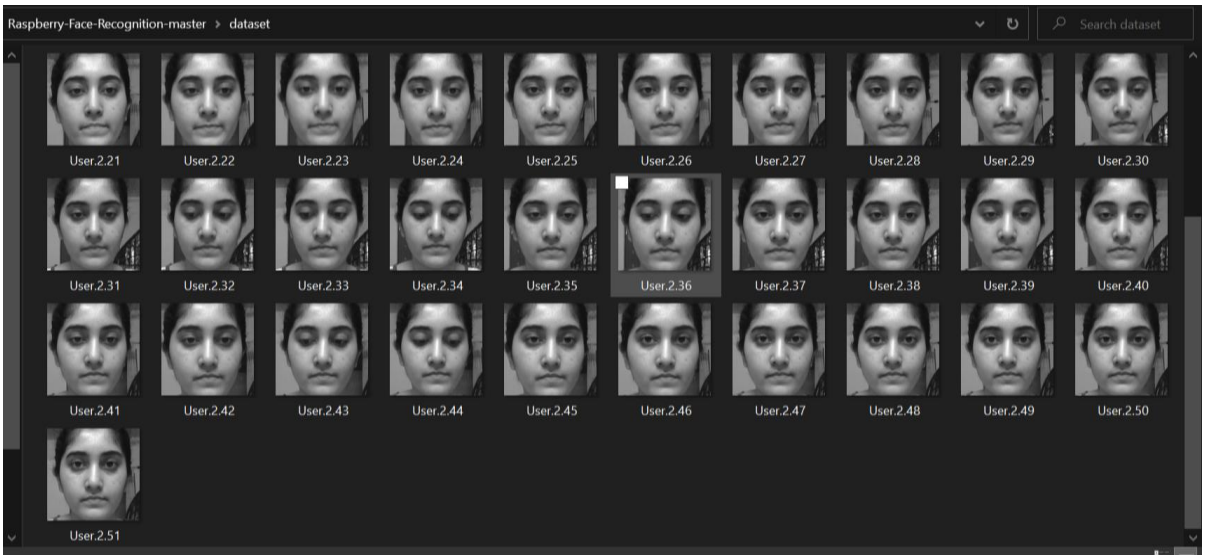

Figure 7.4 Dataset folder

Next, the "Face recognition and password detection" button is pressed for further PIN authentication.

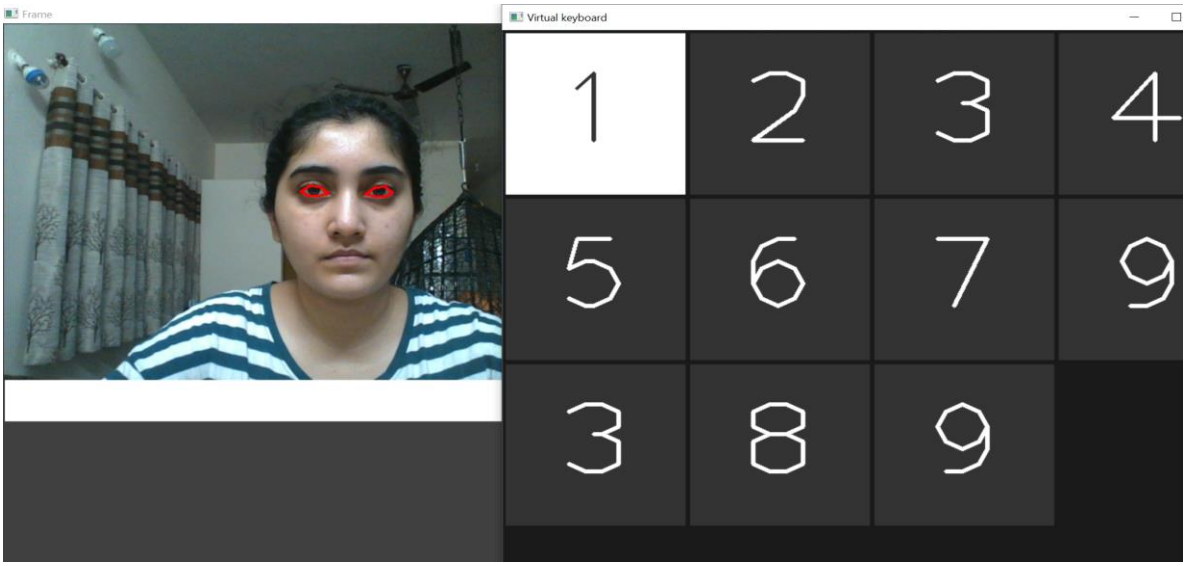

Figure 7.5 Video Frame and Virtual Keypad

When the Face Recognition is authenticated and verified, the virtual keypad and the video frame windows come up. We can see the red highlight which indicates that the eyes are open.

When the digits in the keypad appear in white, as shown in Figure 7.6, and if it happens to be a digit in the user's PIN, then there must be some eye movement i.e. closing of the eyes, so that the highlight of the eye turns green and the digit is selected.

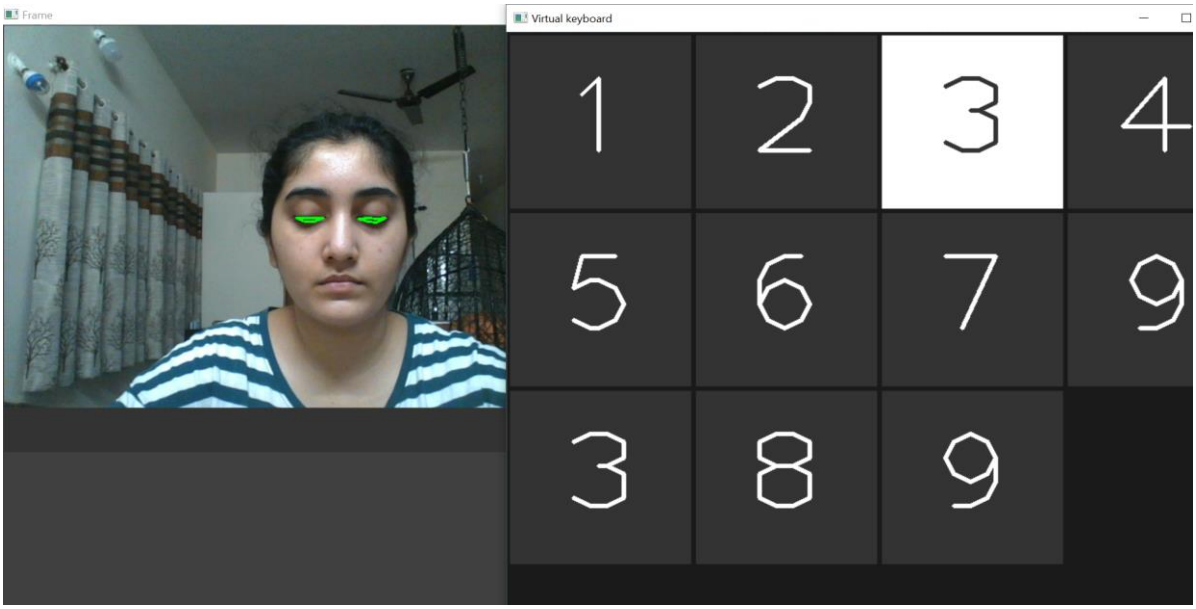

Figure 7.6 Selecting digits via eye movement

Now, if the PIN is verified, an audio plays which says the PIN matches and the prompt tells that the PIN is matched and all the sensitive numbers and data is revealed.

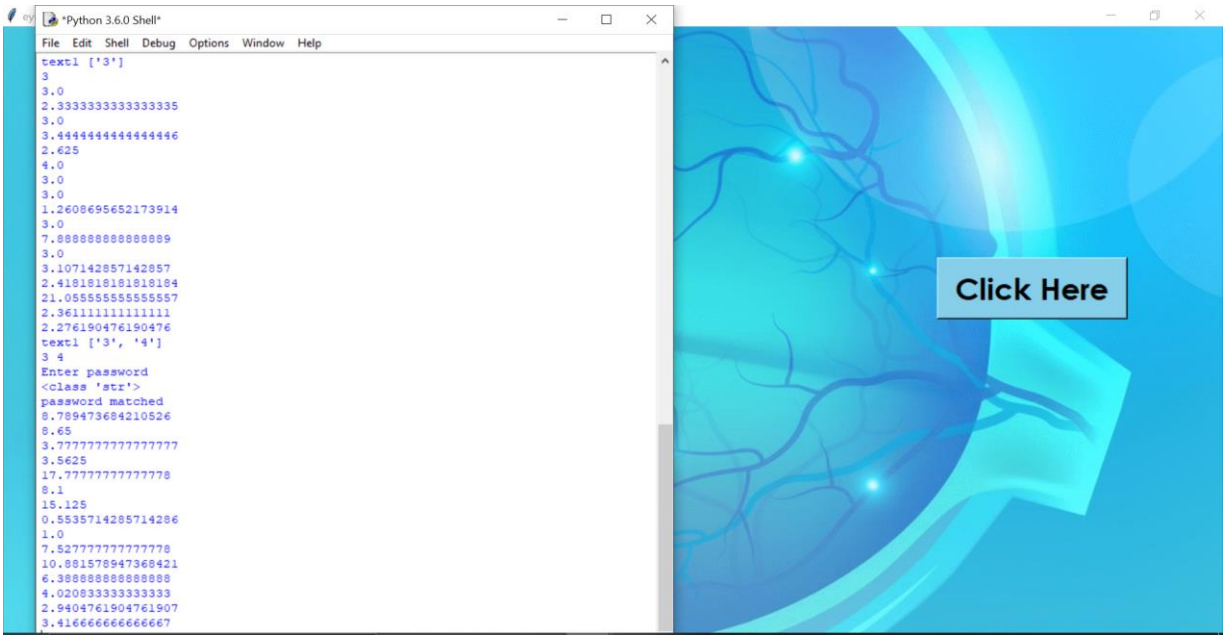

Figure 7.7 PIN Match

If the PIN doesn't match, the prompt says 'PIN not matched'.

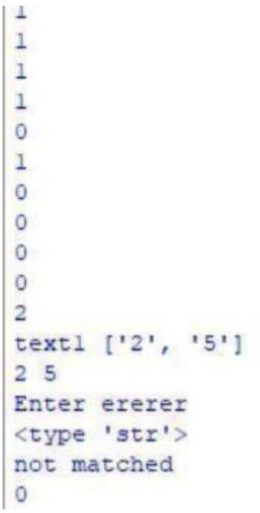

Figure 7.8 PIN not matched

Therefore, this is how the authentication of the PIN is done. We can quit from the application by clicking on the "Quit" button.

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

A smart-camera based eye-tracking system has been incorporated as a new application for gaze-based PIN identification. The system has been successfully tested with numbers, and can be extended to character and digit combination password entry. The user password from various attacks like shoulder surfing and thermal tracking and it is also helpful for the physically disabled persons who are not able to enter password manually.

The stability of the user's gaze will affect the accuracy of the detected pins, and must be accounted for. Currently, the PIN identification is accomplished after real-time eye tracking and eye center computations and recording are completed. Future work includes incorporating the PIN identification algorithm into the-real-time framework for all-in one password identification system. In addition, gaze-based password entry can be extended to mobile devices and other camera-based systems and also in the following places.

In banks the lockers are used for depositing valuables like gold, jewellery and important documents. It can be hired by individuals, firms, trusts, companies etc. These lockers are secured through PINs. One has to enter the PIN manually to get access to his/her locker ,by entering the manually leads to password attacks such as shoulder surfing and thermal tracking. This application can be used in banks to overcome these kinds of attacks by entering the PIN to the locker through the eye movements. If the entered PIN is matched with the existing PIN then the locker is opened so that the individual can use the locker.

Physically challenged people are not able to enter the PIN manually i.e., through the physical movements. This application is useful for those people where the PIN can be entered through the eye tracker system which captures the eye movements continuously to calculate the gazeration and blinking ratio of the eye which helps the user to enter the PIN through their eye movements. This helps to overcome the problem of entering the PIN manually by the physically challenged persons. Colleges, Industries, Company etc.

In companies, industries or institutes there would be some official data that has to be stored securely. For securely storing those data they will apply the PIN so that only authorized persons can access those systems in such cases. If we are trying to enter the PIN manually then there would be more chances of vulnerability. To avoid this they can use the eye tracking system to enter the PIN through their eye pupil movements to get the access for that data.

In homes we use the locker for keeping valuable things such as ornaments, land papers that have to be kept securely .Our system will be helpful for situations so that the user can enter the PIN through eye pupil movements to open the lock.

ATMs in day to day activities, people who are using ATM cards exponentially and at the same ATM card theft activities are also increasing. Basically, entering the pin manually in public places may lead to password vulnerabilities. This can be overcome through our project where the PIN for the ATM system is entered through the eye pupil movement.

Smart phones, laptops We may have some important documents such as soft copy in our smartphones or our laptops or in our system ,we may have set the PINS to access our smartphones, laptop etc .Here we can use the proposed applications for entering the pins to our system.

8.2 Future Work

The main enhancement which can be done is by making sure the system is even more safer for military operations personnel so that small errors like forgetting your passwords and leakage of the same can be overcome by this method. Usage of this technology will be helpful for each and every sector of this corporate world and therefore making sure the resources and equipment required for the same is made extensively available.

Applications of this method can be used further for high end security systems by making some more improvements and research at higher level in all of the experiments in which the subjects were seated between 1 and 2 feet from the camera, it never took more than three involuntary blinks by the user before the eyes were located successfully.

Another improvement is this system's compatibility with inexpensive inbuilt laptop cameras. We can make sure to add the high-resolution colour video CCD camera for better resolution. Perhaps most importantly, external USB cameras which support a higher real-time frame rate of 30 frames per second.

The reliability of the system has been shown with the high accuracy results reported in the previous section. The experiments indicate that the system performs really well in extreme lighting conditions (i.e. when all lights ar0e turned off, leaving the computer monitor as the only light source, and with a lamp aimed directly at the video camera). The accuracy percentages in these cases were approximately the same as those that were retrieved in normal lighting conditions.

REFERENCES

[1] E. Gupta, M. Agarwal and R. Sivakumar, "Blink to Get In: Biometric Authentication for Mobile Devices using EEG Signals",s ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6, doi: 10.1109/ICC40277.2020.9148741.

[2] V. Aharonson, V. Y. Coopoo, K. L. Govender and M. Postema, "Automatic pupil detection and gaze estimation using the vestibulo-ocular reflex in a low-cost eye-tracking setup", in SAIEE Africa Research Journal, vol. 111, no. 3, pp. 120-124, Sept. 2020, doi: 10.23919/SAIEE.2020.9142605.

[3] Khan, W.; Hussain, A.; Kuru, K.; Al-askar, H. Pupil Localisation and Eye Centre Estimation Using Machine Learning and Computer Vision. *Sensors* 2020, *20*, 3785

[4] Ahmad Aljaafreh, Murad Alaqtash, Naeem Al-Oudat, Jafar Abukhait, and Ma'en Saleh, "A Low-cost Webcam-based Eye Tracker and Saccade Measurement System", in INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING, Volume 14, 2020.

[5] A. Siripitakchi, S. Phimoltares, A. Mahaweerawat, 2017, "Eye- Captcha: An Enhanced Captcha Using Eye Movement", 3rd IEEE International conference on Computer and Communications, pp. 2120 – 2126.

[6] Z. Li, M. Li, P. Mohapatra, J. Han, S. Chen, 2017, "iType: Using Eye Gaze to Enhance Typing Privacy", IEEE Infocom on Computer Communications, pp. 1-9.

[7] C. Meng and X. Zhao, "Webcam-Based Eye Movement Analysis Using CNN," IEEE Access, vol. 5, pp. 19581 – 19587, 2017.

[8] K. Krafka et al., "Eye tracking for everyone", in Proc. IEEE Conf. Comput. Vis. Pattern Recognition, Jun. 2016, pp. 2176_2184.

[9] M. Khamis, F. Alt, M. Hassib, E.V. Zezschwitz, R. Hasholzner, A. Bulling, 2016, "GazeTouchPass: Multimodal Authentication Using Gaze and Touch on Mobile Devices", CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 2156 -2164.

[10] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild", in Proc. IEEE Conf. Comput. Vis. Pattern Recognition, Jun. 2015, pp. 4511_4520.

[11] R. Revathy and R. Bama, 2015, "Advanced Safe PIN-Entry Against Human Shoulder-Surfing," IOSR Journal of Computer Engineering (IOSR-JCE), vol 17, issue 4, ver.II, pp.9-15.

[12] P. Kasprowski and K. Harezlak, "The second eye movement verification and identification competition," in Proceedings of the International Joint Conference on Biometrics (IJCB), Clearwater, FL, USA, 2014, pp 1-6.

[13] D. Rozado, 2013, "Using Gaze Based Passwords as an authentication Mechanism for Password Input", 17th European Conference on Eye Movements(ECEM).

[14] M. Martin, T. Marija and A. Sime, 2013, "Eye tracking recognition based graphical authentication", 7th International Conference on Application of Information and Communication Technologies(AICT), pp. 1 -5.

[15] M. Brooks, C.R. Aragon and O.V. Komogortsev, 2013 "Perceptions of interfaces for eye movement biometrics", 2013 International Conference on Biometrics (ICB), pp.1-8.

[16] G. Pan, L. Sun, Z. Wu and S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Web camera", 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4409068.

[17] P. Kasprowski, "Human identification using eye movements," Praca doktorska, Politechnika OEląska, 2004.

[18] D. W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen, M. B. Stegmann, "Eye typing using Markov and active appearance models", in Proc. IEEE Workshop Appl. Comput. Vis. (WACV), Dec. 2002, pp. 132_136.

[19] V. Paul and M. Jones, 2001, "Rapid object detection using a boosted cascade of simple features" IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol.1

[20] D. D. Salvucci and J. H. Goldberg, "Identifying xations and saccades in eye-tracking protocols" in Proc. Symp. Eye Tracking Res. Appl. (ETRA),2000, pp. 71_78.

[21] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In International Conference on Computer Vision, 1998.