Secure Authentication using Eye Tracking

A dissertation submitted in the partial fulfillment for the award of Degree of
Master of Technology
In
Software Technology

By

Pankaj Singh
(Roll no. 2K12/SWT/12)

Under the guidance of

Prof. Manoj Kumar

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI

DECLARATION

I hereby want to declare that the thesis entitled "**Secure Authentication using Eye Tracking"** which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of degree in **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any institution or university for the award of any degree.

Pankaj Singh
Delhi Technological University, Delhi

CERTIFICATE

Delhi Technological University
(Government of Delhi NCR)
Bawana Road, New Delhi-42

This is to certify that the thesis entitled **"Secure Authentication using Eye Tracking"** done by **Pankaj Singh** (Roll Number: **2K12/SWT/12**) for the partial fulfillment of the requirements for the award of degree of **Master of Technology** Degree in **Software Technology** in the **Department of Computer Engineering**, Delhi Technological University, New Delhi is an authentic work carried out by him under my guidance.

Project Guide:
Prof. Manoj Kumar
Associate Professor
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and respect towards my guide **Prof. Manoj Kumar Department of Computer Engineering.**

I am very much indebted to him for his generosity, expertise and guidance which I received from him while working on this project. Without his support and timely guidance the completion of the project would have seemed a far –fetched dream. In this respect I find myself lucky to have my guide. He has guided not only with the subject matter, but also taught the proper style and techniques of documentation and presentation.

Besides my guide, I would like to thank entire teaching and non-teaching staff in the Department of Computer Engineering, DTU for all their help during my tenure at DTU.

Pankaj Singh
M.Tech. Software Technology
2K12/SWT/12

ABSTRACT

Face recognition is commonly used techniques for secure. However, available face authentication systems are vulnerable to simple spoofing attacks. We can easily spoofing, using low resolution photo representative of those commonly posted online. We can put those images in place of real user and break these secure system easily.

In one project SAFE (Secure Authentication with Face and Eyes), an improved face authentication method that uses a commodity gaze tracker to input a secret. During authentication, the user must not only show her face but also gaze at a secret icon that moves across the screen. Through this way we can ensure that a real user is lively present and through eyes and gaze tracker, he/she can provide a second level of authentication after face detection.

We cannot use this addition gaze tracker hardware in existing mobile/tablet phones. We will demonstrate eye tracking capability using eye pupil tracking method. Software will use eye pupil movement points as input and decide where exactly user is looking in device (Top, Bottom, Top left, Top right etc.)

TABLE OF CONTENTS

List of Acronyms

SAFE	Safe Authentication using Face and Eyes
PDA	Personal Device Assistance
SAET	Safe Authentication using Eye Tracking
OpenCV	Open Source Computer Vision

CHAPTER 1

INTRODUCTION

Authentication involves determining whether a user is, in fact, who he or she claims to be. Authentication can be conducted through the use of logon passwords, single sign-on (SSO) systems, biometrics, digital certificates and a public key infrastructure (PKI).

User authentication is critical to ensure proper authorization and access to systems and services, especially since data theft and information security threats are becoming more advanced. Although authentication cannot completely stop information and identity theft, we can make sure that our resources are protected throughout several authentication methods.

There are three factors of authentication to consider: something you know, such as a user ID and password; something you have, such as a smart card; and something you are, which refers to a physical characteristic, like a fingerprint that is verified using biometric technology. These factors can be used alone, or they can be combined to build a stronger authentication strategy in what is known as two-factor or multifactor authentication.

1.1 Mobile Authentication Methods

When mobile devices connect to business networks, user and endpoint authentication play critical roles in preventing misuse, abuse and attack. In this tip, we identify mobile authentication methods and discuss how to leverage authenticated identities to control what mobile laptops, PDAs and smartphones can and cannot do inside your network.

Authentication verifies that users or systems are who they claim to be, based on identity (e.g., username) and credentials (e.g., password). Most highly publicized breaches are attributed to weak or absent authentication -- from unlocked laptops to wireless networks with cracked passwords. Many expensive and embarrassing incidents could be avoided by requiring robust authentication to mobile devices and the networks they use.

Mobile devices are easily lost or stolen, requiring protection against unauthorized access to their data, applications and connectivity. However, mobile users require frequent access for brief periods, making repeated password entry inconvenient. While most mobile laptops are set to require logins, the majority of PDAs and smartphones are not. Mobile passwords are widely available but rarely used unless employers enforce them.

1.2 How to authenticate Identity

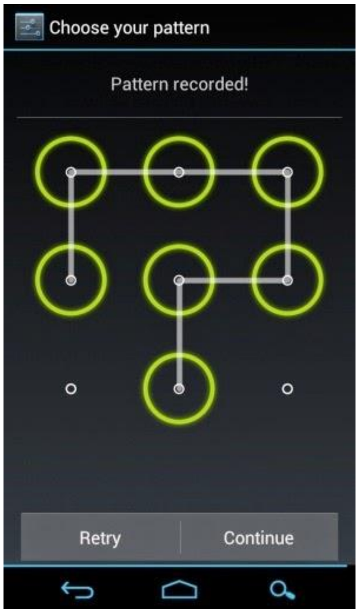
When planning your mobile authentication strategy, strive to combine strength and enforceability with usability. Consider both device and network access credentials and how well each method can satisfy your platform, security and user requirements.

1.2.1 Passwords

If office desktops log into your Windows domain, you'll be tempted to reuse those passwords for users who connect from laptops and PDAs. Because simple passwords are easily guessed, you might enforce length, complexity and timeout rules. But this can make a handheld device very hard to use. If you choose passwords, combine them with policies that cater to mobile needs -- for example, let users receive calls and appointment notifications without password entry and provide a mobile password recovery process.

1.2.2 Non-text Passwords

Entering text on a mobile can be awkward. Alternatives can require the user to tap symbols within a randomly generated matrix or a sequence of points on a photo. Unlocking a device this way could also decrypt other credentials stored on that handheld so the authenticated user can access his company's network. Symbols are handy on PDAs and tablets without keyboards but are not suitable for devices without a mouse or touch-screen.



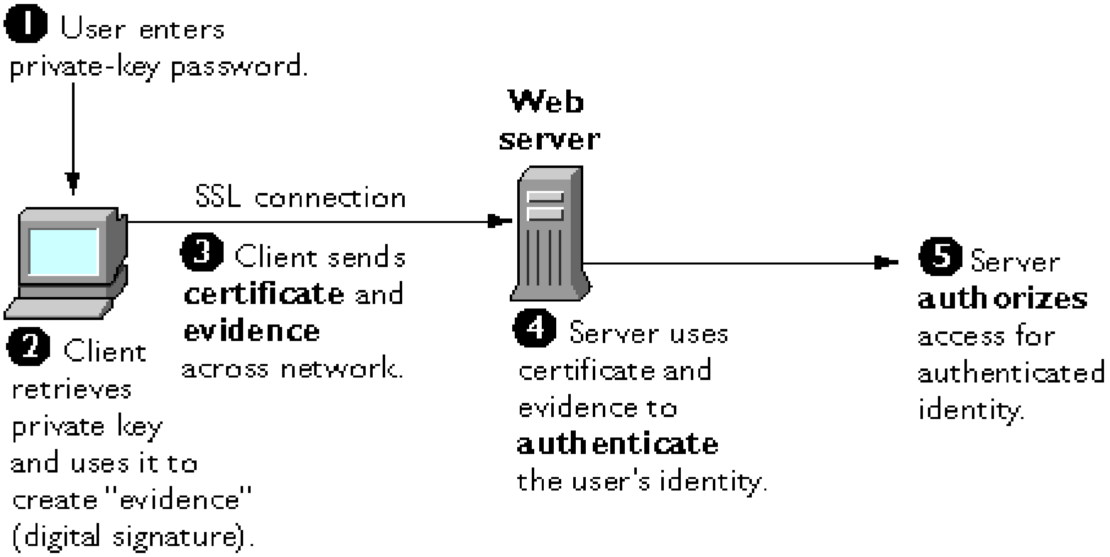
Choose your pattern

Pattern recorded!

Fig-1 Pattern Matching

1.2.3 Certificates

Digital certificates bind an identity to a public/private key pair and are considerably stronger than passwords, so long as the owner's private key is protected. Combining a device lock with certificate-based network authentication is increasingly common -- for example, a Wi-Fi laptop that is unlocked with a password and then uses a certificate on that device for WPA-Enterprise authentication. This method requires a public key infrastructure (PKI) to request, issue, distribute and revoke certificates, but that investment will provide a very strong foundation for access control.



1 User enters private-key password.
Web server
SSL connection
3 Client sends **certificate** and **evidence** across network.
5 Server **authorizes** access for authenticated identity.
2 Client retrieves private key and uses it to create "evidence" (digital signature).
4 Server uses certificate and evidence to **authenticate** the user's identity.

Fig-2 Secure login using Certificate

1.2.4 Smart Card

Certificates can also be used to unlock a device, but doing so requires a way to store and "enter" the owner's private key. This is essentially what a smart card does. A smart card is a security chip, embedded in a credit card, badge or MMC/SD memory. That chip provides safe storage for cryptographic keys used by authentication and encryption algorithms. For example, a laptop may be unlocked by inserting an employee's badge into the laptop's card reader. When that employee launches a VPN tunnel or Wi-Fi connection, a certificate on the smart card can be automatically used for network authentication. Handset identity modules: Smart card-like methods have long been used for cellular network authentication. GSM handsets and data cards contain subscriber identity module (SIM) cards. 3G mobile devices authenticate themselves with universal services identity modules (USIMs), CSIMs (CDMA subscriber identity modules) or removable user identity modules (RUIMs). These identity modules can be leveraged during enterprise network authentication, either alone or in conjunction with user authentication.

1.2.5 Hardware tokens

Many companies authenticate laptop users with small physical devices (hardware tokens) that generate one-time passwords. Each password is part of a series generated from a cryptographic seed known to the network and the user and is valid for only about a minute. The user typically enters his text password, followed by the string displayed by his token. This approach avoids crackers and key loggers, since passwords are not reused. Furthermore, hardware tokens (and other physical methods like smart cards) prevent password sharing. However, they also incur per-user cost for hardware purchase, distribution and replacement.

1.2.6 Biometrics

Like tokens and smart cards, biometrics are typically used for multi-factor authentication. Multi-factor authentication combines at least two of the following: something you know (e.g., password), something you have (e.g., token) and something you are (e.g., fingerprint). Biometric authentication covers everything in that last category: fingerprints, voiceprints, iris scans, handwritten signatures, and so on. Enterprises have resisted biometrics because of cost, but some new business laptops and PDAs include fingerprint readers, and security programs can easily leverage standard handheld features to accept voice input. Biometrics are very convenient on frequently used mobile devices, but environment (e.g., dirt, noise) must also be considered.

Fig -3 Biometric

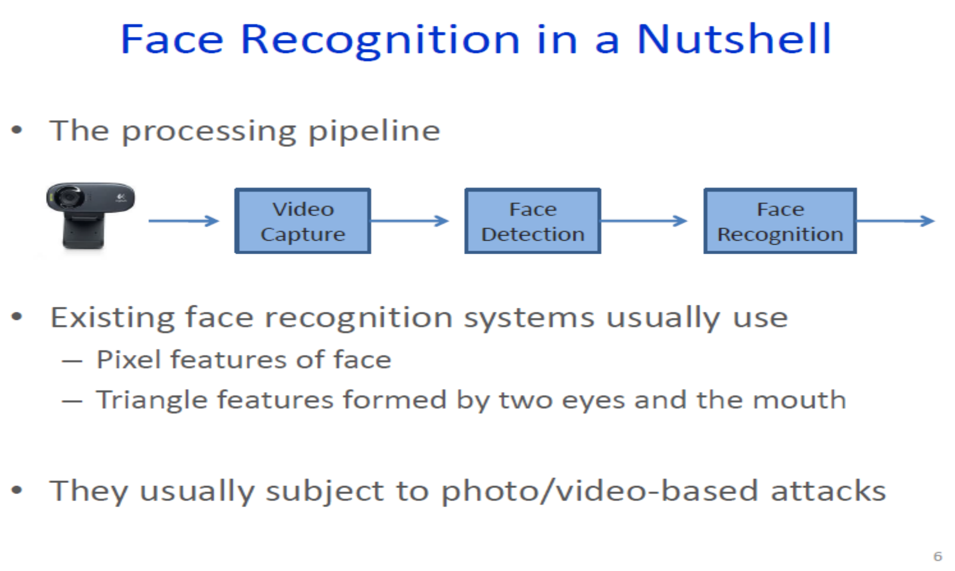
1.2.7 Proximity

A few mobile security products have started to support proximity-based authentication. For example, a PDA or smartphone may stay unlocked indefinitely while communicating with the user's Bluetooth headset. RFID tag readers are being used for proximity-based authentication, permitting connections with mobile devices that pass through a checkpoint and denying connections outside that area. Proximity authentication is not yet common but has the potential to provide more transparent mobile authentication in the future.

1.3 MOTIVATION

Now a day's security is prime concern due to all important data available in Laptop/Mobile/Tablets. As per study mobile user do not lock or use strong password system due to frequent unlocking. Hackers take the benefits of this habits and enjoy the access of unwanted data. Using additional hardware like smartcard/biometrics/proximity makes device more secure and user friendly but we need to pay extra cost for this and sometimes it is not feasible to add these hardware due low cost devices. We thought to use existing feature of mobile device like front facing camera, face recognition, gaze tracking etc. and provide a secure way to authenticate. We studied a very safe system designed by **University of California, Berkeley** "**SAFE: Secure Authentication with Face and Eyes**" As per them they combined face recognition with EYE tracking and provided a very secure way of authentication and it's very tough to hack.

Face authentication is commonly offered as an alternative to passwords for device unlock. However, available face authentication systems are vulnerable to simple spoofing attacks. To defend against these vulnerabilities, we propose a face authentication system that includes a secrecy challenge. During authentication, the user must not only show her face but also gaze at a secret icon that moves across the screen.



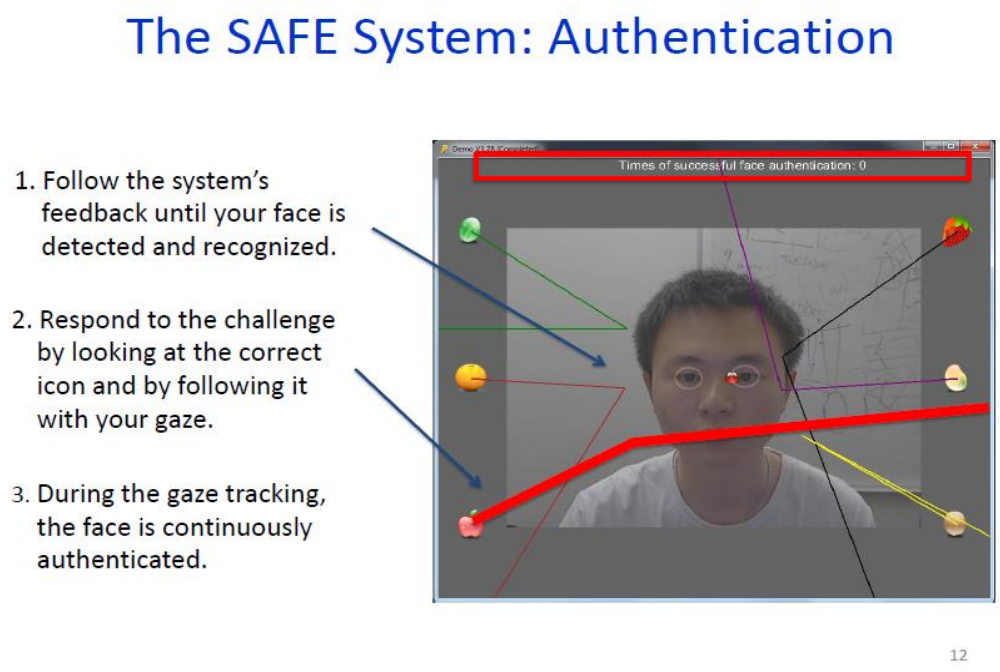
Face Recognition in a Nutshell

- The processing pipeline

- Existing face recognition systems usually use
  - Pixel features of face
  - Triangle features formed by two eyes and the mouth
- They usually subject to photo/video-based attacks

6

Combining face recognition with EYE tracking makes it secure.



The SAFE System: Authentication

1. Follow the system's feedback until your face is detected and recognized.
2. Respond to the challenge by looking at the correct icon and by following it with your gaze.
3. During the gaze tracking, the face is continuously authenticated.

Times of successful face authentication: 0

12

1.4 PROBLEM STATEMENT

Existing system proposed by University **of California, Berkeley** requires additional hardware to track eye movements. We cannot port this solution in mobile devices as user will not carry any additional hardware to authenticate. It will also increase the devices cost.

To resolve such problems, a new method is required to calculate eye movement using front facing camera. We can divide mobile/tablet screens in parts and judge the user current eye viewing location on the basis of eye movements. Front facing camera will check the pixel location coordinate and provide next pixel coordinate as soon as eye pupil moves across mobile/tablet screen.

1.5 SCOPE OF THE THESIS

The scope of this thesis is to study the secure method of authentication using eye tracking without using any additional hardware. We will integrate face detection algorithm with eye tracking using secrecy challenge.

1.6 THESIS ORGANIZATION

Chapter 1 begins with General introduction and related work. It addresses the topics like, Problem Statement, Scope, Related work and thesis organization, different secure authentication methodologies and comparison.

Chapter 2 presents the proposed research methodology which explains the detailed model of inference system used.

Chapter 3 shows the implementation of the proposed methodology also the tools used in it.

Chapter 4 concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

"**SAFE: Secure Authentication with Face and Eyes**" A **University of California, Berkeley** solution, As per them they combined face recognition with EYE tracking and provided a very secure way of authentication and it's very tough to hack.

Face authentication is commonly offered as an alternative to passwords for device unlock. However, available face authentication systems are vulnerable to simple spoofing attacks. To defend against these vulnerabilities, they propose a face authentication system that includes a secrecy challenge. During authentication, the user must not only show her face but also gaze at a secret icon that moves across the screen.

We thought to use existing feature of mobile device like front facing camera, face recognition, gaze tracking etc. and provide a secure way to authenticate.

2.1 Existing SAFE Solution

Typing in a Password/PIN…

…takes away the instant gratification of checking your device

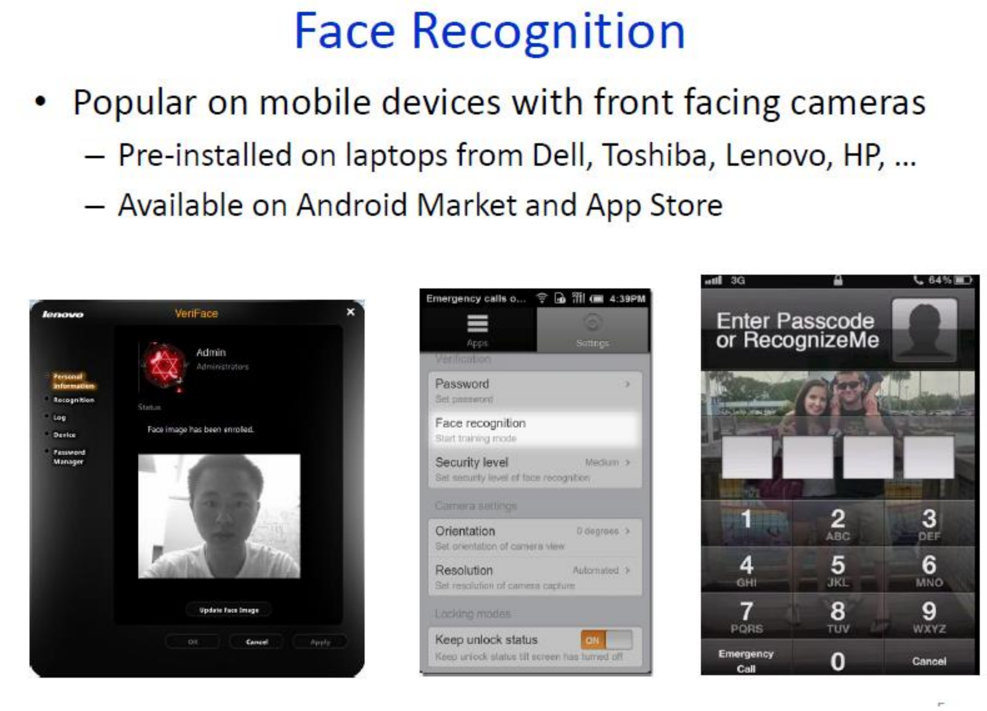
…requires concentration and two hands

…estimated 38-54% of users who do not lock their mobile phones

Project Objectives

- Is there a better way to authenticate with your device?
- Can we leverage the existing features of mobile devices and emerging technologies?
  - E.g., front-facing cams, face-recognition, gaze- tracking, …
  - Are these technologies mature enough?
- Is hands-free authentication procedure feasible?

4



Face Recognition

- Popular on mobile devices with front facing cameras
  - Pre-installed on laptops from Dell, Toshiba, Lenovo, HP, …
  - Available on Android Market and App Store

Face image has been enrolled.

Password — Set password

Face recognition — Start training mode

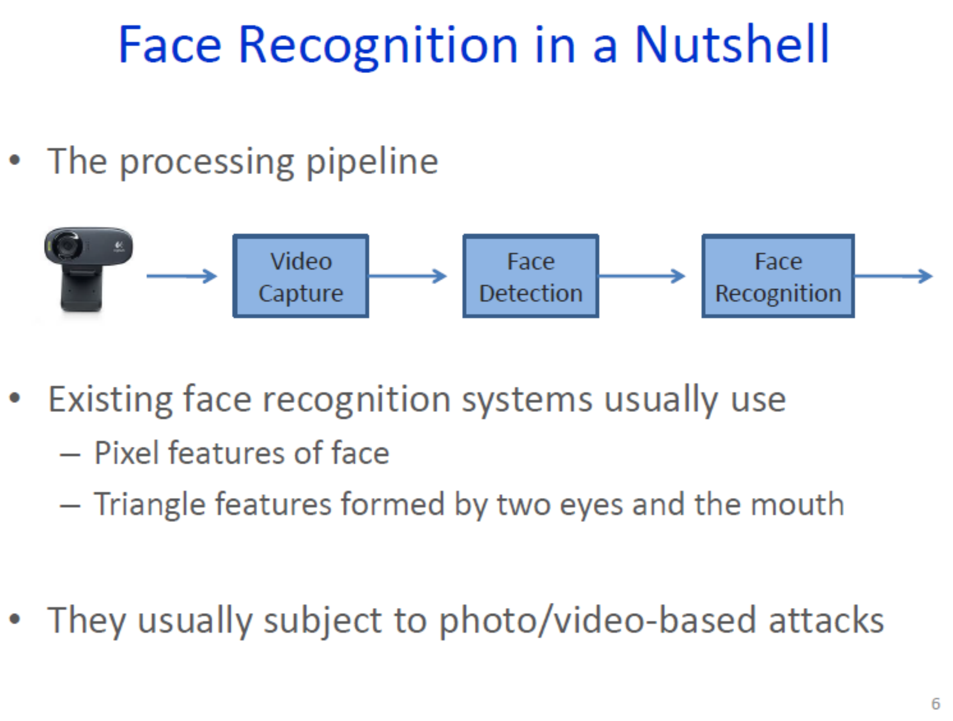
Security level: Medium — Set security level of face recognition

Orientation: 0 degrees — Set orientation of camera view

Resolution: Automated — Set resolution of camera capture

Keep unlock status: ON — Keep unlock status till screen has turned off

Enter Passcode or RecognizeMe



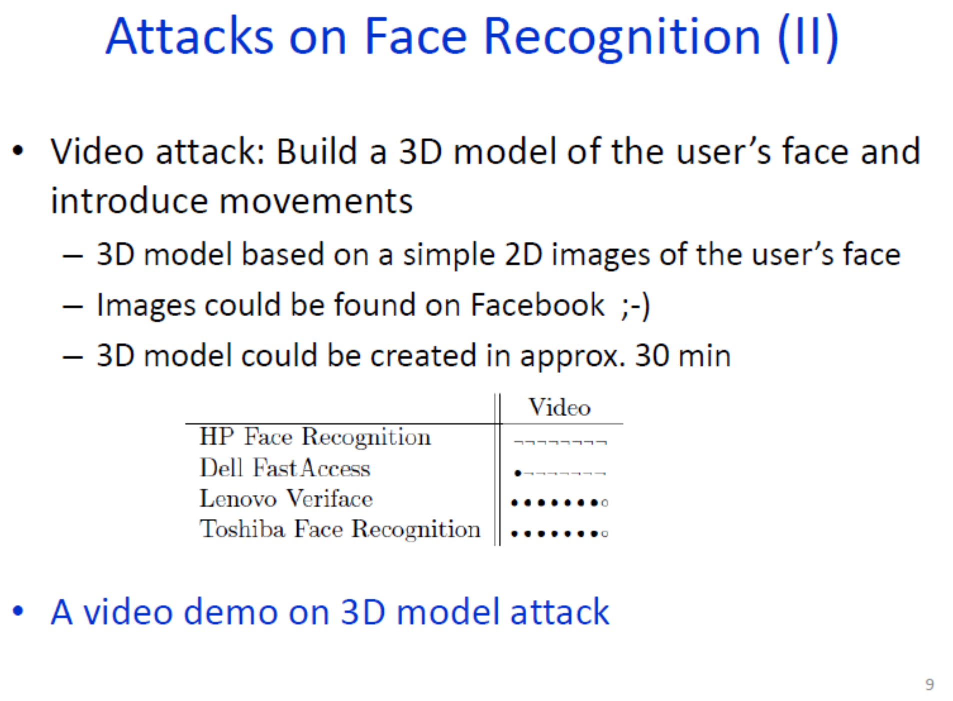
Face Recognition in a Nutshell

- The processing pipeline

- Existing face recognition systems usually use
  - Pixel features of face
  - Triangle features formed by two eyes and the mouth

- They usually subject to photo/video-based attacks

6



Attacks on Face Recognition (II)

- Video attack: Build a 3D model of the user’s face and introduce movements
  - 3D model based on a simple 2D images of the user’s face
  - Images could be found on Facebook ;-)
  - 3D model could be created in approx. 30 min

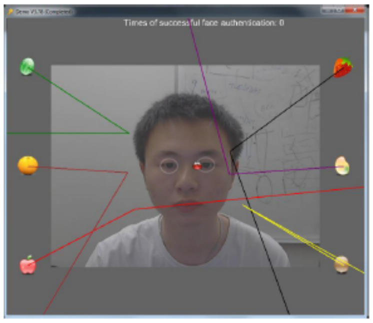
	Video
HP Face Recognition	¬¬¬¬¬¬¬¬¬
Dell FastAccess	•¬¬¬¬¬¬¬¬
Lenovo Veriface	•••••••○
Toshiba Face Recognition	•••••••○

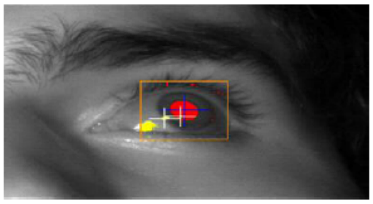
- A video demo on 3D model attack

9

Our SAFE System

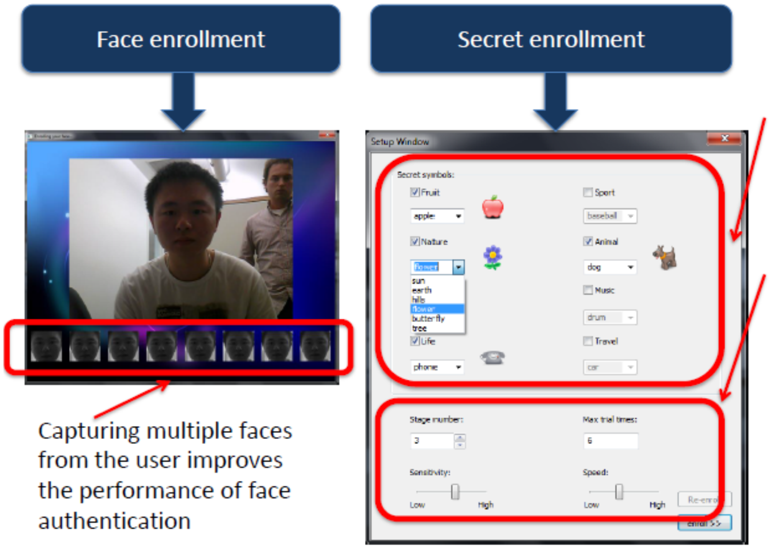
- Face recognition for identity
  - Based on facial features from images
  - Pixel values, relative position between eyes and mouth, etc.
- Video-based gaze-tracking
  - Tracking corneal reflections & pupils
  - Using web cam with a controlled source of IR light
- Challenge-response protocol
  - Secret verification (using gaze-tracking)
  - Limited number of the response attempts





10

The SAFE System: Enrollment



Secret symbols:

☑ Fruit — apple
☐ Sport — baseball
☑ Nature — flower (sun, earth, hills, flower, butterfly, tree)
☑ Animal — dog
☐ Music — drum
☑ Life — phone
☐ Travel — car

Stage number: 3
Max trial times: 6
Sensitivity: Low High
Speed: Low High

Capturing multiple faces from the user improves the performance of face authentication

User parameters:

- Choosing a secret icon

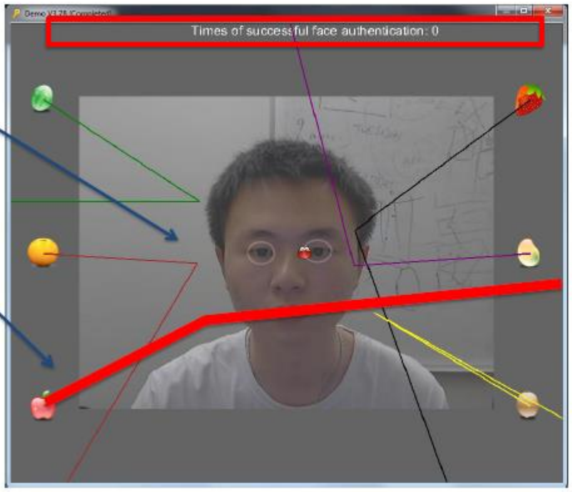
System parameters:

- No. of authentication rounds
- Speed of the challenge
- Number of attempts allowed
- Lock-out time for next attempt

11

The SAFE System: Authentication

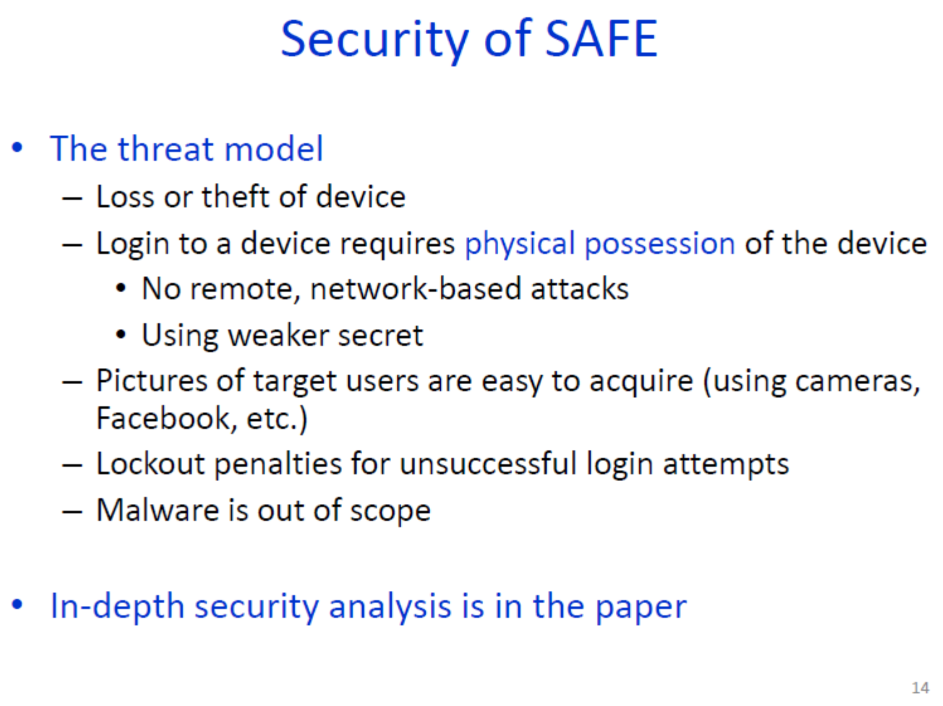
1. Follow the system's feedback until your face is detected and recognized.
2. Respond to the challenge by looking at the correct icon and by following it with your gaze.
3. During the gaze tracking, the face is continuously authenticated.



Times of successful face authentication: 0

The SAFE System Implementation

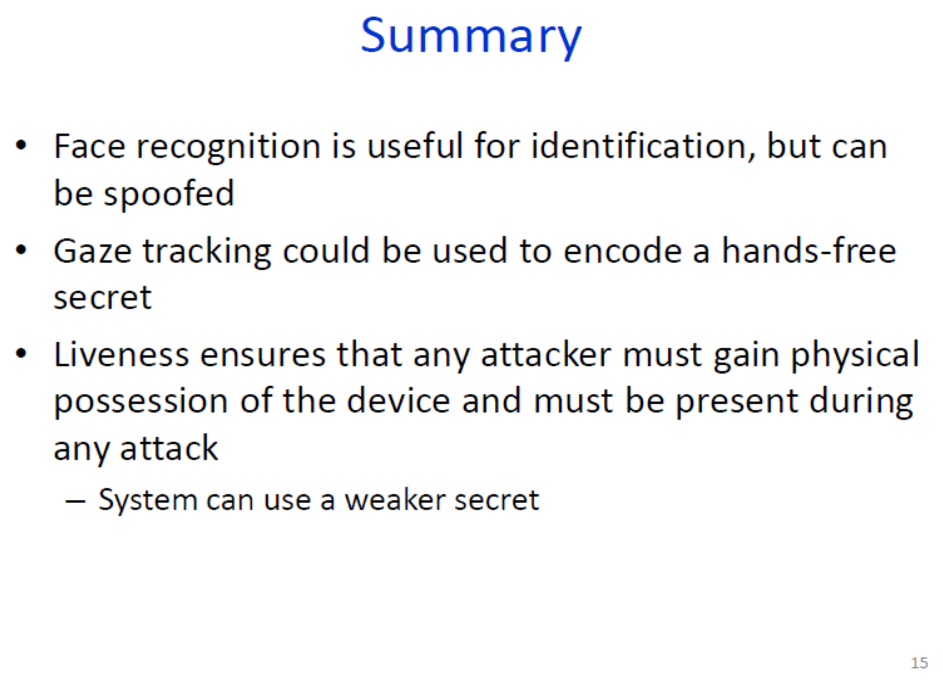
- Technology-related problems
  - Inaccuracy of gaze
  - Face-recognition performance
  - Integration of multiple components on mobile devices
- System implementation
  - Samsung Tablet with Windows 7
  - ITU Gaze Tracker
  - Face recognition based on OpenCV
  - Multi-core implementation
- The SAFE video demo



Security of SAFE

- The threat model
  - Loss or theft of device
  - Login to a device requires physical possession of the device
    - No remote, network-based attacks
    - Using weaker secret
  - Pictures of target users are easy to acquire (using cameras, Facebook, etc.)
  - Lockout penalties for unsuccessful login attempts
  - Malware is out of scope
- In-depth security analysis is in the paper

14



Summary

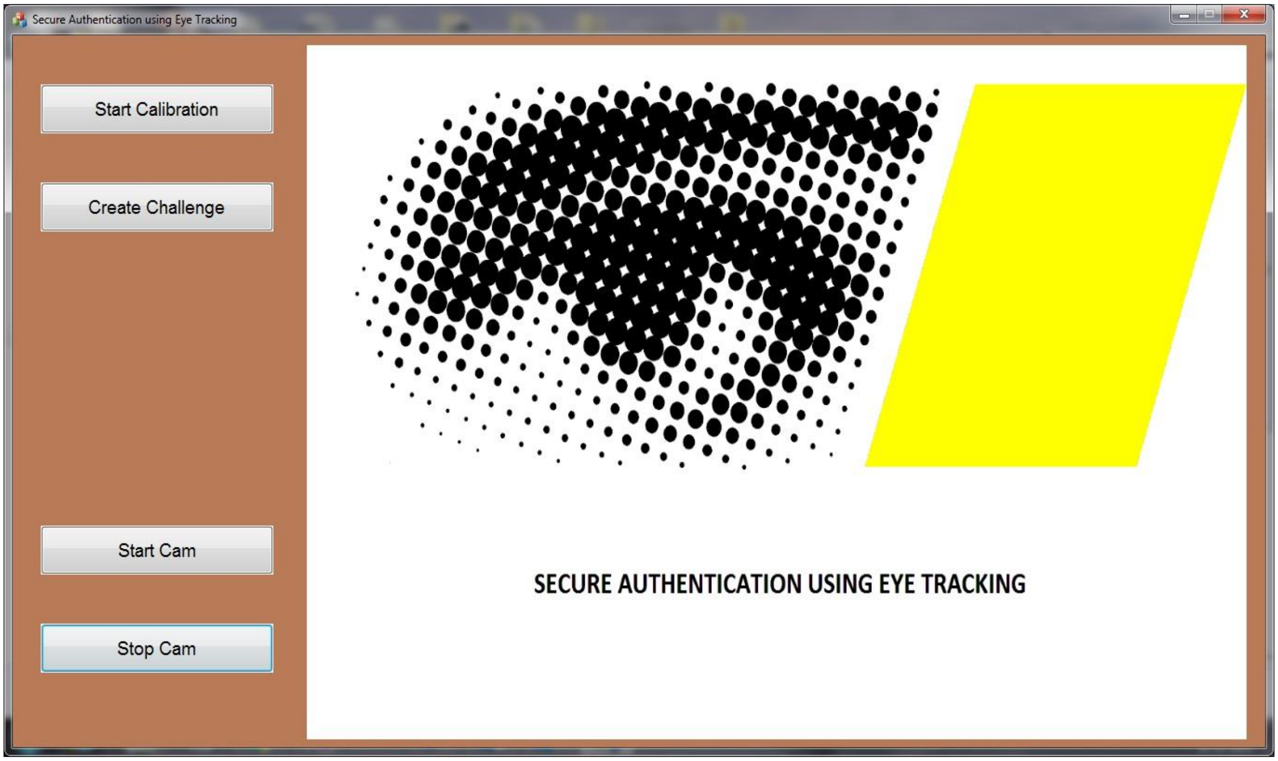
- Face recognition is useful for identification, but can be spoofed
- Gaze tracking could be used to encode a hands-free secret
- Liveness ensures that any attacker must gain physical possession of the device and must be present during any attack
  - System can use a weaker secret

15

2.2 Proposed Solution SAET

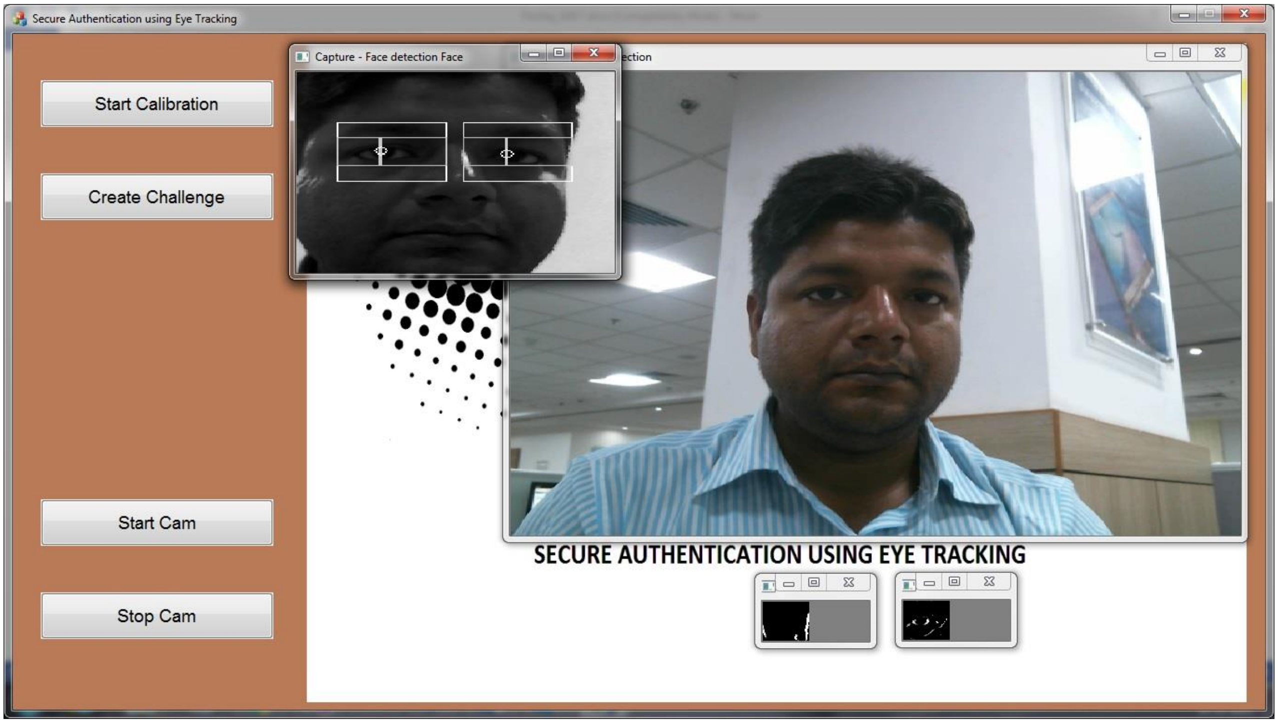
Existing system proposed by University of California, Berkeley requires additional hardware to track eye movements. We cannot port this solution in mobile devices as user will not carry any additional hardware to authenticate. It will also increase the devices cost.

To resolve such problems, a new method is required to calculate eye movement using front facing camera. We can divide mobile/tablet screens in parts and judge the user current eye viewing location on the basis of eye movements. Front facing camera will check the pixel location coordinate and provide next pixel coordinate as soon as eye pupil moves across mobile/tablet screen.


Secure Authentication using Eye Tracking
Start Calibration
Create Challenge
Start Cam
Stop Cam
SECURE AUTHENTICATION USING EYE TRACKING

2.2.1 Start Camera

Start Camera will use device front camera and display the user image. Software will also draw the small circle around the eye pupil and that circle will always follow the eye pupil movements. Software will provide this circle coordinate that will be nothing but the coordinate of eye movement. We will apply calculation on this coordinate and judge the current eye viewing location.

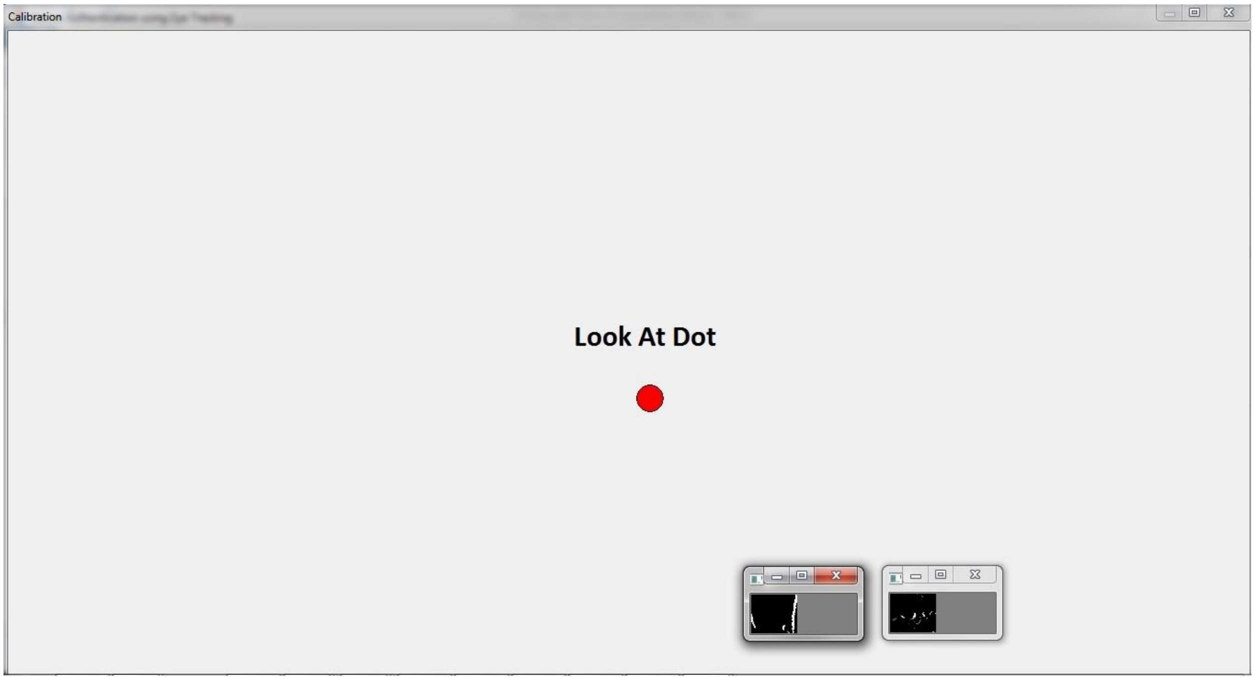

Secure Authentication using Eye Tracking
Capture - Face detection Face
ection
Start Calibration
Create Challenge
Start Cam
Stop Cam
SECURE AUTHENTICATION USING EYE TRACKING

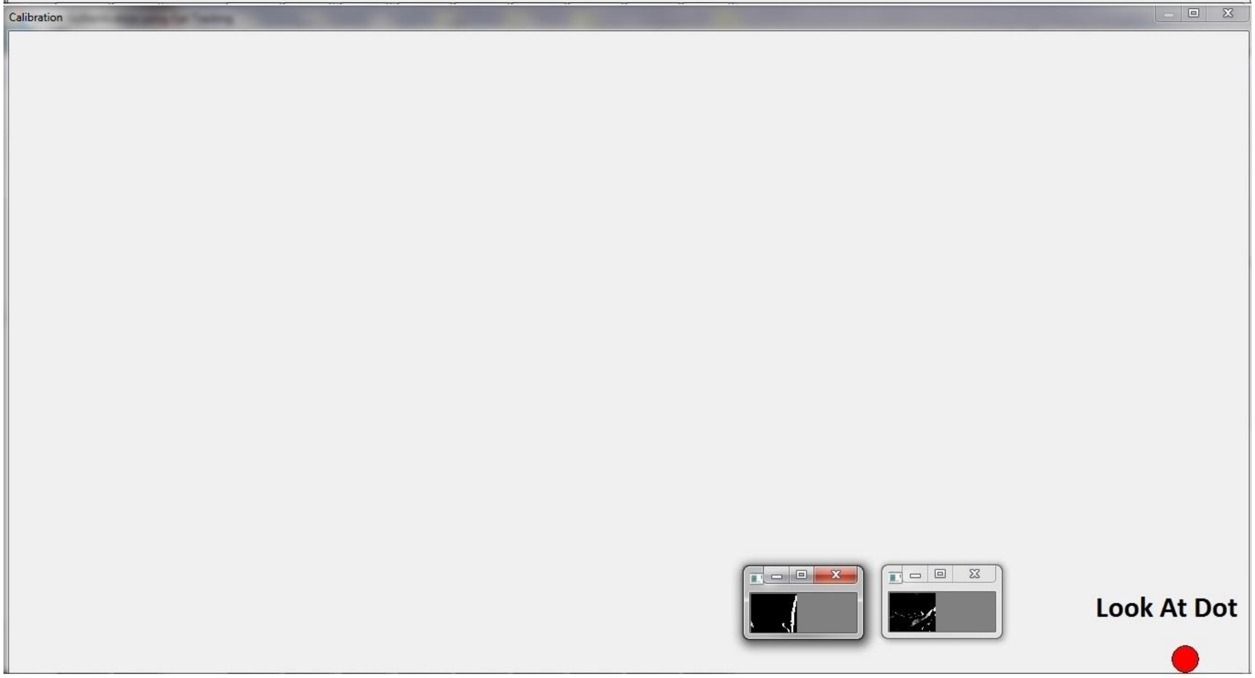
2.2.2 Stop camera

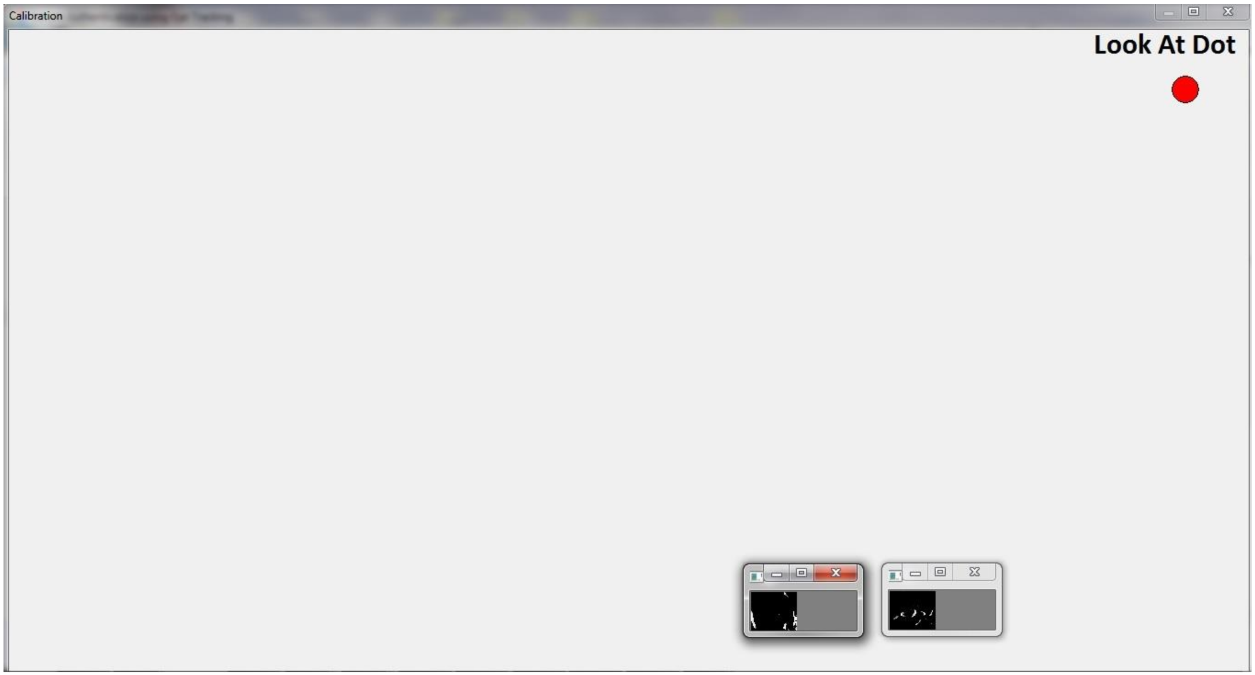
Stop Camera will off front camera. We provided this Start Camera/Stop Camera feature to check user whether our software support for their device camera hardware or not. We are using OPEN CV libraries to open camera and display live feed provide by front camera.

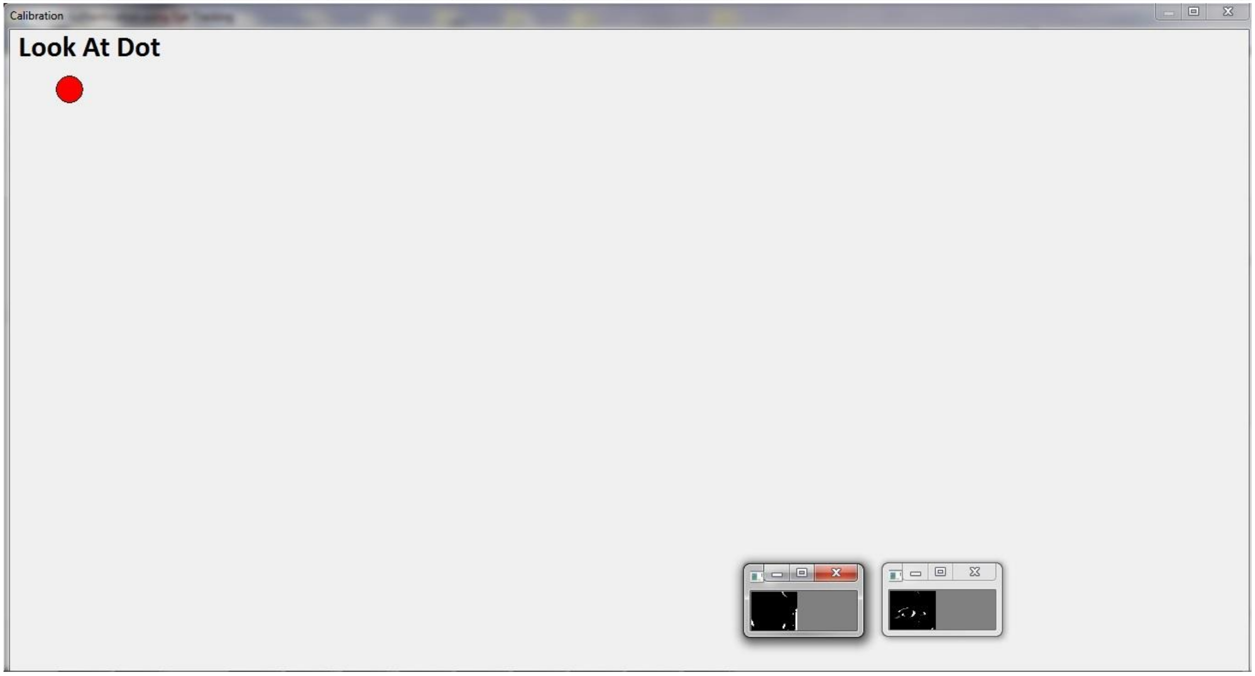
2.2.3 Start Calibration

Eye tracking data may be used for various purpose. In most cases it is used to estimate a gaze point – that is the place where a person is looking at. Most of the device provides the eye tracking point but without gaze point we cannot assure about the current viewing location.







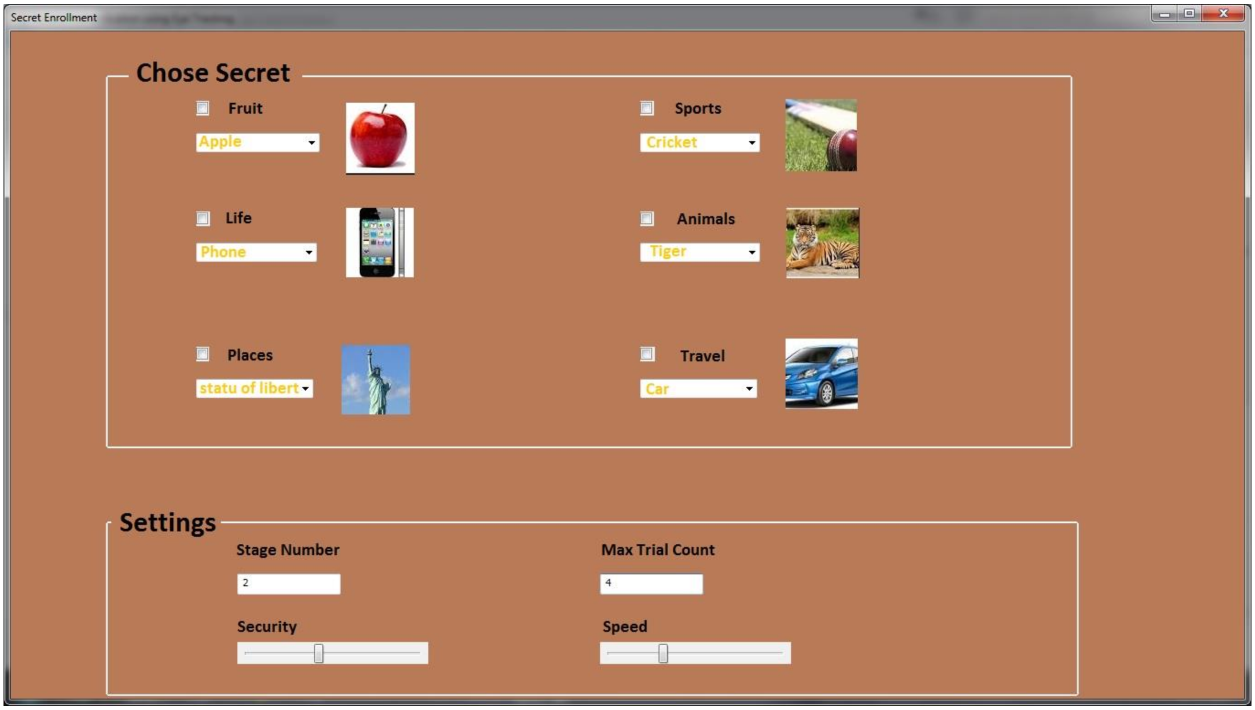


To get this gaze point we should study about the behavior of user eye pupil while looking various screen coordinate vertically and horizontally. Every human being eye movements are different when they view any screen and that's why it is important to know about current user eye pupil behavior in order to judge accurate current viewing location.

2.2.4 Create Challenge

Secure authentication will require secret password and we are providing a Create challenge screen to user to select their secret word and related images. These images will be displayed at any random position of screen and user needs to focus on that area in order to get secure access. We have chosen total 6 positions in screen like left middle, top left, top right, right middle, bottom left and bottom right. Authentication screen will display images in all these location and only one image will your pre authorized image. User needs to draw attention on that image for 15~30 second.

User can also select number of stages like more than one time authentication screen will come and their different preset images. We also providing max number of attempt option and after reaching that limit your device will locked for certain time and that will decided on the basis of security. User can also select speed to limit time for one authentication screen (very from 15~30 seconds).


Secret Enrollment

Chose Secret

□ Fruit: Apple

□ Sports: Cricket

□ Life: Phone

□ Animals: Tiger

□ Places: statu of libert

□ Travel: Car

Settings

Stage Number: 2

Max Trial Count: 4

Security

Speed

CHAPTER 3

Methodology

3.1 Open CV

OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

3.2 Eye Detection and Tracking

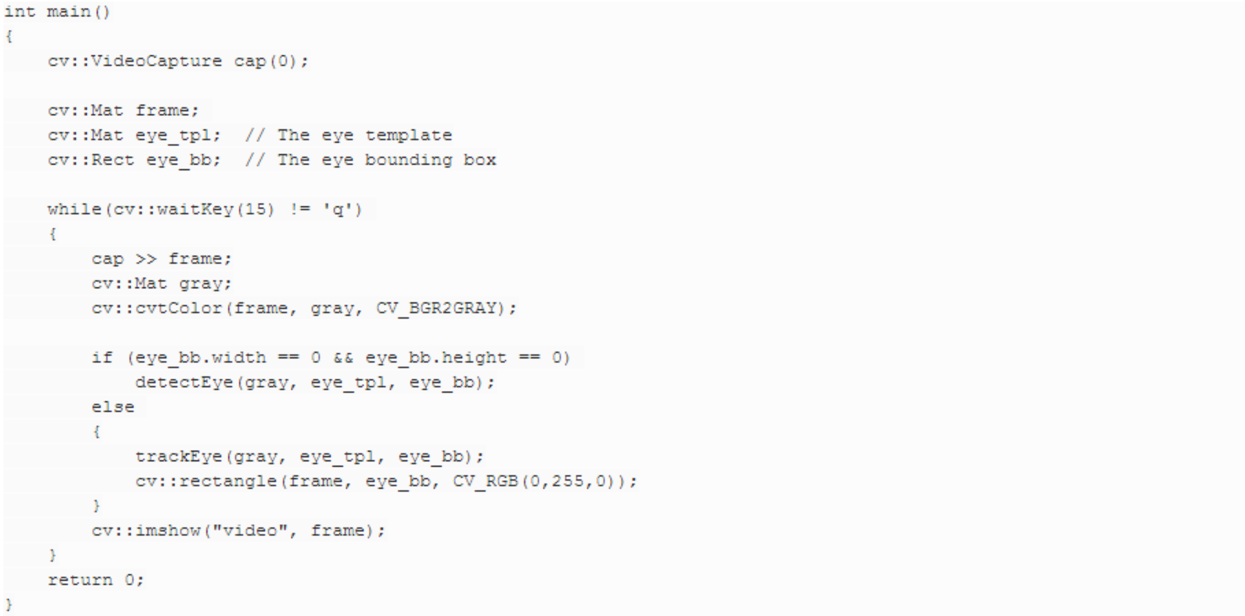
This is fast eye detection and tracking program that takes the input from webcam. The program using OpenCV’s face detector for detecting the user’s face and eye. For tracking the user’s eye, it is using the template matching method.

If you look into the OpenCV samples directory, you'll find facedetect.cpp which will detect user's face from webcam using Viola-Jones method. The program performs very well in detecting human faces, but it runs rather slow because of the complex algorithm.

I want to take facedetect.cpp one steps further for detecting and tracking user's eye in real-time. In order to achieve high speed, the program need to perform the face and eye detection only *once* at the program startup. After the eye is successfully detected, an eye template is created at runtime and will be used for tracking the eye using template matching method. This will greatly increase the speed of the real-time tracking.

The skeleton of the program

Let's take a look at the skeleton of the program.



```
int main()
{
    cv::VideoCapture cap(0);

    cv::Mat frame;
    cv::Mat eye_tpl;  // The eye template
    cv::Rect eye_bb;  // The eye bounding box

    while(cv::waitKey(15) != 'q')
    {
        cap >> frame;
        cv::Mat gray;
        cv::cvtColor(frame, gray, CV_BGR2GRAY);

        if (eye_bb.width == 0 && eye_bb.height == 0)
            detectEye(gray, eye_tpl, eye_bb);
        else
        {
            trackEye(gray, eye_tpl, eye_bb);
            cv::rectangle(frame, eye_bb, CV_RGB(0,255,0));
        }
        cv::imshow("video", frame);
    }
    return 0;
}
```

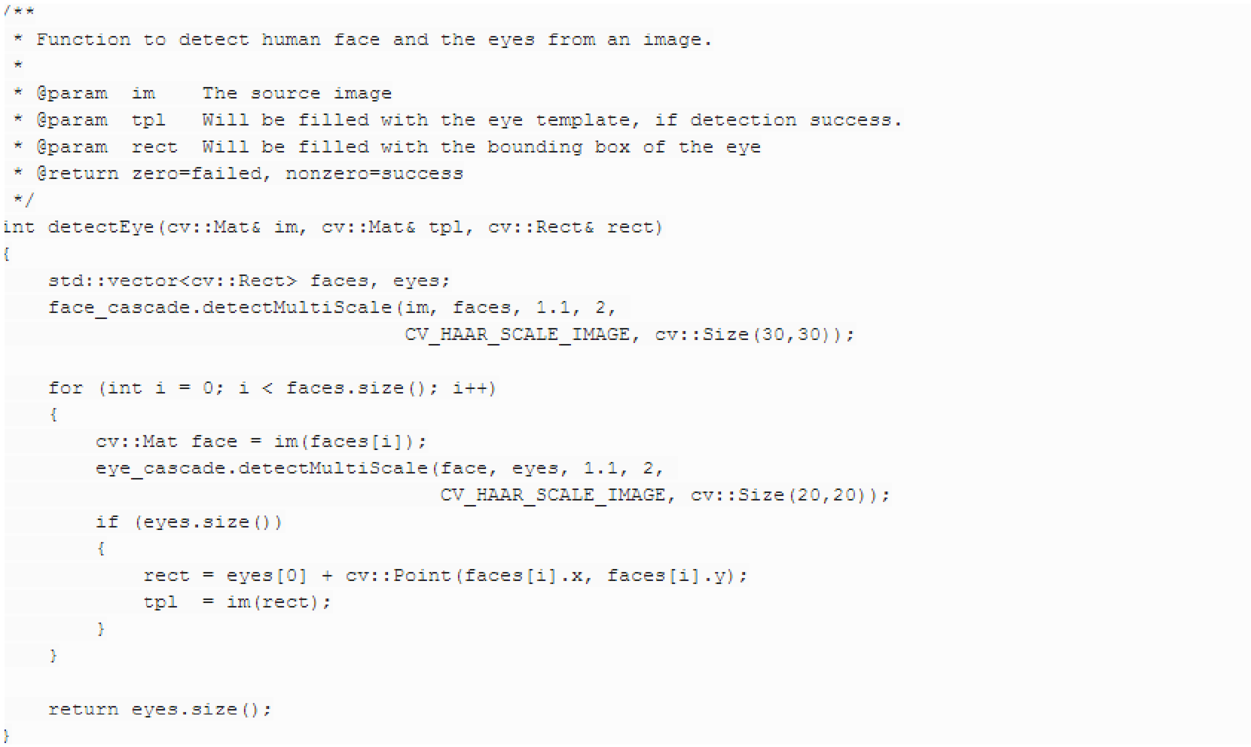
In the main() function, the program open video stream from webcam. Each frame is converted to grayscale to reduce processing time. If the bounding box of the eye is still empty, the program calls thedetectEye() function. It is the function for detecting user's face and eye. If the eye is successfully located, the function will return an eye template and its bounding box.

Given the bounding box is set, the program calls the trackEye() function. This function takes the current frame, eye template, and eye bounding box as the input. The eye template will be used for locating the eye in the given frame with template matching method. If the eye successfully located, the bounding box will be updated for the new location of the eye.

If somehow the eye tracking is lost, the eye bounding box will be cleared so the program will call the detectEye() function again.

Detecting user's face and eye

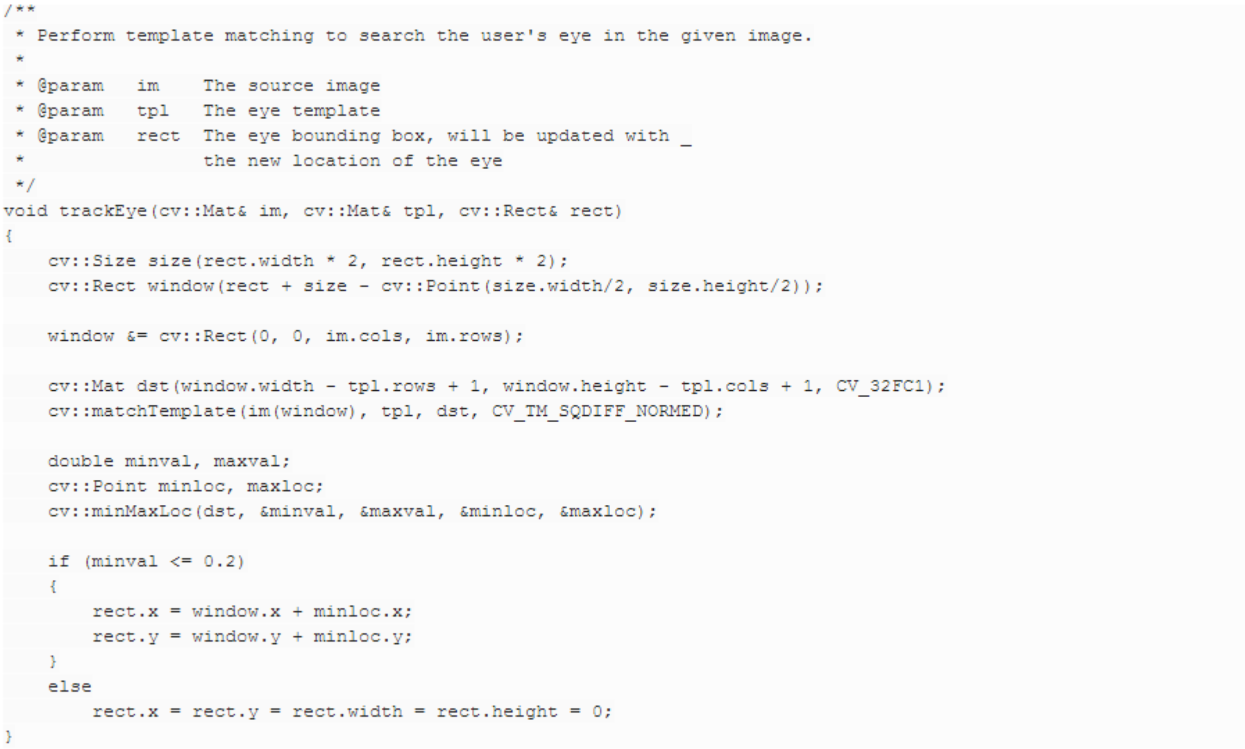
The face and eye detection is implemented in the detectEye() function. Given an image, the function tries to detect human face in it. If success, it will continue with detecting the eye. If success, it will create and returns the eye template and its bounding box.



```
/**
 * Function to detect human face and the eyes from an image.
 *
 * @param  im    The source image
 * @param  tpl   Will be filled with the eye template, if detection success.
 * @param  rect  Will be filled with the bounding box of the eye
 * @return zero=failed, nonzero=success
 */
int detectEye(cv::Mat& im, cv::Mat& tpl, cv::Rect& rect)
{
    std::vector<cv::Rect> faces, eyes;
    face_cascade.detectMultiScale(im, faces, 1.1, 2,
                                  CV_HAAR_SCALE_IMAGE, cv::Size(30,30));

    for (int i = 0; i < faces.size(); i++)
    {
        cv::Mat face = im(faces[i]);
        eye_cascade.detectMultiScale(face, eyes, 1.1, 2,
                                     CV_HAAR_SCALE_IMAGE, cv::Size(20,20));
        if (eyes.size())
        {
            rect = eyes[0] + cv::Point(faces[i].x, faces[i].y);
            tpl  = im(rect);
        }
    }

    return eyes.size();
}
```

Tracking user's eye with template matching

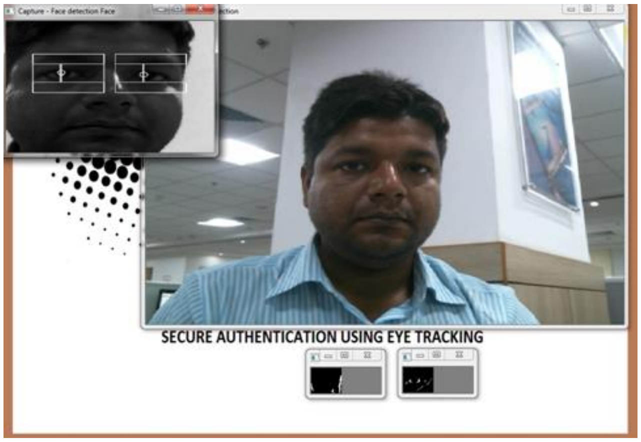
Given the eye template and its bounding box is set, this function will locate the eye in the given frame with template matching method. The template matching is performed in a search window to increase the speed. If success, the bounding box will be updated to the new location of the eye.



```
/**
 * Perform template matching to search the user's eye in the given image.
 *
 * @param   im    The source image
 * @param   tpl   The eye template
 * @param   rect  The eye bounding box, will be updated with _
 *                the new location of the eye
 */
void trackEye(cv::Mat& im, cv::Mat& tpl, cv::Rect& rect)
{
    cv::Size size(rect.width * 2, rect.height * 2);
    cv::Rect window(rect + size - cv::Point(size.width/2, size.height/2));

    window &= cv::Rect(0, 0, im.cols, im.rows);

    cv::Mat dst(window.width - tpl.rows + 1, window.height - tpl.cols + 1, CV_32FC1);
    cv::matchTemplate(im(window), tpl, dst, CV_TM_SQDIFF_NORMED);

    double minval, maxval;
    cv::Point minloc, maxloc;
    cv::minMaxLoc(dst, &minval, &maxval, &minloc, &maxloc);

    if (minval <= 0.2)
    {
        rect.x = window.x + minloc.x;
        rect.y = window.y + minloc.y;
    }
    else
        rect.x = rect.y = rect.width = rect.height = 0;
}
```

The result

When the program executed, method detect the face and the eye. But when the eye is successfully detected, the video plays smoothly.


Capture - Face detection Face
SECURE AUTHENTICATION USING EYE TRACKING

3.3 Tools Used to Develop this Program

I used visual studio 2013 for demonstration of this project. This project is using OPENCV library for face and eyes detection. MFC based dialogs used in this project. This project will be redeveloped in Android/iOS and porting will be done for secure authentication.

CHAPTER 4

CONCLUSION & FUTURE WORK

4.1 CONCLUSION

Now a days PDA have all the personal as well as official data stored and very important to keep those data away from illegally access. Two way authentication will provide a very secure and personal way of access and no body other than real user will get that access. User can select their pre authorized Images/Symbols for secure authentication and provide their consent via eye tracking only while unlocking the device.

This authentication system will also help to prevent the situation where passwords can be stolen using spy camera installed on roofs and tracked user keypress events. This system will also provide a user friendly way to authenticate as you do not need to enter you highly complicated and lengthy password again and again while unlocking PDA. Software will use the existing PDA feature like front camera and track eye movement through in build logic so there will be no need to purchase any addition hardware like gaze tracker.

4.2 Future work

Smart PDA have all the important things at one place like your bank E-Passbook, Banking APPs, Social Media APPs (Facebook, twitter) etc. We can associate your daily activates with secure login and software will ask you random question based on your last week activities like whom you transferred 1000 $ last week or which episode you watched on you tube recently. These question can be linked with eye tracking and make this system more secure.

REFERENCES

[1] **Authentication Wiki page** https://en.wikipedia.org/wiki/Authentication

[2] **Two-factor Authentication** https://en.wikipedia.org/wiki/Two-factor_authentication

[3] **Mobile Authentication Methods** http://searchmobilecomputing.techtarget.com/feature/Managing-mobile-authentication-methods

[4] SAFE: Secure Authentication with Face and Eyes

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6927175

[5] Real Time Eye tracking using Smart Camera

http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=6176373

[6] Explore New Eye Tracking and Gaze Locating Methods http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6722242

[7] Analysis of Eye Tracking Techniques in Usability and HCI Perspective

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6828034

[8] Research and implementation of RSA algorithm for encryption and decryption.

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6021216