

Authentication Lock for Application Integration Face Recognition Security

Muhammad Aliff Romi Bin Sharipudin, Firoz bin Yusuf Patel Dawoodi

Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia
Johor, Malaysia

maliffromi@gmail.com, firoz@utm.my

Abstract— An application created for android environment to provide user with better security using biometric security. Authentication Lock for Application Integration is an android mobile's application that implemented Google's Vision API for facial recognition. It features an algorithm to calculate the ratio of a user's face to be stored as data and matching it again during the facial recognition process. Authentication Lock also provides an alternative method to gain access to the lock application using PIN. As conclusion, Authentication Lock for Application Integration is to help users to secure access to their applications from others by using biometric security.

Keywords—Biometric, Face Recognition, Android, Mobile Application, Vision API, PIN Authentication.

1. Introduction

Security is a state where an environment is secured from malicious intent. The implementation of security is widely used from personal computer to user's smartphone. The development of security has changed significantly over the years, from simple passcode to using user's biometric features to give security. The advance technology in present day has made it possible to integrate biometric security with smartphones.

However, there remains two problems when biometric security is to be implemented on mobile application. First, some biometric security requires external tools and these tools are mostly expensive. Second, most existing applications on the Android Market namely FaceLock[1] does provide user with Face Recognition features to secure access for their applications, Although, the non-paid version only allows user to lock only one applications, and for full function of the application, the user is required to pay for it. As a result, the user may not receive full functioning biometric security they require. Thus, an application called "Authentication Lock for Application Integration" are created to satisfy user's needs for biometric security for no payment.

Authentication Lock for Application Integration is developed to enable face recognition security for user and secure their applications from unauthorized user. Authentication Lock has slight advantage over FaceLock in term of availability and functionality, the application is always available running in the background to provide security and to keep track of the locked application.

2. Background

In this section, the basic knowledge used in developing this application is described. The content contains biometric security, face recognition, Vision API and android are all explained briefly.

2.1. Biometric Security[2]

Biometric represents what the user are, what they have and only they have the unique traits. This can be their eyes, voice, fingerprint, face and etc. Biometric security is a method where user's feature become security itself. This kind of security are hard to decrypt as not everyone has the same features as one another.

2.2. Face Recognition[3]

Face Recognition is a process of recognizing a facial feature based on picture. This process is mostly use to authenticate which user is which. It uses an algorithm to recognize a facial feature, and most algorithm used in face recognition are Histogram of Oriented Gradients (HOG), face landmark estimation and Deep Convolutional Neural Network.

2.3. Vision API[4]

Google's Vision API is an open source code for android environment. The API features face track, a tracking features that able to track user's face when they are facing the camera. This feature enables the application to capture the user's face for Enrollment in the application process. From Enrollment, the application uses the picture for Identification with the application's database to ensure the identity of the user. If the user's feature match with the existing database, they are to go through Authentication process to allow user to access secured applications.

2.4. Android

Android is an operating system for smartphone environment. Most of the key mobile applications available on the market are made for android environment as it uses an open source Java Language. It is build from the ground-up to enable developers o create mobile applications.

3. Method

3.1. System Architecture Overview

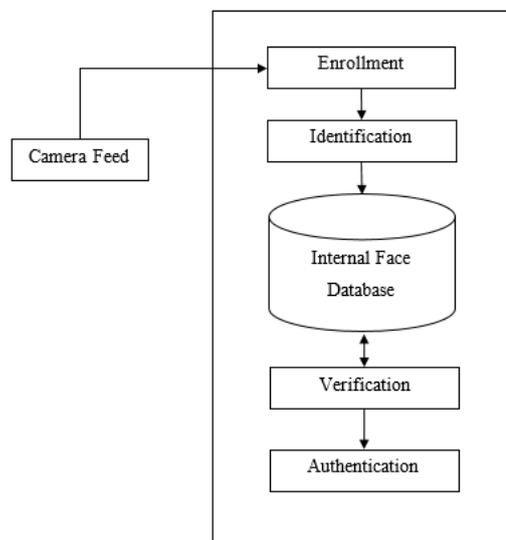


Figure 1. System Architecture Overview

The overview of system architecture of Authentication Lock for Application Integration are shown in Figure 1. The Vision API capture the image from camera feed and process the image through enrollment activity. The enrollment image is processed through identification activity which connects with the application's database stored internally. The image will then be verified if the image matches with the database. If the image matches, the authentication activity will allow access.

3.2. System Structure

This application uses two process for authentication. The first process is the Face Tracking. The Face Tracking process involves user and the camera. The process starts when user access the application. The user will be prompt to set their PIN prior to first time boot. Face Tracking process will capture the face value and store the face information within the application to be use for face recognition activity.

The second process involves Face Recognition. The Face Recognition process starts with the camera capturing the user's face. The camera will try to identify the face on the camera preview screen. Following the identification of the user's face, the face is verified with the internal application's database to ensure that the face matches with the original face save inside the application's database. Following the verification process, if the user's face matches with face stored in the database, the application allows access to use the locked application by going through the Authentication process. If the application does not recognize the user's face, Authentication Lock will then redirect the user to PIN lock for alternative authentication. Figure 2 shows the process involved in the system structure.

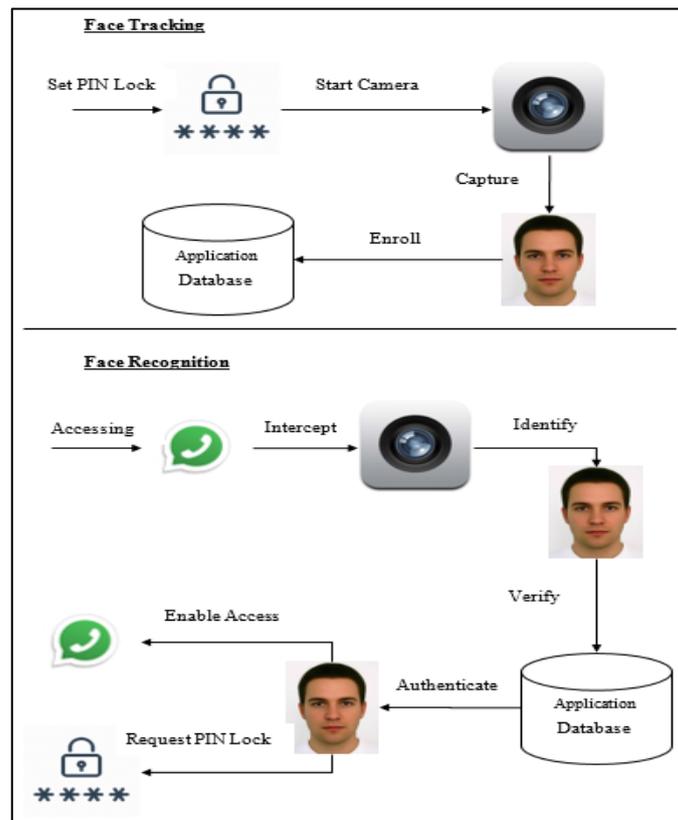


Figure 2. System Structure

4. Results

Authentication Lock for Application Integration is implemented using Java language on Android Studio 2.2.0.0. The application is able to integrate with Google's Vision API to allow the Face Tracking and Face Recognition process.

4.1. User Interface and Layout Design

The application's user interface is as shown in Figure 3.

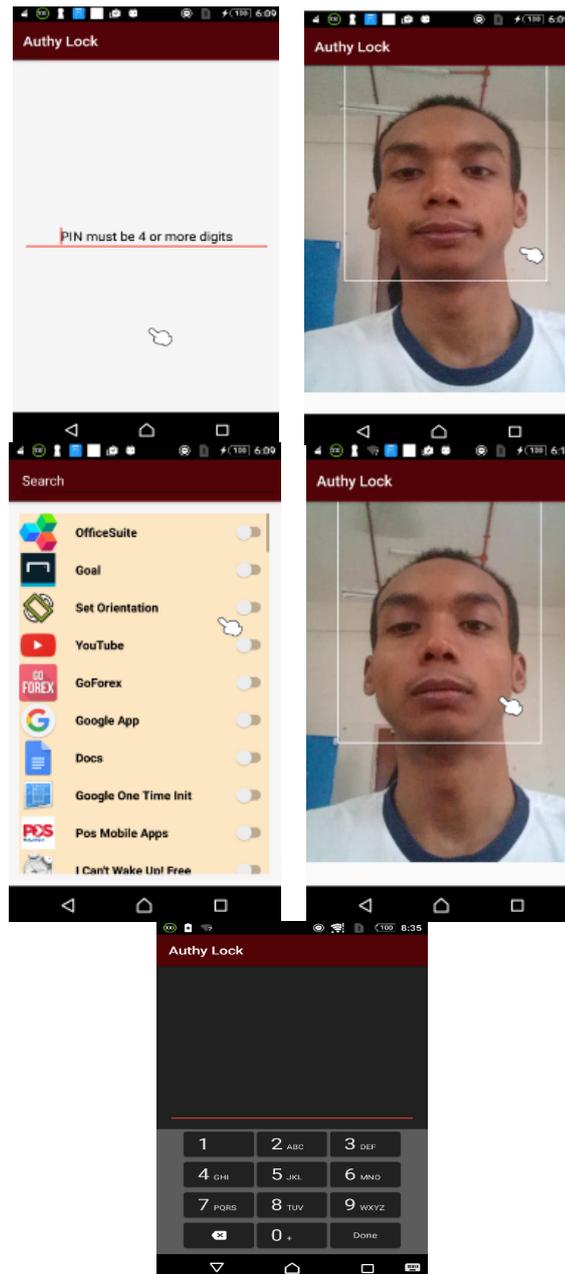


Figure 3. User Interface and Layout

4.2. Code Impementation

The Google's Vision API is the main function for Authentication Lock. The Vision API allows the face detection function using its' library. Aside from that, the application list is used to track down installed application. The code implementation is shown in Figure 4 and Figure 5

```
public void startIfReady() throws IOException {
    if (mStartRequested && mSurfaceAvailable) {
        mCameraSource.start(mSurfaceView.getHolder());
        if (mOverlay != null) {
            Size size = mCameraSource.getPreviewSize();
            int min = Math.min(size.getWidth(), size.getHeight());
            int max = Math.max(size.getWidth(), size.getHeight());
            if (isPortraitMode()) {
                // Swap width and height sizes when in portrait, since it will be rotated by
                // 90 degrees
                mOverlay.setCameraInfo(min, max, mCameraSource.getCameraFacing());
            } else {
                mOverlay.setCameraInfo(max, min, mCameraSource.getCameraFacing());
            }
            mOverlay.clear();
        }
    }
}
```

Figure 4. Vision API implementation

4.3. Algorithm

The algorithm used in the coding is shown in figure 5.

```
faceDetailsAvg.avg();
Log.v("Face information: ", "Being saved");
Log.v("eyes distance ratio", String.format("%.2f", faceDetailsAvg.eyesDistanceRatio));
Log.v("righteyenosebase ratio", String.format("%.2f", faceDetailsAvg.rightEyeNoseBaseDistanceRatio));
Log.v("lefteytenosebase ratio", String.format("%.2f", faceDetailsAvg.leftEyeNoseBaseDistanceRatio));
Log.v("nosebasemouth ratio", String.format("%.2f", faceDetailsAvg.noseBaseMouthDistanceRatio));
Log.v("rightmouthleft ratio", String.format("%.2f", faceDetailsAvg.rightMouthLeftMouthDistanceRatio));
Log.v("rightmouthbottom ratio", String.format("%.2f", faceDetailsAvg.rightMouthBottomMouthDistanceRatio));
Log.v("leftmouthbottom ratio", String.format("%.2f", faceDetailsAvg.leftMouthBottomMouthDistanceRatio));
Log.v("righteyemouth ratio", String.format("%.2f", faceDetailsAvg.rightEyeMouthDistanceRatio));
Log.v("lefteyemouth ratio", String.format("%.2f", faceDetailsAvg.leftEyeMouthDistanceRatio));
saveFaceInformation();
cleanFaceDetailsArray();
index = 0;
}
```

Figure 5. Get Average Calculation for Face Distance

4.4. Activities Testing

The aim of the application is to provide secure access to the applications. Each activity that provides this security are tested by 3 participant users. The results of the applications indicated that the applications successfully run by these 3 participants. The results of the activities test are shown in tables below. The table are arranged in the order of the activity sequence.

Table 1. Set PIN Test Results

No.	Test Procedure	Expected Results	Test Results	
			Success	Failure
1.	Click on 'Authy Lock' icon to launch the application	The Set PIN activity will be displayed	✓	
2.	On the Set PIN page	The interface will display	✓	

		message request for PIN		
--	--	-------------------------	--	--

Table 2. Face Tracking Results

No.	Test Procedure	Expected Results	Test Results	
			Success	Failure
1.	PIN Confirmation	Request user's camera access will display	✓	
2.	Enable Camera Access	The smartphone's front camera will come in view	✓	
3.	Detect Face image	Capture and save the face value	✓	

Table 3. List Application Results

No.	Test Procedure	Expected Results	Test Results	
			Success	Failure
1.	Enable 'Authy Lock' Access to write over applications	List of installed applications will be displayed	✓	
2.	Click button to enable and disable 'Authy Lock'	Message for secure access will be displayed	✓	

Table 4. Face Recognition Results

No.	Test Procedure	Expected Results	Test Results	
			Success	Failure
1.	Click button to enable 'Authy Lock' on application	Front camera will be displayed	✓	
2.	Use unrecognized face	PIN Lock will be displayed	✓	
3.	Does not use face	PIN Lock will be displayed	✓	
4.	Use recognized face	Enable application access	✓	
5.	Input PIN lock	Enable application access	✓	

5. Conclusions

Authentication Lock was design and develop for security purpose. The usage of biometric security for locking the application provides new level of security for user. Furthermore, the application is open source and free for user to use. In addition, this application run in background all day and can be used anytime. More importantly, the integration of PIN lock provides alternative access to the locked application if the front camera could not identify the face.

Acknowledgement

The implementation and documentation of this project development would not have proceeded without the support from individuals.

First of all, I would like to thank my parents for giving support through the intense development of the project.

I would also like to thank my supervisor Mr. Firoz bin Yusuf Patel Dawoodi, for without his guidance through the development and documentation process, this project was able to proceed smoothly.

Last but not least, I would also like to thank the Faculty of Computing for giving me such great knowledge.

References

1. Faclock.mobi, "Facelock", Facelock.mobi. [online] Available at <http://www.facelock.mobi/> [Accessed: May 15,2017]
2. L. Graham, "Biometrics: The future of digital security," CNBC.com [online] Available at <http://www.cnbc.com/2016/04/05/biometrics-future-of-digital-cyber-security.html> [Accessed: May 15,2017]
3. A. Geitgety, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning", medium.com [online] Available at <https://medium.com/@ageitgy/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78> [Accessed: May 15,2017]
4. M. Adayemi, "Machine Learning for Android Developers with the Mobile Vision API— Part 1—Face Detection", hackernoon.com [online] Available at <https://hackernoon.com/machine-learning-for-android-developers-with-the-mobile-vision-api-part-1-face-detection-e7e24a3e472f> [Accessed: May 15,2017]