

Investigación:

Reconocimiento de huella digital

Versión: 1

TP inicial - Laboratorio de Construcción de Software-Proyecto Profesional I

Alumnos:

- Hirschfeld Agustina Camila (camilahirschf@gmail.com)
- Veron Adrián (adrianveron4@hotmail.com)
- Cabrera Juan Manuel (amnauju5@gmail.com)

Profesores:

- Juan Carlos Monteros (jcmonteros@campus.ungs.edu.ar)
- Evelyn Aragon (eve_aragon@hotmail.com)

Revisiones del documento

Fecha	Revisión	Descripción
25/03/2024	1	Primera versión del documento.

Introducción e historia:

El reconocimiento a través de huella digital se implementa en dispositivos móviles desde el Año 2013 con el lanzamiento del iPhone 5s. Previamente, entre el año 2011 y 2012 hubo algunos dispositivos móviles con esta tecnología, pero no llegó a destacar en el mercado ya que esta funcionalidad no era primordial según la demanda de los usuarios en aquellos años.

En cuanto a reconocimiento de huella digital se refiere, nos podemos encontrar con 4 tipos diferentes de sensores:

- **Sensores ópticos:**
Los sensores ópticos son los más antiguos entre los sensores de huellas dactilares. Funcionan mediante un microchip sensible a la luz (ya sea un dispositivo de carga acoplada o un sensor de imagen CMOS similar a los usados por las cámaras de tipo Réflex). Cuando se coloca un dedo sobre el escáner, éste ilumina la superficie de la piel y captura una imagen de la huella digital.
- **Sensores capacitivos:**
Estos escáneres miden las pequeñas diferencias en la carga eléctrica a través de la superficie de la piel. Cuando se coloca un dedo sobre el escáner, las crestas de la huella dactilar tocan la superficie del sensor, mientras que los valles no. Esto crea un patrón de variaciones de capacitancia que puede traducirse en una imagen digital de la huella digital.
- **Sensores ultrasónicos:**
Estos escáneres utilizan ondas de ultrasonido para detectar los detalles de las huellas dactilares. El escáner envía un pulso ultrasónico que rebota hacia el sensor después de tocar las crestas y valles de la huella digital. Las ondas sonoras que regresan se capturan y procesan para formar una imagen.
- **Sensores térmicos:**
Estos son menos comunes y funcionan detectando las diferencias de temperatura entre las crestas y los valles de la huella digital.

Implementación en celulares:

Para la implementación de reconocimiento vía huella digital en celulares, primeramente, tenemos que separar entre aquellos dispositivos que tienen sensor para esta funcionalidad y aquellos que no. Por un lado, se encuentran los dispositivos iPhone que desde el año 2022 ya no cuentan con sensor de huella digital y el reconocimiento biométrico lo realizan mediante Face ID (Reconocimiento facial). Por otro lado, se encuentran los dispositivos Android contando con una gran variedad de marcas y son contados los casos de dispositivos Android que no cuenten con sensor de huella digital.

Una vez mencionado lo anterior, podemos determinar que lo más sensato para el desarrollo solicitado será optar por realizar el mismo de forma tal que sea compatible con dispositivos Android.

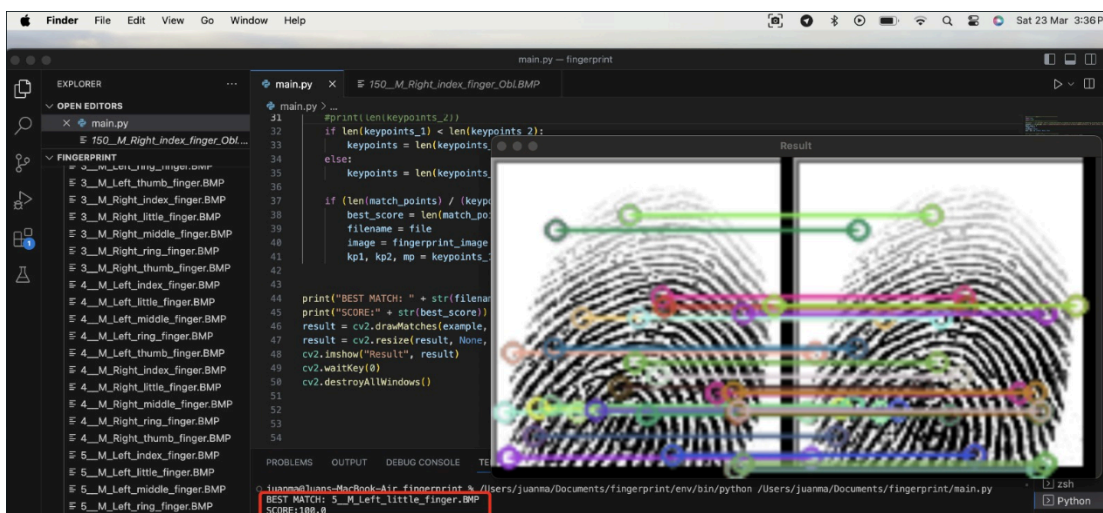
En un primer acercamiento, para poder implementar un login con reconocimiento de huella digital en dispositivos Android podemos definir que el desarrollo se deberá realizar mediante Android Studio (Con el lenguaje que corresponda, comúnmente se utiliza Java/Kotlin) integrando las funcionalidades de Machine Learning, realizadas en python, a través de una API.

Por otro lado, también se evaluó la opción de realizar un desarrollo de tipo web en lugar de APK (Android Application Package) pero el inconveniente con esta opción es que el uso de reconocimiento biométrico en desarrollos de tipo web es algo muy poco común.

Obstáculos detectados:

Como se mencionó previamente en la sección “Introducción e historia” hay varios tipos de sensores para el reconocimiento de huella digital y el utilizado por los celulares es el de tipo capacitivo.

En un principio comenzamos el desarrollo de la funcionalidad sin conocer esta diferencia entre tipos de sensores y avanzamos con un código en Python utilizando la librería Opencv que nos permitía comparar huellas y dar un porcentaje de coincidencia con un alto nivel de precisión, el inconveniente se encuentra en que esto solo funciona para los datos que se obtienen de los sensores de tipo óptico.



Code File Edit Selection View Go Run Terminal Window Help

main.py -- fingerprint

main.py x 150_M_Right_Index_Finger_Obl.BMP

main.py > ...

```
1
2 #-----Imports-----
3 import os
4 import cv2
5 #-----Variables-----
6 #example = cv2.imread('/Users/juanma/Documents/fingerprint/archive/SOCOFing/SOCOFing/Altered/Altered-Hard/150_M_Right_index_finger_Obl.BMP')
7 example = cv2.imread('/Users/juanma/Documents/fingerprint/archive/SOCOFing/Test/5_M_Left_little_finger.BMP')
8 best_score = 0
9 image = None
10 filename = None
11 kp1, kp2, mp = None, None, None
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

juanma@Juans-MacBook-Air fingerprint % /Users/juanma/Documents/fingerprint/env/bin/python /Users/juanma/Documents/fingerprint/main.py

The fingerprint does not match, access denied!

SCORE:0

juanma@Juans-MacBook-Air fingerprint % /Users/juanma/Documents/fingerprint/env/bin/python /Users/juanma/Documents/fingerprint/main.py

BEST MATCH: 5_M_Left_little_finger_Obl.BMP

SCORE:150.0

Granted access!

2024-03-24 11:11:05.677 Python[20613:696772] WARNING: Secure coding is not enabled for restorable state! Enable secure coding by implementing NSApplicationDelegate.applicationSupportsSecureRestorableState: and returning YES.

juanma@Juans-MacBook-Air fingerprint % /Users/juanma/Documents/fingerprint/env/bin/python /Users/juanma/Documents/fingerprint/main.py

BEST MATCH: 5_M_Left_little_finger_Obl.BMP

SCORE:50.0

Granted access!

2024-03-24 11:12:10.791 Python[20632:697434] WARNING: Secure coding is not enabled for restorable state! Enable secure coding by implementing NSApplicationDelegate.applicationSupportsSecureRestorableState: and returning YES.

juanma@Juans-MacBook-Air fingerprint % /Users/juanma/Documents/fingerprint/env/bin/python /Users/juanma/Documents/fingerprint/main.py

BEST MATCH: 5_M_Left_little_finger_Obl.BMP

SCORE:170.66666666666667

Granted access!

2024-03-24 11:12:46.449 Python[20650:697756] WARNING: Secure coding is not enabled for restorable state! Enable secure coding by implementing NSApplicationDelegate.applicationSupportsSecureRestorableState: and returning YES.

juanma@Juans-MacBook-Air fingerprint % /Users/juanma/Documents/fingerprint/env/bin/python /Users/juanma/Documents/fingerprint/main.py

BEST MATCH: 5_M_Left_little_finger.BMP

SCORE:100.0

Granted access!