Password Manager with Facial Recognition

Final Project Report

BINUS
UNIVERSITY
INTERNATIONAL

Edward Raphael (2702355391)

L2CC

Jude Joseph Lamug Martinez MCS

Object Oriented Programming

Table of Contents

Abstract

This report was created as a documentation of the Final Project in an Object Oriented Programming course. The contents of this project will showcase the student's abilities in coding using the Java Programming Language. In this particular report, the project consists of a facial recognition system as a login system to a simple password manager.

Project Specifications

Introduction

The requirements for this project is to apply the topics learned within the semester about Object Oriented Programming into a project that solves any modern problems. The problem chosen in this project was in regards to the security of your account information.

Idea Inspiration

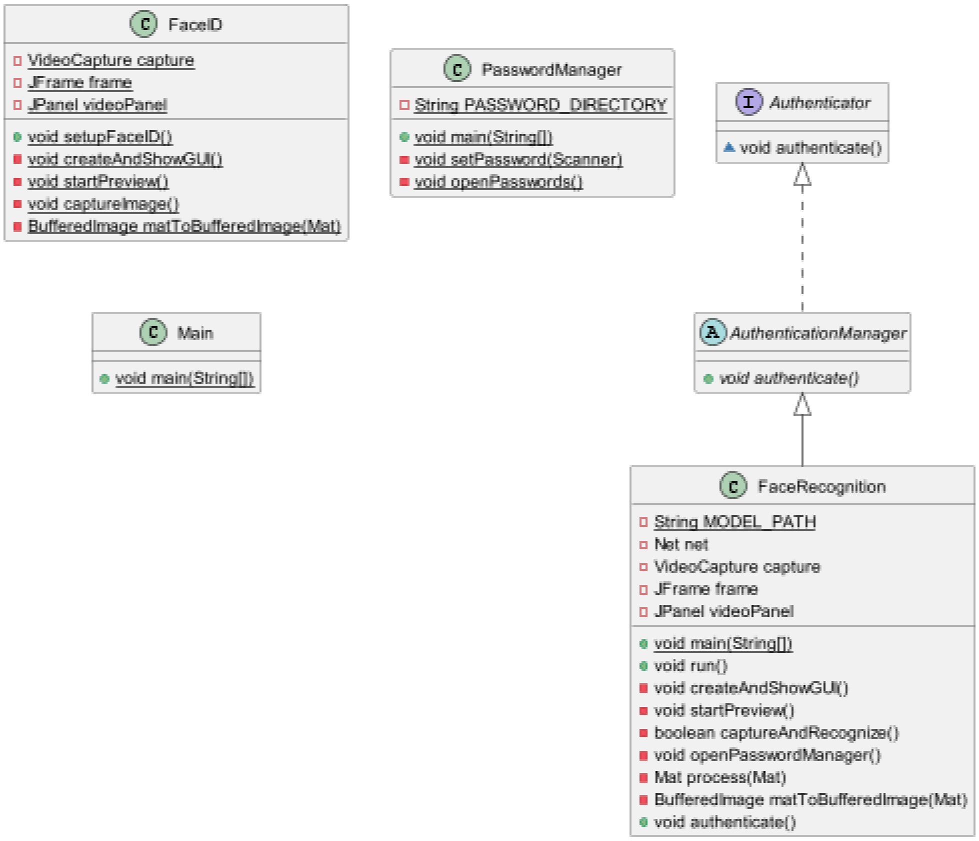
The main attraction of this project is essentially the facial recognition feature. However, I wanted to add a form of management system that can be accessed through the authentication of facial recognition. Thus, I chose a password management system to provide a platform that stores private information in a hidden manner.

Solution Scheme

The user would first be given the option to set up their face id. If the program detects that there is no stored image of the user, then the password management system won't be visible. Click the button to set up your face id and a window will emerge with a button to capture your image, this image will be stored in a local file as the comparing image for your next login. The program will redirect itself to the password management system once the requirements are done. The system first loads up an OpenCV library and a pre-trained neural network for face recognition.The password management system can now be accessed by taking another image and comparing the euclidean distance as the method of authentication. The features of the management system are fairly simple, you are simply given the options to choose whether you'd like to store password information or to access stored information. The file format at which the data is stored is a txt file that is stored in your local file explorer.

Discussion

Class Diagram



Main Components

- Main: This is the main class that runs the whole program. It creates the GUI for the option buttons for the Password Manager and Face ID setup.
- FaceRecognition: This class handles the face recognition and authentication while accessing the password manager. It captures images from the webcam, processes the captured image and

compares them with a reference image to complete the authentication. OpenCV is used here for image processing and deep learning-based face recognition.

- FaceID: Sets up the face id, and captures the reference image of the user's face. OpenCV is used here to access the webcam and process the image.
- PasswordManager: Handles all the features of the password management system. Saves the stored password into a single file directory in a txt file format.

Program Flow

1. The user first is given the option to set up their face id by clicking a button, A video preview from your webcam in a GUI will then be shown with a button to capture your reference image.
2. Once the setup is completed, the program will redirect you to a new window that enables you to access your password management system.
3. Just like before, a video preview will appear, allowing you to capture another image as the input to the face recognition system.
4. If the comparison of the euclidean distance reaches a certain value, the login process will be successful and the management features will be available to you via terminal.
5. The application will provide you with 2 features, first, you may store your username and password information in a txt file, and secondly, you may open your stored information by clicking the option to open the file directory.

Algorithms and Data Structures

- Face Recognition: It contains an algorithm for comparing euclidean distances and uses a pre-trained learning model for facial recognition.
- Image Processing: Converts the image from Mat to BufferedImage to operate the image processing from OpenCV and Java's BufferedImage for GUI display. This allows for the preview and images to be displayed on the GUI.

- GUI Handling: Manages the event handling for button clicks and window closing. Programmed for direct interaction between the buttons and operations.

Challenges and Limitations

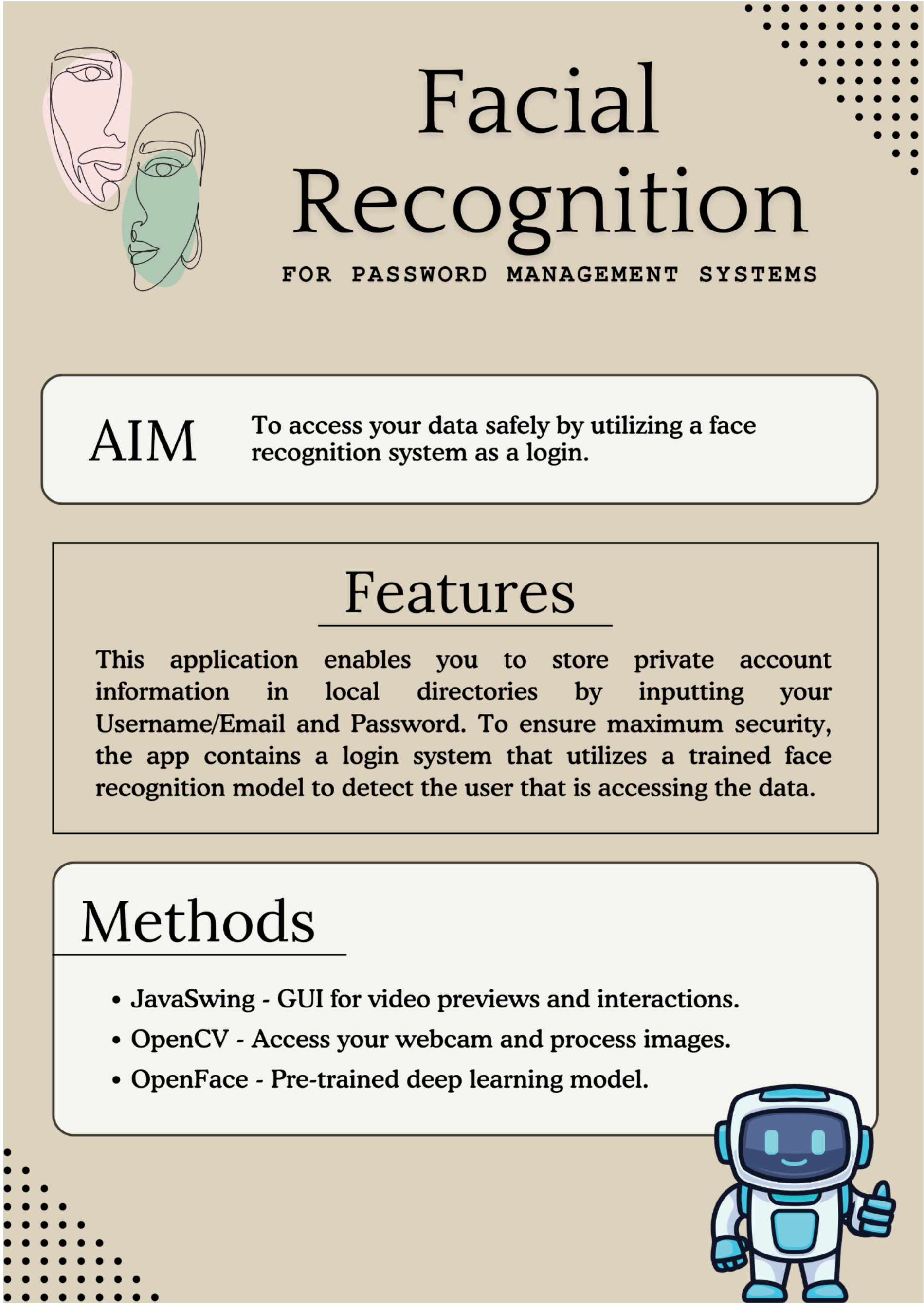
- Recognition Accuracy: The accuracy of the face recognition highly relies on the pre-trained deep learning model, this program relies on the “Openface” learning model.
- Sensitivity to environmental changes: The change in lighting or different facial poses may affect the performance and accuracy of the algorithm.
- Image Quality: The camera quality from device to device may differ, thus making the performance of the facial recognition inconsistent.
- Basic Comparisons: The program relies on the comparison between the captured image and one reference image. This limits the amount of scenarios that the captured image can be taken in.
- Security Concerns: As of right now, the directory of the stored information is not encrypted nor protected in any way, thus opening a path for unknown access.

Evidence of Program

Source Code & Poster

Source Code = https://github.com/Edvade/OOP-Final-Project_2702355391_Edward-Raphael.git

Poster



Facial Recognition

FOR PASSWORD MANAGEMENT SYSTEMS

AIM To access your data safely by utilizing a face recognition system as a login.

Features

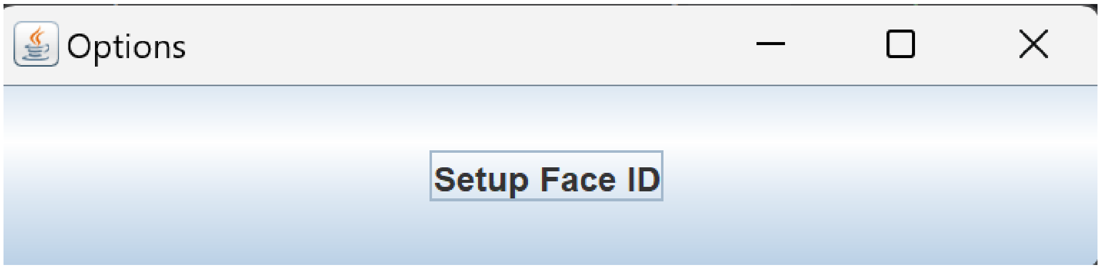
This application enables you to store private account information in local directories by inputting your Username/Email and Password. To ensure maximum security, the app contains a login system that utilizes a trained face recognition model to detect the user that is accessing the data.

Methods

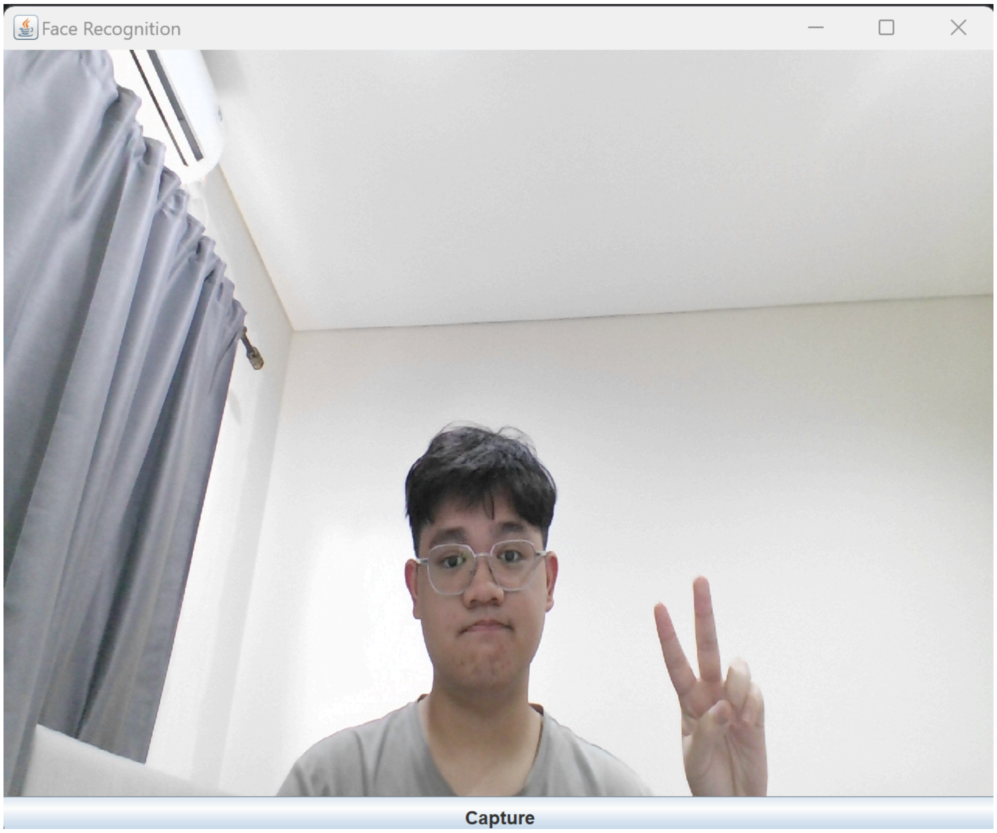
- JavaSwing - GUI for video previews and interactions.
- OpenCV - Access your webcam and process images.
- OpenFace - Pre-trained deep learning model.

Application Tutorial

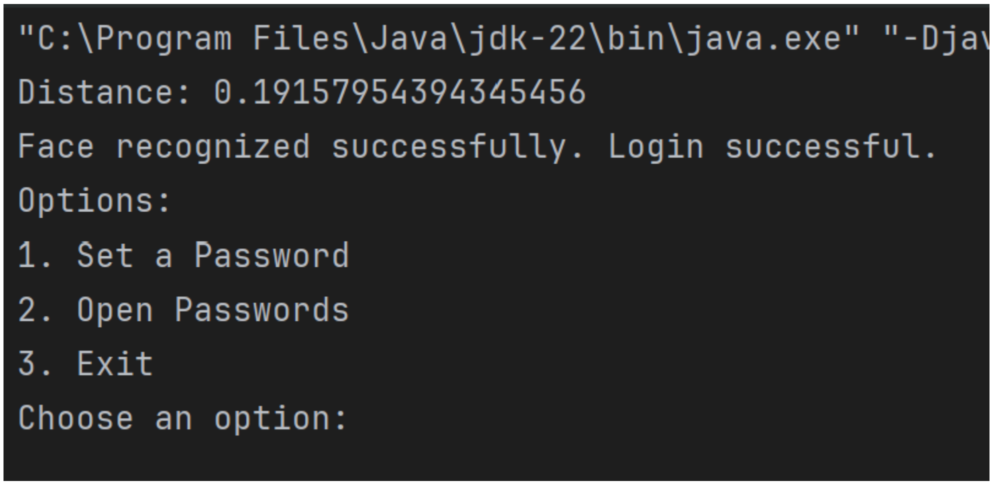
1. Setup Face ID.



2. Once FaceID is set up, you will be redirected to a new page to access the password management system.



3. If the authentication is successful, the features of the management system will appear in the terminal.



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-Djav
Distance: 0.19157954394345456
Face recognized successfully. Login successful.
Options:
1. Set a Password
2. Open Passwords
3. Exit
Choose an option:
```

4. By Clicking the 2nd option, the program will open the directory at which your passwords are stored.

Lesson Learned

I find that java is a much more complicated programming language than python. The main thing for me is the difficulty of importing libraries like OpenCV and using external softwares like maven. To develop this program, we can train the learning model to a more advanced state to increase the accuracy of the recognition.

References

https://opencv-java-tutorials.readthedocs.io/en/stable/08%20-%20Face%20Recognition%20and%20Tracking.html

https://www.youtube.com/watch?v=-5WMXkm-bEE&t=889s

https://github.com/vogster/openface/tree/master

https://www.javatpoint.com/real-time-face-recognition-in-java

https://stackoverflow.com/questions/1595422/face-recognition-in-java