

03 画像認識を用いて本人確認を簡略化するアプリ

6番 江六前 宏昭, 16番 中野 力哉

指導教員 ソソラ

1. 研究の背景

顔認識に興味を持ち、画像認識の仕組みについて学習しながら、アプリを作成したいと考えた。

入試等での本人確認の際に、顔認識を用いることで簡略化することができるのではないかと考えた。

2. 研究概要

顔写真を登録し、画像認識を用いリアルタイムで学習させた人物の中から類似度の高い人物の情報を表示し特定する。

今回はWeb版のアプリとPysimpleGUI版のアプリをそれぞれ作成した。

3. 開発環境

使用言語	Python HTML CSS JavaScript
開発環境	VSCode Visual Studio2017以降
ライブラリ	OpenCV Dlib Flask PysimpleGUI Face_Recognition

3.1 Flaskについて

PythonでWebアプリケーションを開発するための軽量なWebフレームワークである。HTTPリクエストを受け取り、レスポンスを返すことができ、JavaScriptからの通信が可能である。

3.2 PysimpleGUIについて

PythonでGUIを作成できるライブラリである。他のGUIを作成できるライブラリとは異なり、簡単にデスクトップアプリを作成することができます。

3.3 Dlibについて

Dlibは、顔から68個のポイントの情報を取得し（図1,2），そのポイントを元に顔の特徴を抽出しデータと照合し人物の特定を行う。

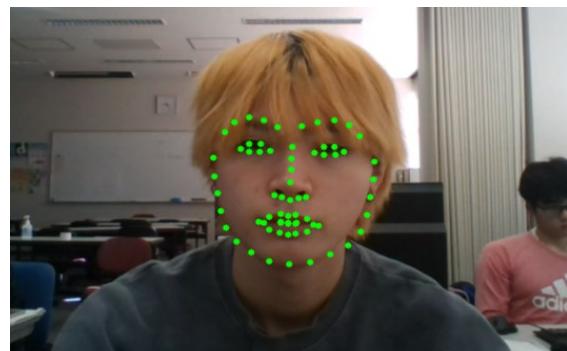


図1. 検証例

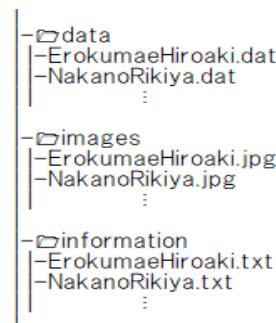


図2. Dlibのランドマーク

3.4 人物情報のファイル

人物情報の登録では画像ファイルと本人情報の書かれたテキストファイルの2つを保存する。

また、画像ファイルに対してエンコードした値をデータファイルとして保存を行う。



を行い.datファイルに保存する。

図3. 情報ファイル一覧

4. 本研究の流れ

ステップ1：データ登録

人物の写真と情報の登録を行う。

アプリでは、写真をカメラで撮影又は既存のファイルから指定する。情報は「名前」、「フリガナ」、「その他本人に関する情報」を入力する(図4, 5)。写真と入力された情報をそれぞれ.jpgファイルと.txtファイルに保存する。



図4. 登録画面 (Web版)



図5. 登録画面 (PysimpleGUI版)

ステップ2：顔検出

カメラを起動し映像を取得する。取得した映像に対してOpenCVやDlibなどのライブラリを使用して顔の特徴を抽出し検出を行う。検出した特徴をもとにラウンドマーク同士の距離を測り、エンコードを行う。

事前に登録していた画像に対しては、顔写真的ファイルを保存する際に顔の検出・エンコード

ステップ3：比較・表示

抽出した顔同士を比較する際にエンコードしたデータから類似度を計算し判別する。判別が成功し、人物が特定された場合は、図6および7に示すように、本人の登録時の顔や情報が画面上に表示される。特定ができなかった場合やカメラに人物の顔が映らなかった場合は、空欄のまま、つまり何も表示されないようにした。



図6. 結果 (Web版)



図7. 結果 (PysimpleGUI版)

5. まとめ

本研究を通して、Dlibを用いた顔のランドマークの取得とFace_Recognitionを用いた類似度の計算についての理解を深めることができた。さらに、FlaskやPysimpleGUIを使用して、アプリを完成させることができた。

6. 参考文献

- [1] <https://www.computervision.zone/courses/face-attendance/>