COMP0016 2022 Team 21 ItchyAI

Development Manual

Overview

Project Title: ItchyAI
Team Member: Lucas Marrie, Yubin Lee, Ruochong Yao

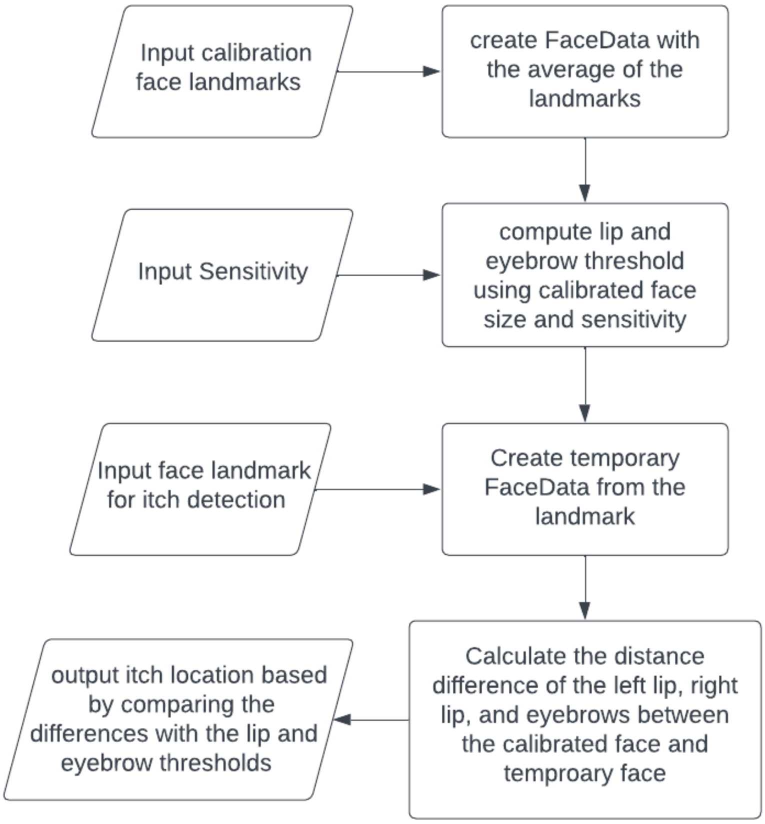
Project Introduction

The need for assistive robotics for people with disabilities has become increasingly apparent in recent years. According to the World Health Organization, around 15% of the world's population lives with some form of disability, and many of these individuals struggle with daily tasks that many of us take for granted. People like Henry who is quadriplegic are paving the way in assistive robotics. This is the first part of a project to help people like Henry with a range of tasks, starting simply to help him scratch when there is an itch on his face. Our project focuses more on AI face detection, in which we build a system that helps to detect an itch on clients' faces. After conducting some thorough research we've discovered that an itch is caused by a stimulation of nerve endings which could be detected via the use of sensory electrodes, however, this puts convenience and safety into question.

Algorithm Architecture

The Itch detection algorithm is comprised of two different classes, the first being the **FaceData class**, it's role being to store and compute information about a face, provided a set of landmarks. The second being the **ItchDetection class**, its role being to compare two different FaceData objects (the calibrated face, and current face to find itch) to deduce the presence and location of an itch.

Below is a flow chart giving a top-level view of the itch detection described above.


Input calibration face landmarks
create FaceData with the average of the landmarks
Input Sensitivity
compute lip and eyebrow threshold using calibrated face size and sensitivity
Input face landmark for itch detection
Create temporary FaceData from the landmark
Calculate the distance difference of the left lip, right lip, and eyebrows between the calibrated face and temproary face
output itch location based by comparing the differences with the lip and eyebrow thresholds

For FaceData section:

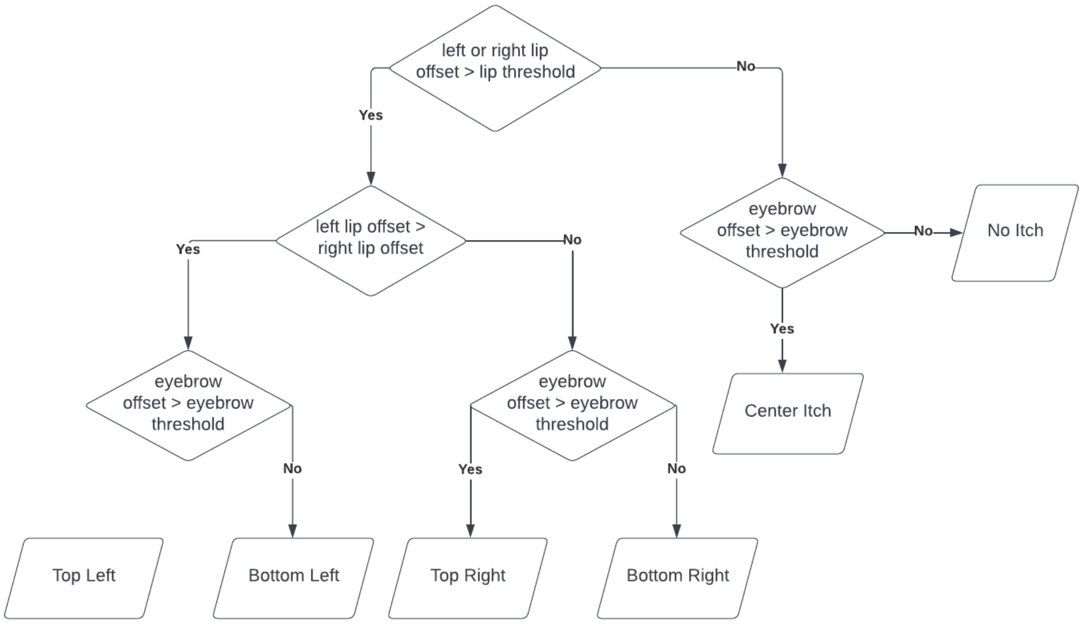
We use face-recognition library based on OpenCV and Dlib. The algorithm will capture the frame of video and analyse it as facial data. Characteristic points on user's face will be converted into landmarks during calibration process.

Three main processes will happen:

1. Creating a Face Data
   a) Averaging landmarks
   b) Caching key points
   c) Calculating the maximum standard deviation
2. Computing key points
   a) Calculating centre
   b) Calculating chin mean distance
   c) Calculating face y angle
3. Computing offsets
   a) Distance to vector
   b) Axis offset
   c) Landmark offset

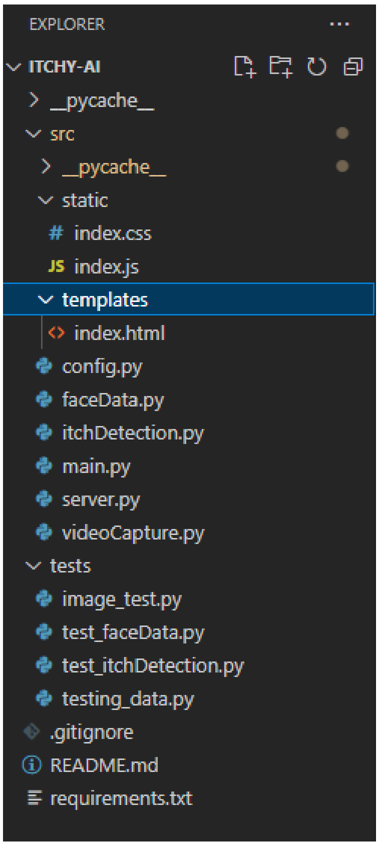
For ItchDetection section:

The graph below shows how the algorithm will transfer the facial data into the position of the itchiness.


left or right lip offset > lip threshold
Yes
No
left lip offset > right lip offset
Yes
No
eyebrow offset > eyebrow threshold
No
No Itch
Yes
Center Itch
eyebrow offset > eyebrow threshold
No
eyebrow offset > eyebrow threshold
Yes
No
Top Left
Bottom Left
Top Right
Bottom Right

Project tree:

The graph below illustrates the main files in the repository.



```
EXPLORER
ITCHY-AI
  __pycache__
  src
    __pycache__
    static
      index.css
      index.js
    templates
      index.html
    config.py
    faceData.py
    itchDetection.py
    main.py
    server.py
    videoCapture.py
  tests
    image_test.py
    test_faceData.py
    test_itchDetection.py
    testing_data.py
  .gitignore
  README.md
  requirements.txt
```

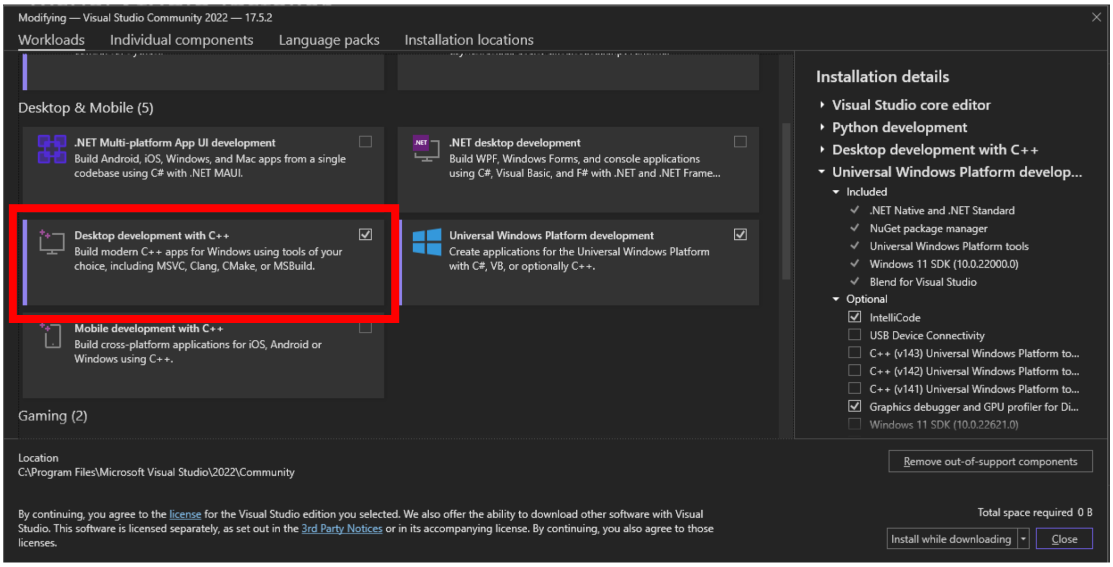
Deployment

ItchyAI Deployment Manual

ItchyAI can recogonize human's face and analyze the facial data to get the position of the itchiness on face.

Prerequisites

- Git
- Python 3.10.9
- Visual Studio with C++ package



Modifying — Visual Studio Community 2022 — 17.5.2

Desktop & Mobile (5)

□ .NET Multi-platform App UI development
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.

□ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

☑ Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

☑ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

□ Mobile development with C++
Build cross-platform applications for iOS, Android or Windows using C++.

Gaming (2)

Installation details

- Visual Studio core editor
- Python development
- Desktop development with C++
- Universal Windows Platform develop...
  - Included
    - ✓ .NET Native and .NET Standard
    - ✓ NuGet package manager
    - ✓ Universal Windows Platform tools
    - ✓ Windows 11 SDK (10.0.22000.0)
    - ✓ Blend for Visual Studio
  - Optional
    - ☑ IntelliCode
    - □ USB Device Connectivity
    - □ C++ (v143) Universal Windows Platform to...
    - □ C++ (v142) Universal Windows Platform to...
    - □ C++ (v141) Universal Windows Platform to...
    - ☑ Graphics debugger and GPU profiler for Di...
    - □ Windows 11 SDK (10.0.22621.0)

Location
C:\Program Files\Microsoft Visual Studio\2022\Community

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 B

Build and Run:

- **Install**

Clone the repository:

```
git clone https://github.com/Itchy-AI/Itchy-AI.git
```

Installing dependencies:

```
pip install -r requirements.txt
```

Dependencies use for the algorithm:

```
click==8.1.3
cmake==3.26.0
colorama==0.4.6
face-recognition==1.3.0
face-recognition-models==0.3.0
Flask==2.2.3
itsdangerous==2.1.2
Jinja2==3.1.2
MarkupSafe==2.1.2
```

```
numpy==1.24.2
opencv-python==4.7.0.68
Pillow==9.4.0
Werkzeug==2.2.3
dlib==19.24.0
```

- **Run ItchyAI**

Running The python test platform

```
python src/main.py
```

Running The local host test server

```
python src/server.py
```

- **Run testing**

Unit Testing

```
cd tests && python -m unittest && cd ..
```

Itegration/image testing

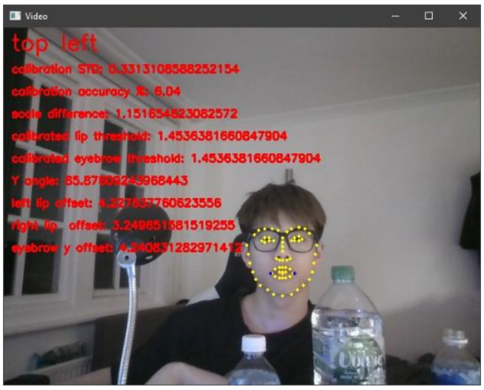
you must populate the test_images folder with test scenarios first following the integration test section of the portfolio website, then you run:

```
python tests/image_test.py
```

ShowCase

Different ItchyAI version

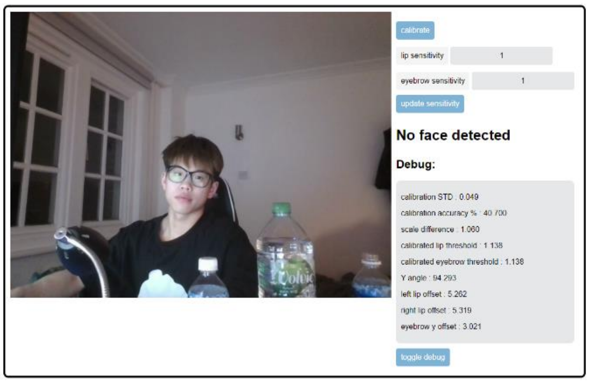
1. Experimental platform version:



```
top left
calibration STD: 0.3313108588252154
calibration accuracy %: 6.04
scale difference: 1.151654823082572
calibrated lip threshold: 1.4536381660847904
calibrated eyebrow threshold: 1.4536381660847904
Y angle: 85.87809243968443
left lip offset: 4.227637760623556
right lip  offset: 3.249651581519255
eyebrow y offset: 4.240831282971412
```

Experimental version

2. Web version：



lip sensitivity 1

eyebrow sensitivity 1

No face detected

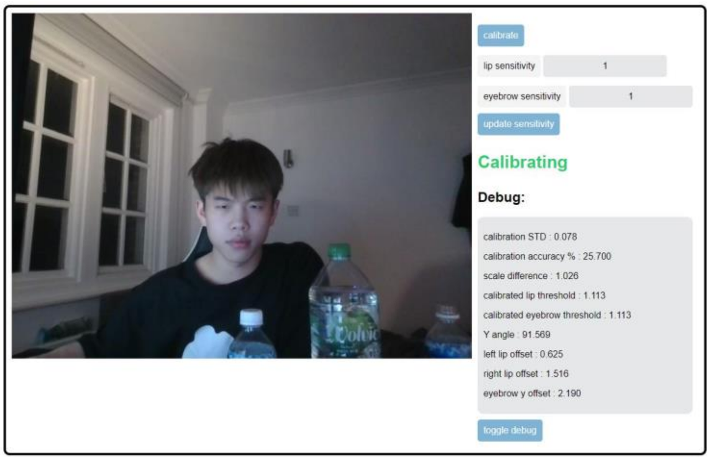
Debug:

```
calibration STD : 0.049
calibration accuracy % : 40.700
scale difference : 1.060
calibrated lip threshold : 1.138
calibrated eyebrow threshold : 1.138
Y angle : 94.293
left lip offset : 5.262
right lip offset : 5.319
eyebrow y offset : 3.021
```

Web version

Detection process

1. Calibration：



lip sensitivity 1

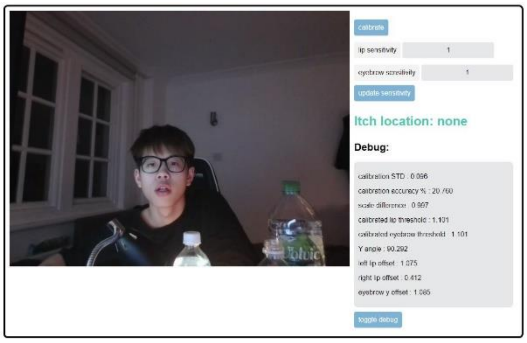
eyebrow sensitivity 1

Calibrating

Debug:

```
calibration STD : 0.078
calibration accuracy % : 25.700
scale difference : 1.026
calibrated lip threshold : 1.113
calibrated eyebrow threshold : 1.113
Y angle : 91.569
left lip offset : 0.625
right lip offset : 1.516
eyebrow y offset : 2.190
```

2. Calculation：



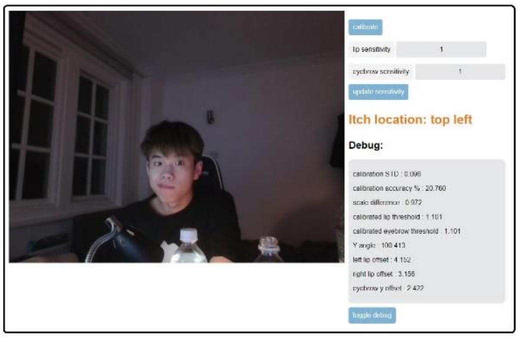
lip sensitivity 1

eyebrow sensitivity 1

Itch location: none

Debug:

```
calibration STD : 0.096
calibration accuracy % : 20.760
scale difference : 0.997
calibrated lip threshold : 1.101
calibrated eyebrow threshold : 1.101
Y angle : 90.292
left lip offset : 1.075
right lip offset : 0.412
eyebrow y offset : 1.085
```



lip sensitivity 1

eyebrow sensitivity 1

Itch location: top left

Debug:

```
calibration STD : 0.096
calibration accuracy % : 20.760
scale difference : 0.972
calibrated lip threshold : 1.101
calibrated eyebrow threshold : 1.101
Y angle : 100.413
left lip offset : 4.152
right lip offset : 3.156
eyebrow y offset : 2.422
```