

座位姿勢改善をサポートする アプリの作成

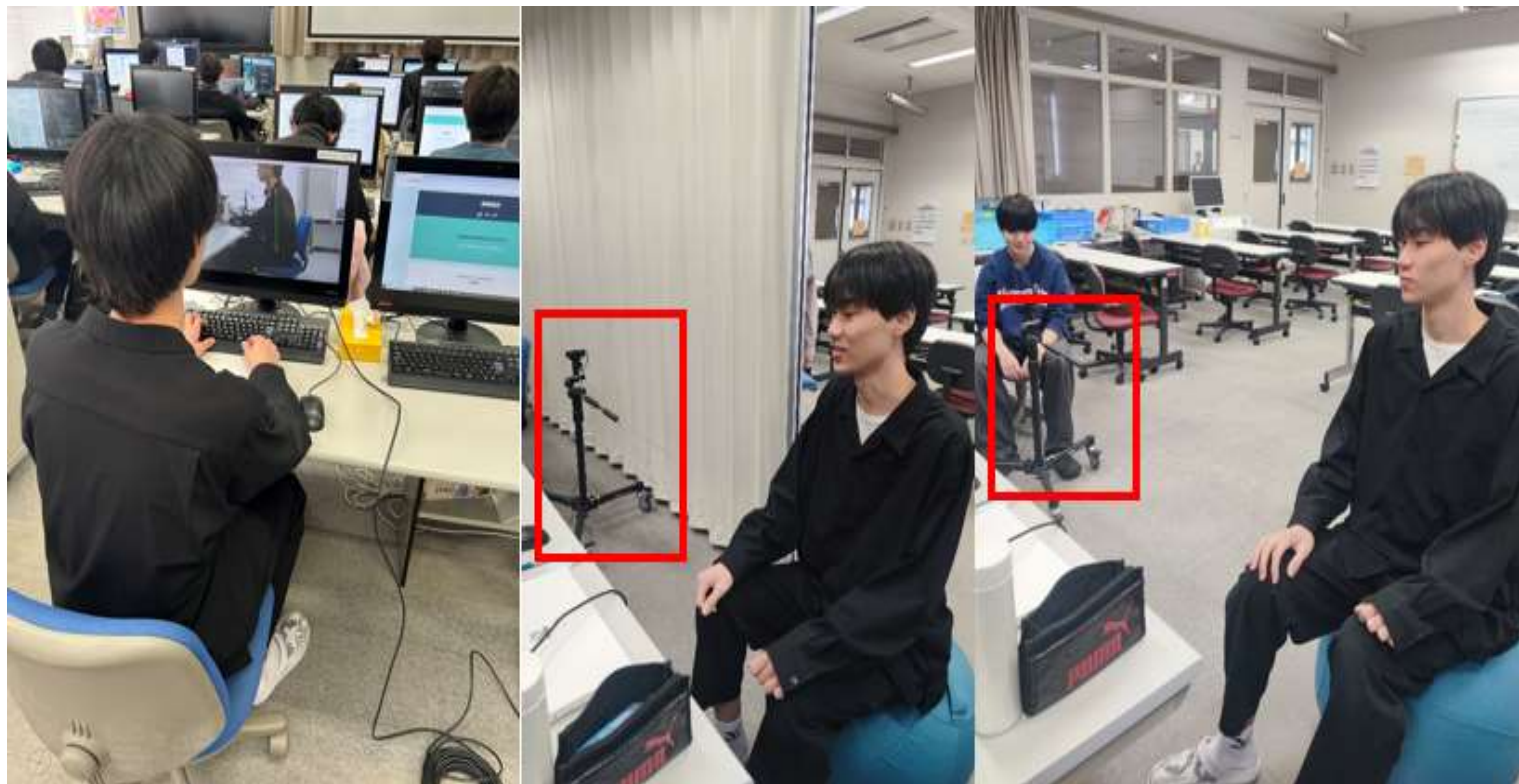
浅沼颯斗 角地湧成 上村武之 高橋遼

指導教員 ソソラ

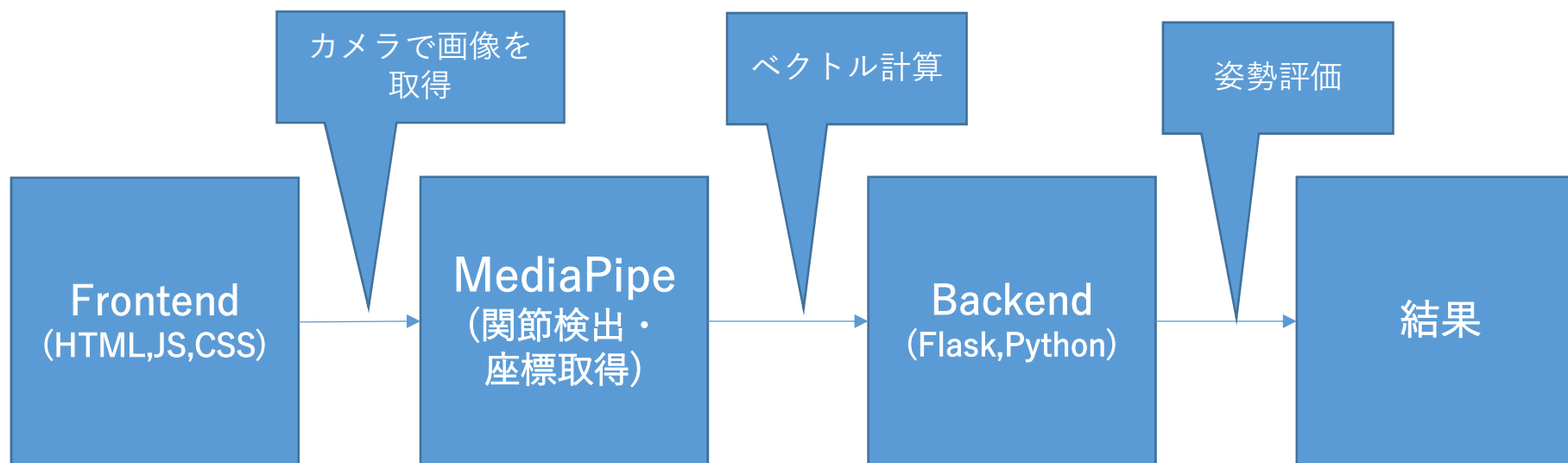
はじめに

- 長時間のデスクワークは腰痛や肩こりなどの健康問題が増える
- MediaPipeの学習を通して姿勢改善に活用できると考えた
- 姿勢を点数化し、姿勢の良し悪しを把握できる
- 自分に最適なグッズを見つけるきっかけ

姿勢を測定している検証の様子



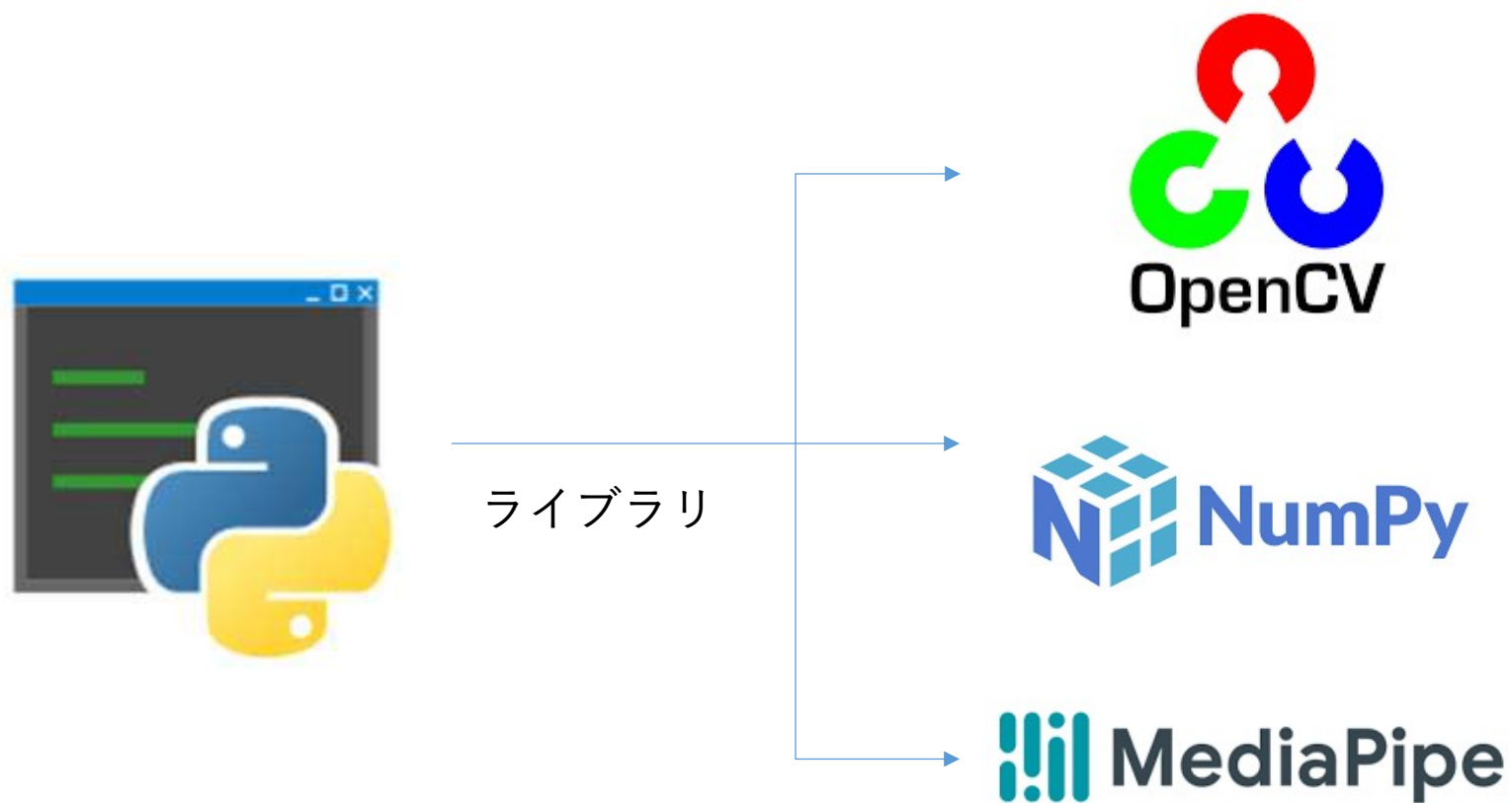
アプリの概要



開発環境

OS	Windows10
エディタ	Visual Studio Code
開発言語	<u>Python</u> , HTML、JS、CSS
使用ライブラリ	<u>OpenCV</u> <u>MediaPipe</u> <u>Numpy</u> Flask

使用技術



使用技術



MediaPipeとは

3.2 MediaPipe について

「MediaPipe」は Google が提供する AI ソリューションである。MediaPipe で取得できる位置情報（ランドマーク）を以下の図に示す。

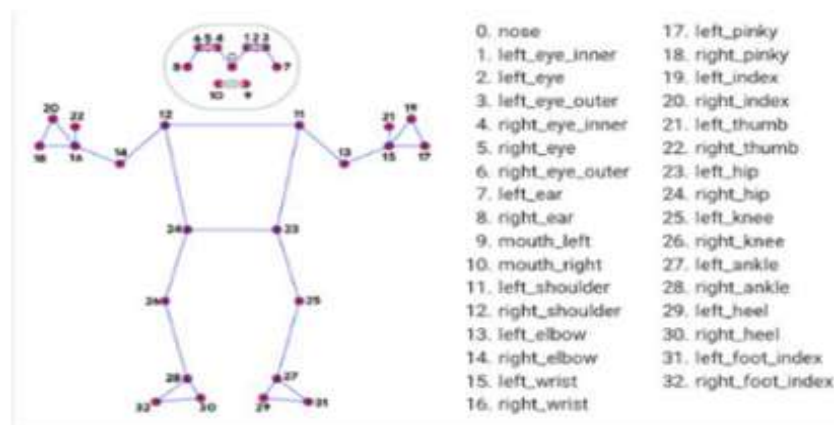
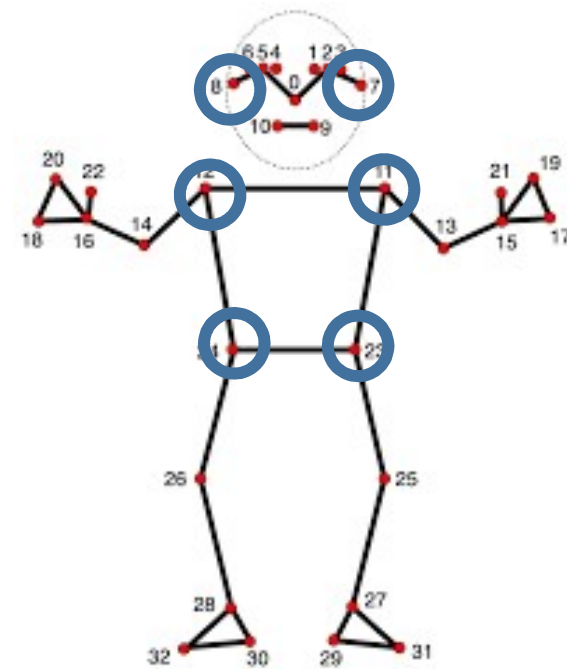


図 3.2 MediaPipe のランドマーク



姿勢の検出・ 角度の計算について

角度の計算（座標の取得）



角度を求める

$\vec{a} = (a_x, a_y), \vec{b} = (b_x, b_y)$ のなす角を ϑ とすると,

$$\cos \vartheta = \frac{a_x b_x + a_y b_y}{\sqrt{a_x^2 + a_y^2} \sqrt{b_x^2 + b_y^2}}$$

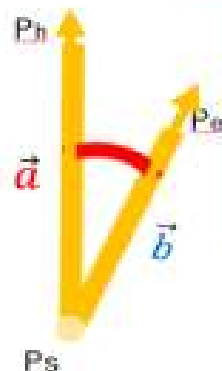
角度計算の関数

$\vec{a} = (a_1, a_2)$ と $\vec{b} = (b_1, b_2)$ のなす角を θ とすると、

$$\cos \theta = \frac{a_1 b_1 + a_2 b_2}{\sqrt{a_1^2 + a_2^2} \sqrt{b_1^2 + b_2^2}}$$

内積(dot Product)

ベクトルの大きさ
(magnitude)



ベクトル間の角度を計算する関数

```
def calculate_angle(v1, v2):
```

```
    """
```

2つのベクトル間の角度を計算します。

Parameters:

v1 (tuple): ベクトル1 (x, y)

v2 (tuple): ベクトル2 (x, y)

Returns:

float: ベクトル間の角度 (度)。

```
    """
```

```
    dot_product = np.dot(v1, v2)
```

```
    magnitude_v1 = np.linalg.norm(v1)
```

```
    magnitude_v2 = np.linalg.norm(v2)
```

```
    cosine_angle = dot_product / (magnitude_v1 * magnitude_v2)
```

```
    angle = np.arccos(np.clip(cosine_angle, -1.0, 1.0))
```

```
    return np.degrees(angle)
```

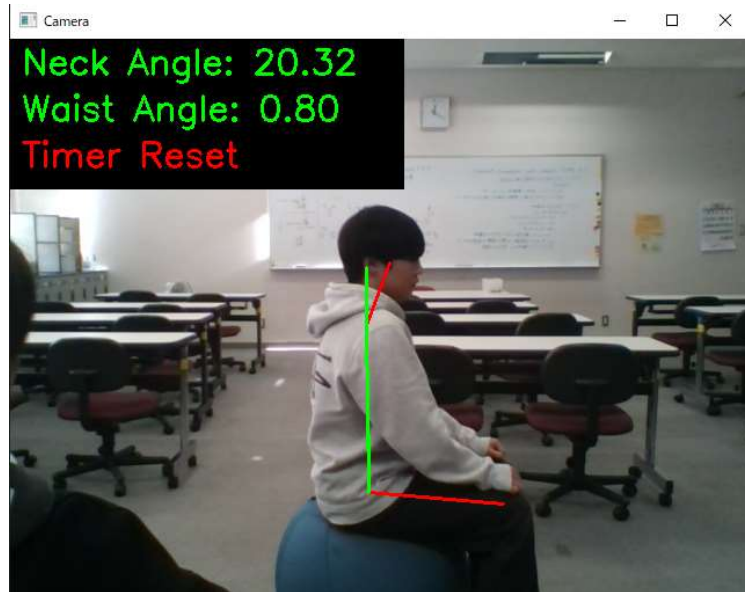
-
1. 首の角度の検出
 2. 腰の角度の検出

角度の計算（体の中心の計算）

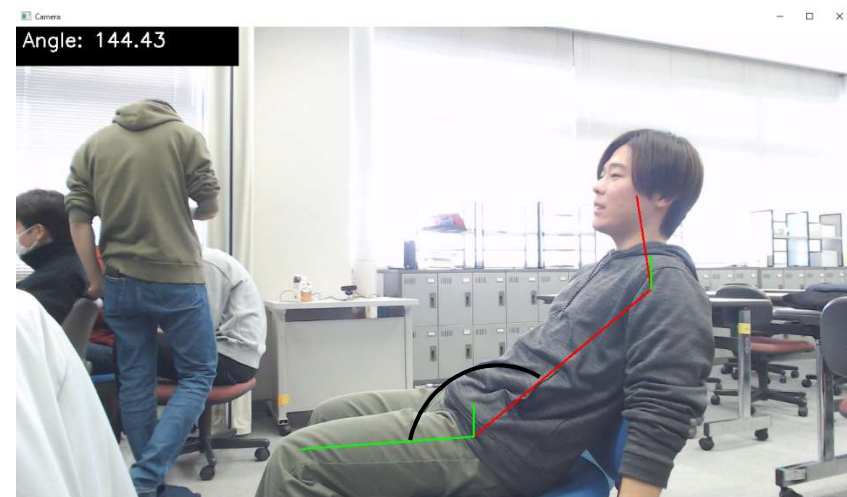
```
ear_mid = (  
    (ear_left_coords[0] + ear_right_coords[0]) // 2,  
    (ear_left_coords[1] + ear_right_coords[1]) // 2,  
)  
shoulder_mid = (  
    (shoulder_left_coords[0] + shoulder_right_coords[0]) // 2,  
    (shoulder_left_coords[1] + shoulder_right_coords[1]) // 2,  
)  
hip_mid = (  
    (hip_left_coords[0] + hip_right_coords[0]) // 2,  
    (hip_left_coords[1] + hip_right_coords[1]) // 2,  
)  
knee_mid = (  
    (knee_left_coords[0] + knee_right_coords[0]) // 2,  
    (knee_left_coords[1] + knee_right_coords[1]) // 2,  
)
```



角度の計算（首と肩のベクトル計算）

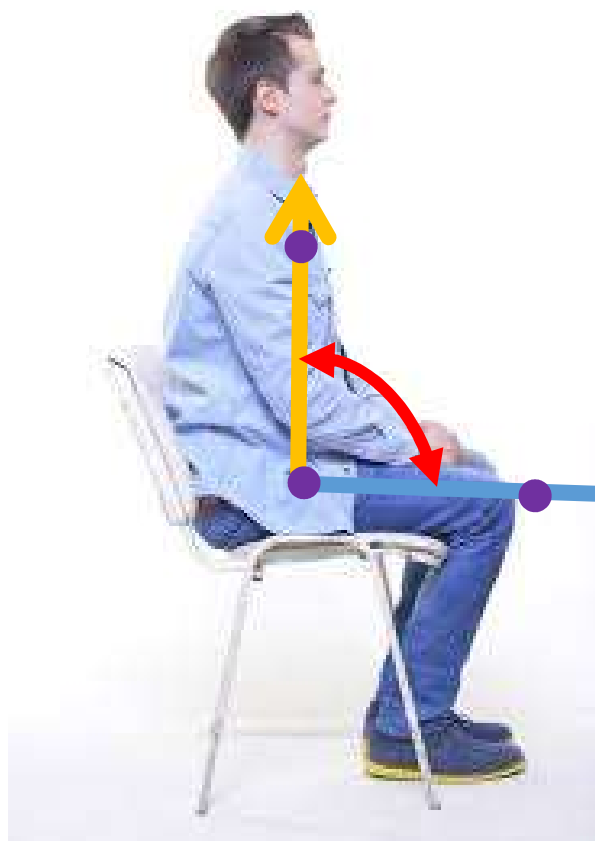


首の角度計算



腰の角度計算

腰の角度の検出方法



```
waist_angle = calculate_angle ( hip_to_shoulder,  
                                hip_to_knee )
```

$\vec{a} = (a_1, a_2)$ と $\vec{b} = (b_1, b_2)$ のなす角を θ とすると、

$$\cos \theta = \frac{a_1 b_1 + a_2 b_2}{\sqrt{a_1^2 + a_2^2} \sqrt{b_1^2 + b_2^2}}$$

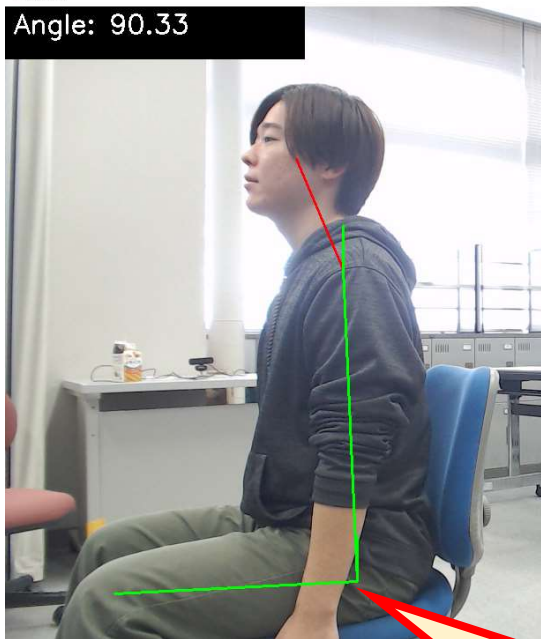
検証

良い姿勢(約 90°)

悪い姿勢(約 144°)



Camera
Angle: 90.33



Camera
Angle: 144.43



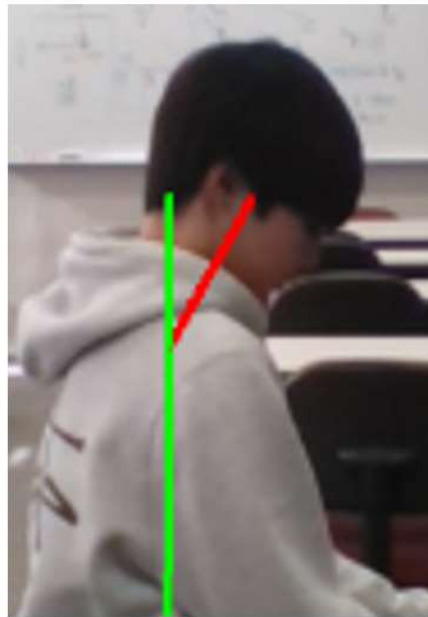
胴体の角度を求め
るためのベクトル

姿勢の評価（点数化）について

減点基準 首

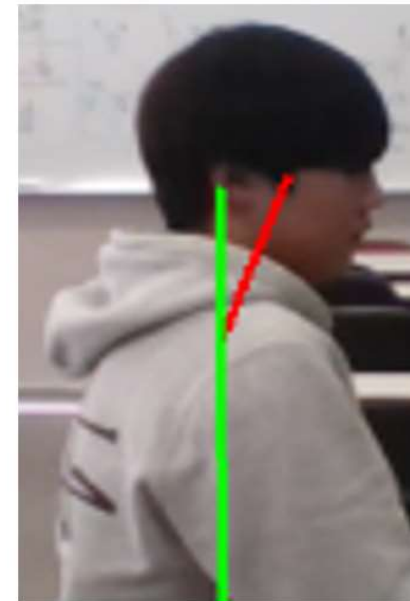
36°

減点される



20°

減点されない



減点基準 腰

低い	中間	高い
		
77°	89°	108°

図5 腰角度の検出様子

減点基準 腰

低い	中間	高い
		
77°	89°	108°

図5 腰角度の検出様子

いい姿勢とは



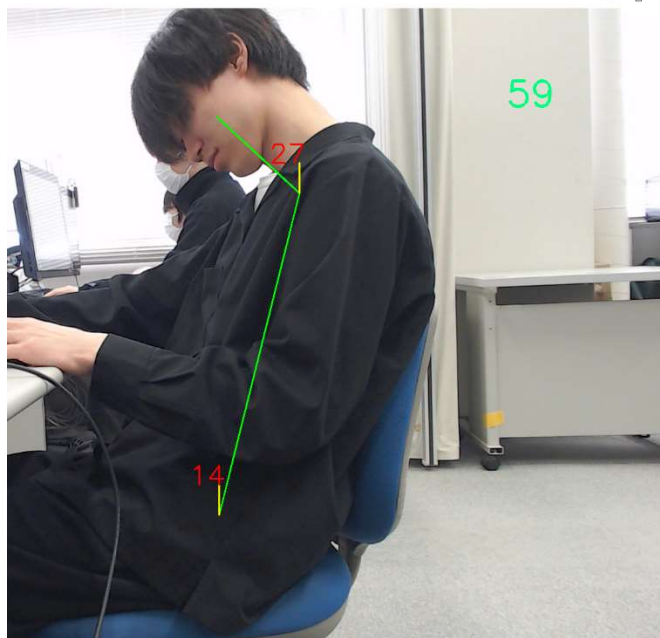
首の角度理想 : $0 \sim 20^{\circ}$

腰の角度の理想 : 90°

点数の算出方法

$100 - (\text{首のずれ}(\alpha) + \text{腰のずれ}(\beta)) = \text{点数}$

$$100 - (27 + 14) = 59 \text{点}$$

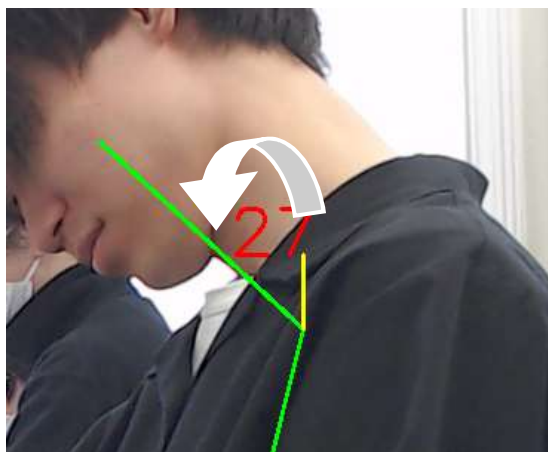


点数の算出方法

首の角度(α)

基準値の 20° より下向きになった角度 1° ごとに-1点

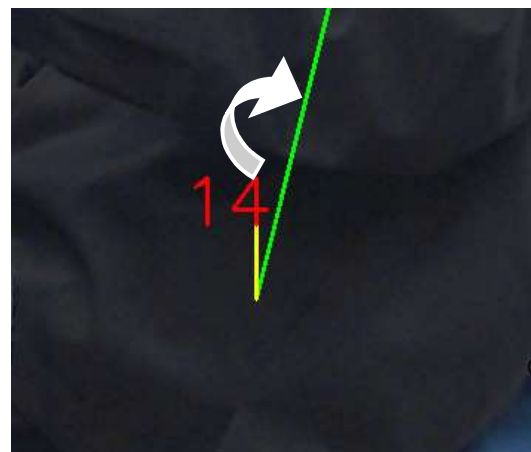
$$\alpha = 47^\circ - 20^\circ = 27^\circ$$



腰の角度(β)

基準値の 90° からのずれ 1° ごとに-1点

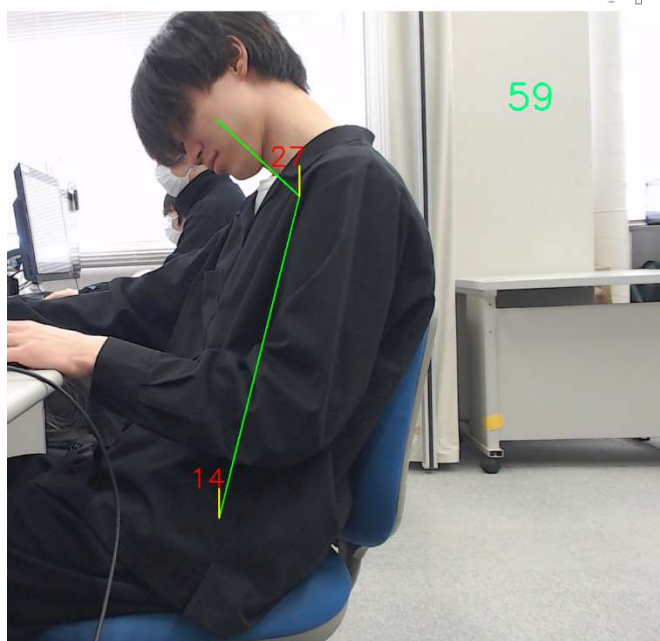
$$\beta = 14^\circ$$



点数の算出方法

$100 - (\text{首のずれ}(\alpha) + \text{腰のずれ}(\beta)) = \text{点数}$

$$100 - (27 + 14) = 59 \text{点}$$



姿勢サポートグッズ

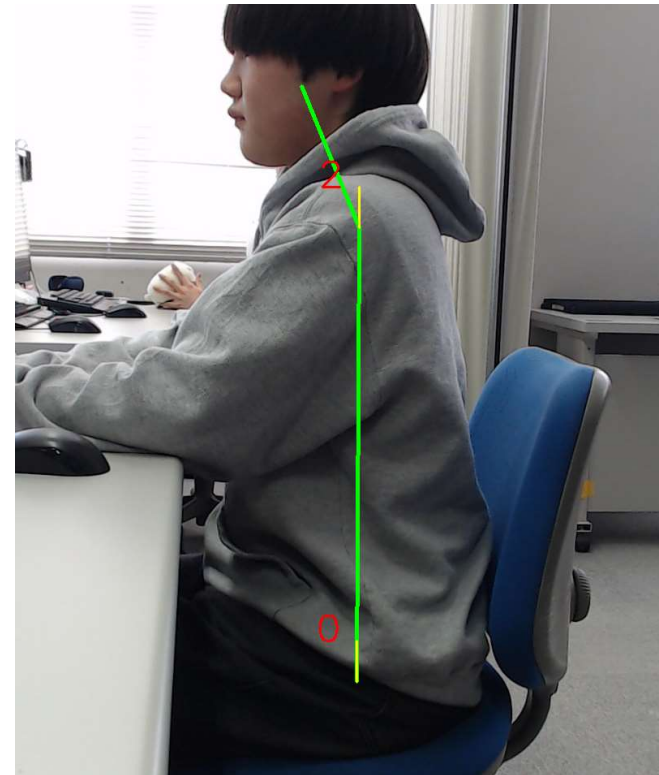


サポートグッズの比較

椅子



97点

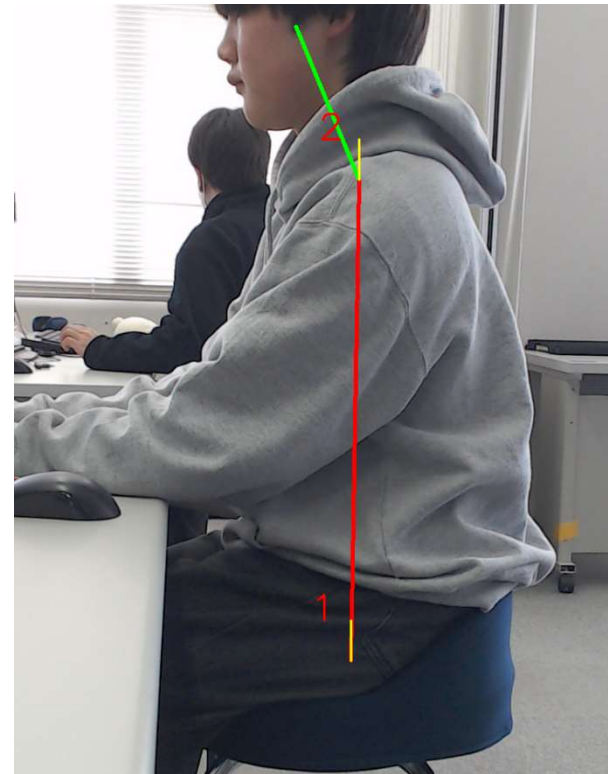


サポートグッズの比較

クッション



97点

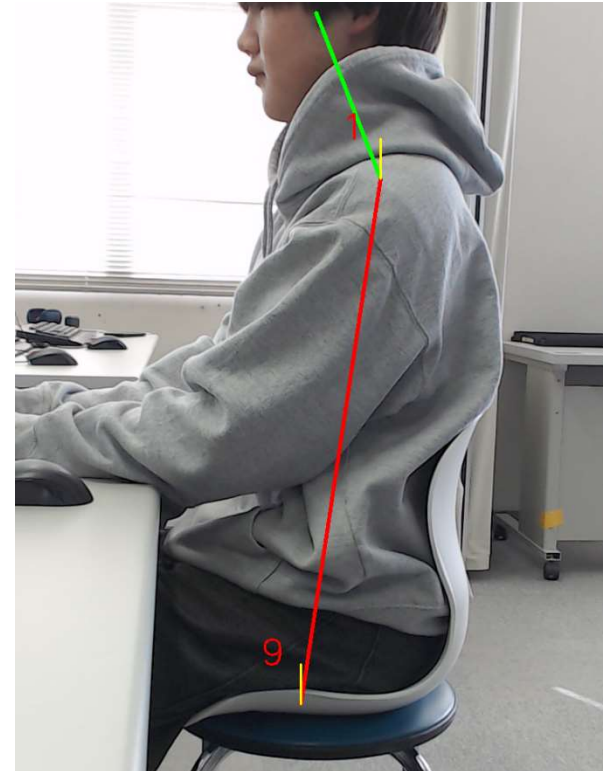


サポートグッズの比較

姿勢サポート椅子



90点

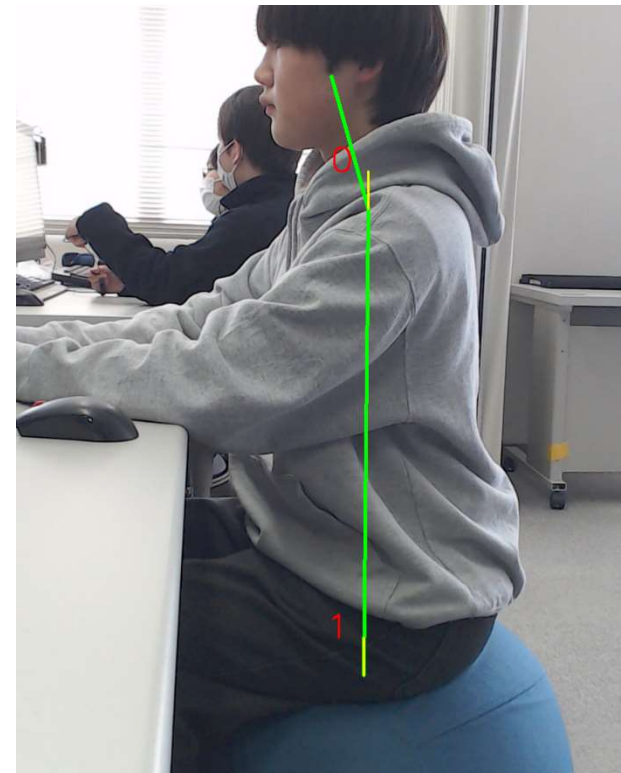


サポートグッズの比較

バランスボール



99点



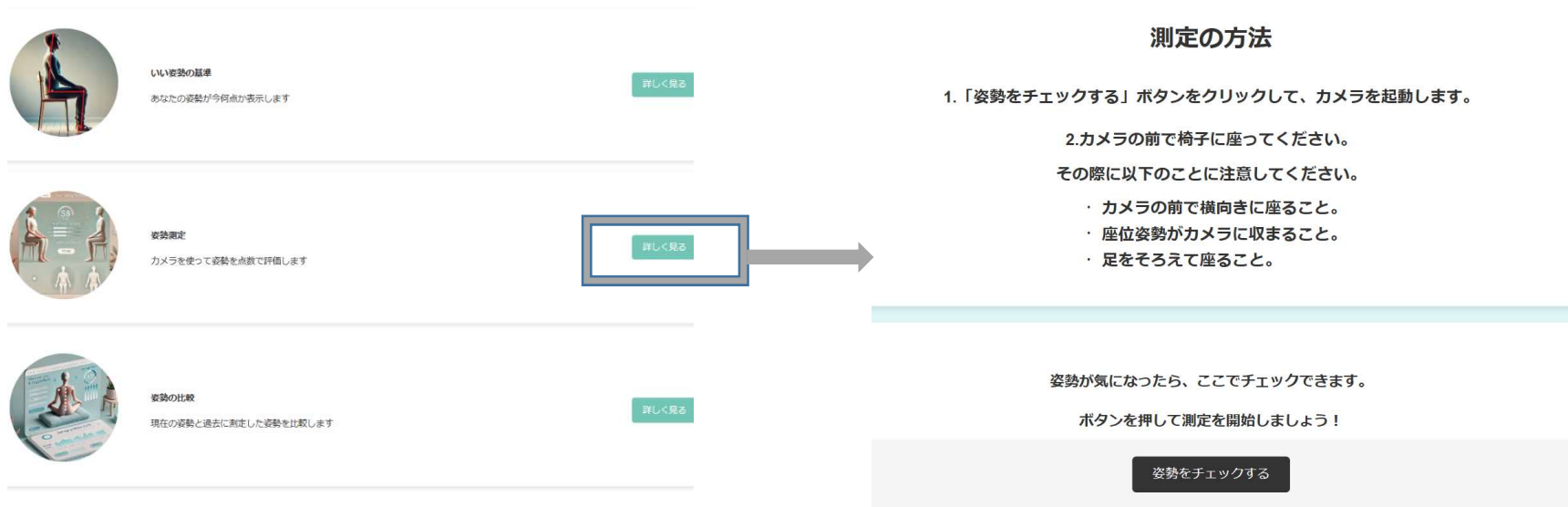
実装したアプリについて

ホームページの構成

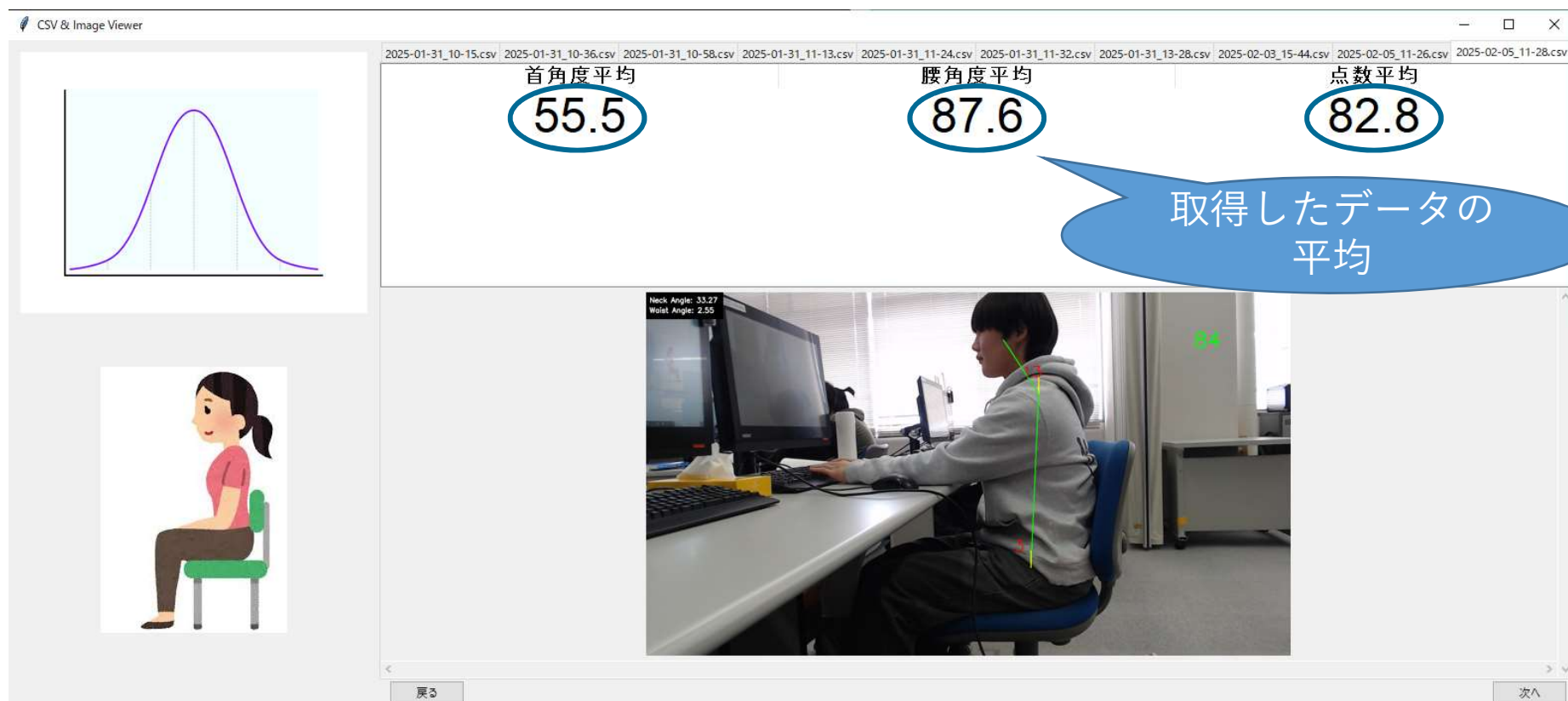
本ホームページは、以下の4つのページで構成されています。

1. トップページ：サイトの入り口となるページ
2. 基準ページ：正しい姿勢について説明しているページ
3. 測定ページ：座位姿勢の測定を行うページ
4. 比較ページ：測定結果を比較するページ

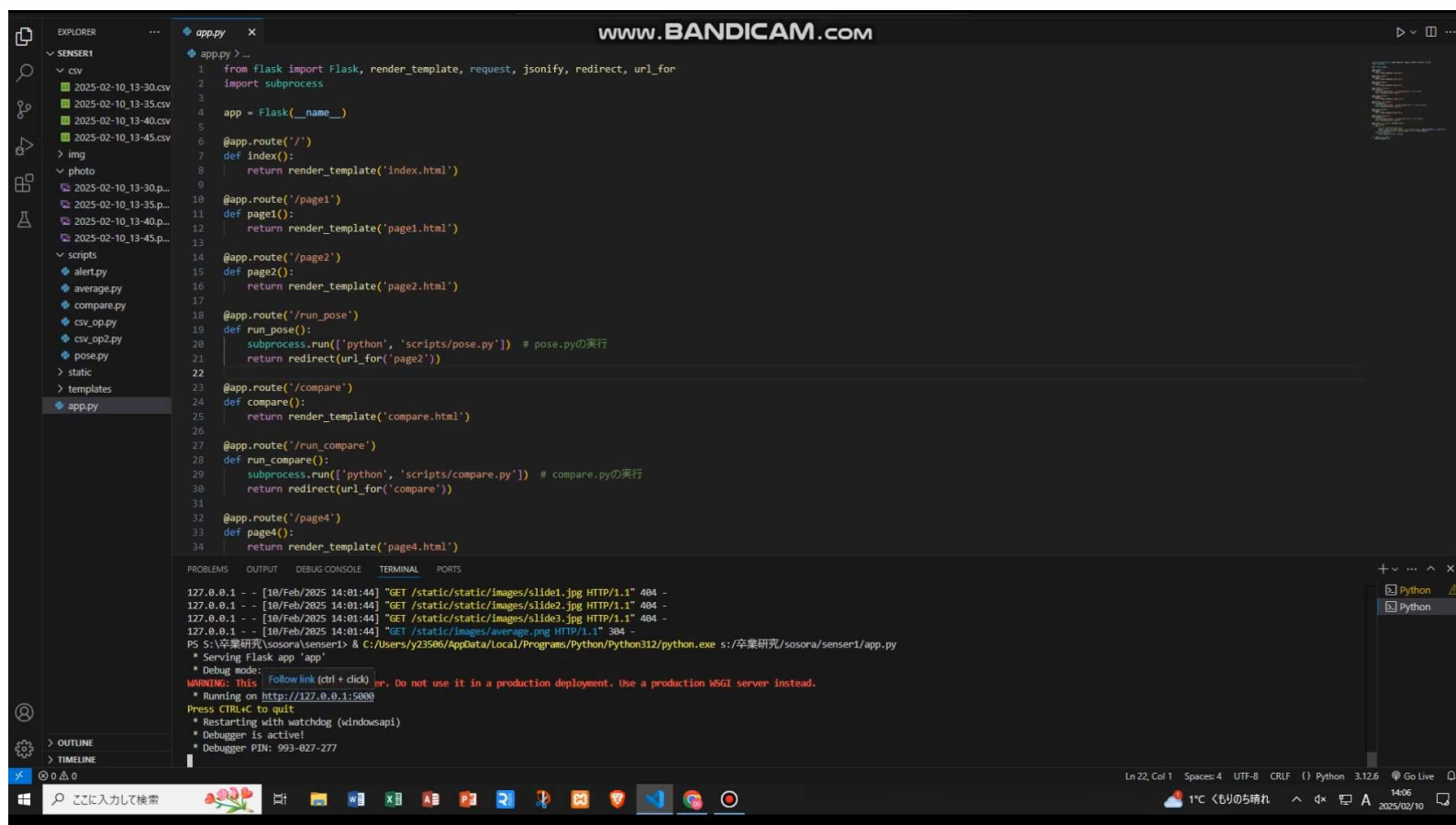
姿勢測定の手順



比較機能の動作



アプリケーションの動作



```
app.py
1 from flask import Flask, render_template, request, jsonify, redirect, url_for
2 import subprocess
3
4 app = Flask(__name__)
5
6 @app.route('/')
7 def index():
8     return render_template('index.html')
9
10 @app.route('/page1')
11 def page1():
12     return render_template('page1.html')
13
14 @app.route('/page2')
15 def page2():
16     return render_template('page2.html')
17
18 @app.route('/run_pose')
19 def run_pose():
20     subprocess.run(['python', 'scripts/pose.py']) # pose.pyの実行
21     return redirect(url_for('page2'))
22
23 @app.route('/compare')
24 def compare():
25     return render_template('compare.html')
26
27 @app.route('/run_compare')
28 def run_compare():
29     subprocess.run(['python', 'scripts/compare.py']) # compare.pyの実行
30     return redirect(url_for('compare'))
31
32 @app.route('/page4')
33 def page4():
34     return render_template('page4.html')
```

```
127.0.0.1 - - [10/Feb/2025 14:01:44] "GET /static/static/images/slide1.jpg HTTP/1.1" 404 -
127.0.0.1 - - [10/Feb/2025 14:01:44] "GET /static/static/images/slide2.jpg HTTP/1.1" 404 -
127.0.0.1 - - [10/Feb/2025 14:01:44] "GET /static/static/images/slide3.jpg HTTP/1.1" 404 -
127.0.0.1 - - [10/Feb/2025 14:01:44] "GET /static/images/average.png HTTP/1.1" 304 -
PS S:\卒業研究\sosora\senser1> & C:\Users\y23506\AppData\Local\Programs\Python\Python312\python.exe s:\卒業研究\sosora\senser1\app.py
* Serving Flask app "app"
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 993-027-277
```

おわりに

- 座位姿勢をサポートするアプリの開発
- MediaPipeやOpenCVについて学習・姿勢解析
- 椅子用クッションなどのグッズの選択肢を提供

ご清聴ありがとうございました

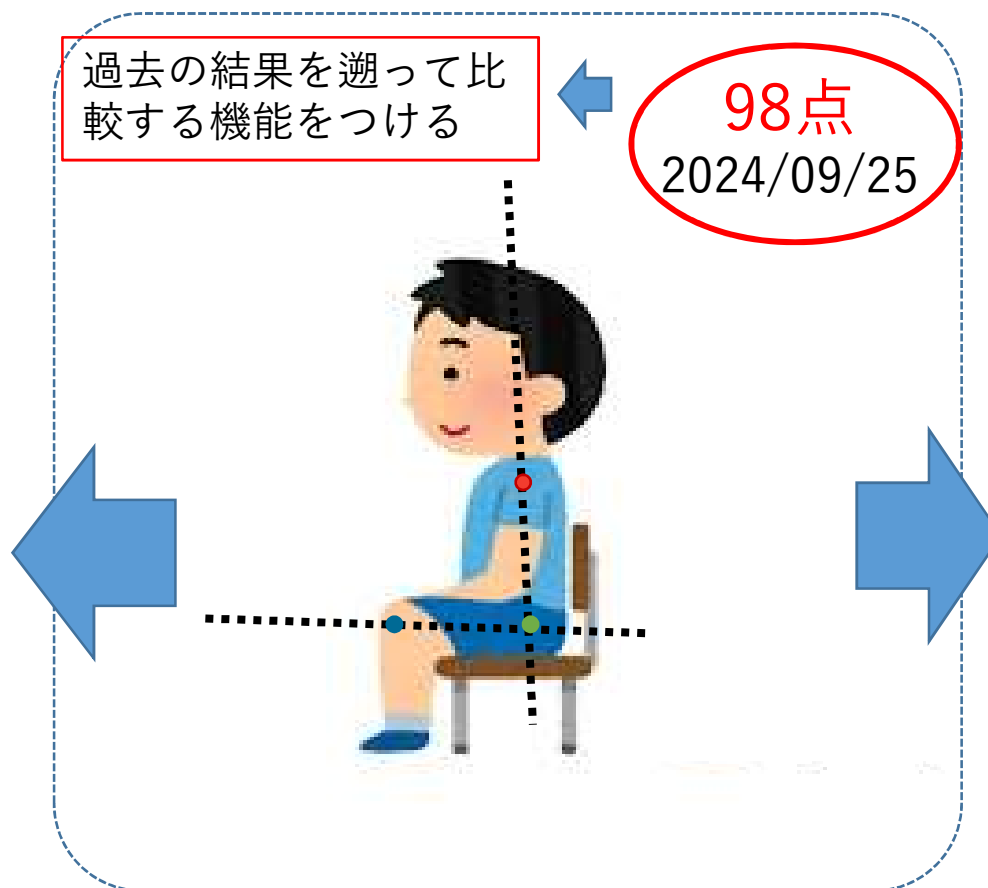
整えた姿勢での評価

測定モード: 「整った座位姿勢」

カメラの前で自分の姿を見ながら
整った姿勢を意識して測定する機能

過去の結果を遡って比較する機能をつける

98点
2024/09/25



自然体の姿勢での評価

測定モード: 「自然な座位姿勢」

カメラの前で何も意識せずに行動し続け、一定の間隔で測定し、姿勢の平均をとり、その数値で評価する

過去の結果を遡って比較する機能をつける

90点
2024/09/25

