

2024 年度（令和 6 年度）

創造工学セミナーⅡ Final Report

# 関節角度による 運動姿勢の評価に関する研究

## 研究メンバー

S521001 青井 一馬

S521007 池田 周磨

## 指導教員

金丸 隆志 教授

## 所属研究室

知能機械研究室

# 目次

第1章 緒論 (青井)	4
1.1 研究背景	4
1.1.1 近年の日本における健康意識の高まり	4
1.1.2 フィットネスクラブ会員数の増加	5
1.1.3 ボディビル人口の増加	6
1.1.4 ボディビルにおけるポーズの重要性	7
1.2 先行研究	8
1.2.1 ダイビングスキルの定着を促す MediaPipe を使用した動作比較システムの開発	8
1.3 本研究の目的	9
第2章 ポーズ認識 (池田)	10
2.1 使用環境	10
2.1.1 MediaPipe について	11
2.2 2次元の関節角度算出手法	13
2.2.1 ピクセルについて	13
2.2.2 関節角度算出手法	15
2.3 2次元の関節角度算出手法と平面角の変化に関する実験	16
2.3.1 実験方法	16
2.3.2 実験結果	17
2.4 推定対象のポーズ	18
2.4.1 評価ポイント	19
2.5 ポーズの判定に用いる角度	20
2.6 2次元座標を用いた機械学習	21
2.6.1 機械学習の概要	21
2.6.2 機械学習の流れ	22
2.6.3 座標から角度への移行手法	23
2.6.4 良いポーズ・悪いポーズの定義	24
2.6.5 機械学習の条件	27
2.6.6 機械学習の結果	28
2.7 ポーズ認識	29
第3章 関節角度による運動姿勢の評価システム精度実験 (青井)	33
3.1 実験1	33
3.1.1 実験方法	33
3.1.2 実験結果1	35

3.1.3 考察.....	36
3.2 実験2：学習データの改善.....	37
3.2.1 実験方法.....	37
3.2.2 実験結果2.....	38
3.2.3 考察.....	39
3.3 本システムに対応しないポーズ.....	40
第4章 RealSense（池田）.....	41
4.1 RealSense について.....	41
4.2 3次元の関節角度算出手法.....	43
4.2.1 スケールの統一.....	43
4.2.2 関節角度算出手法.....	45
4.3 3次元の関節角度算出手法と平面角の変化に関する実験.....	46
4.3.1 実験方法.....	46
4.3.2 実験結果.....	47
4.4 3次元座標を用いた機械学習.....	48
4.4.1 Excel ファイルに記録するデータ.....	49
4.4.2 奥行きを考慮した良いポーズ・悪いポーズの定義.....	50
4.4.3 機械学習の条件.....	52
4.4.4 機械学習の結果.....	52
4.5 奥行きを考慮したポーズ認識.....	53
第5章 結論（青井）.....	55
参考文献.....	56
謝辞.....	58

# 第1章 緒論 (青井)

## 1.1 研究背景

本節では、関節角度による運動姿勢の評価を目指した研究背景について説明する。

### 1.1.1 近年の日本における健康意識の高まり

近年の日本は新型コロナウイルスにより、健康意識が高まっている。感染症法が定める類型は、危険度や感染力、社会的影響に応じて1類から5類までに分類されている。新型コロナウイルスは当初、結核と同等の2類だったが、現在はインフルエンザと同等の5類に移行された。

九州産業大学がおこなった「コロナ禍における大学生の健康意識を生活習慣、健康度の自己評価」の研究によると、コロナ禍最中に入学した学生の健康意識はコロナ禍以前に入学した学生と比べると高いことが分かった[1]。図1-1は2019年に入学したコロナ禍未経験の学生と2020年、2021年に入学したコロナ禍最中の学生を対象に、運動習慣についてのアンケートをグラフにしたものである[1]。コロナ禍最中の2021年はコロナ禍未経験の2019年に比べ男女共に運動習慣がある学生が増加していることがわかる。また、運動習慣があることで精神的な健康度の悪化の予防に有意に関連していることも示されており、今後も運動習慣のある人の増加が考えられる。

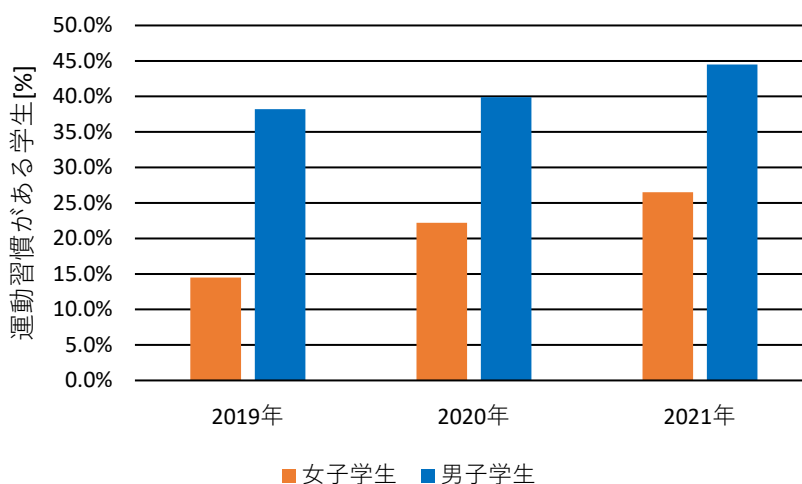


図1-1 運動習慣のある学生の割合[1]

### 1.1.2 フィットネスクラブ会員数の増加

運動習慣のある人の増加で、フィットネスクラブの利用者数も増加傾向にある。図 1- 2 は経済産業省が 2024 年 12 月に発表した、フィットネスクラブの会員数の調査結果[2]である。新型コロナウイルスが流行した 2020 年に利用者数は大きく減少したものの、2023 年現在は増加傾向にある。

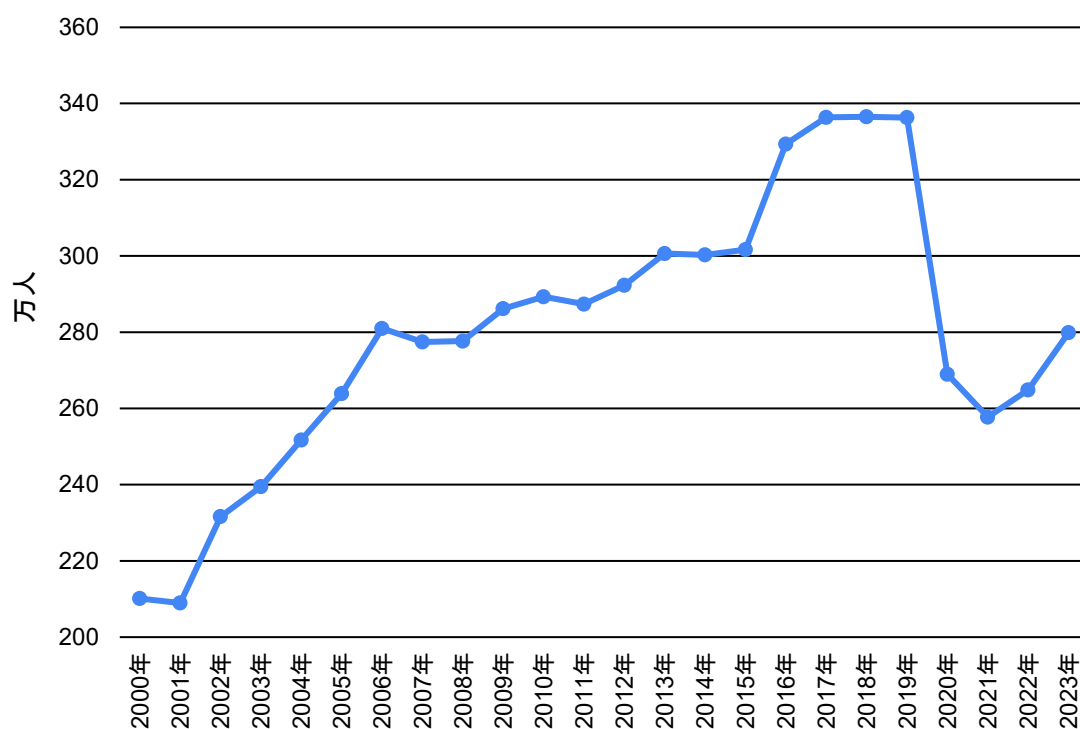


図 1-2 フィットネスクラブの利用者数の推移[2]

### 1.1.3 ボディビル人口の増加

1.1.2 項で説明したフィットネスクラブの利用者数の増加により、ボディビルの人口も増加傾向にある。図 1- 3 は日本のボディビル団体である JBBF の登録選手数の推移である [3][4][5]。コロナ禍の 2020 年度には登録者数が大きく減少したが、コロナ禍前の 2017 年度と 2023 年度を比較すると登録者数が 2000 人以上増加したことがわかる。

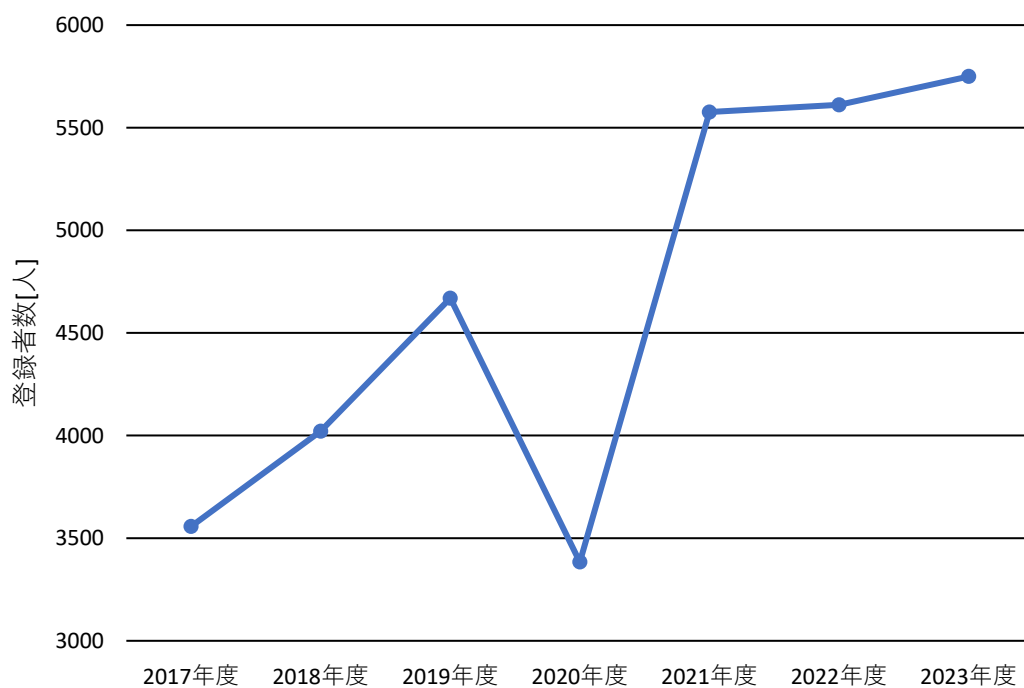


図 1- 3 JBBF の登録者の推移[3][4][5]

#### 1.1.4 ボディビルにおけるポーズの重要性

ボディビルのポーズには特定の筋肉を強調させる役割がある。そのため、筋肉を強調した効果的なポーズをとる必要がある。

図 1-4 は腕を最大限に強調できているポーズである。それに対し、図 1-5 は肘が肩より上がりすぎているため腕を最大限に強調することができていないと判断される[6]。しかし、競技者が自分で客観的にポーズを評価することは難しい。

そこで、ボディビルのポーズ時の姿勢を評価できるシステムの構築を目指すこととした。

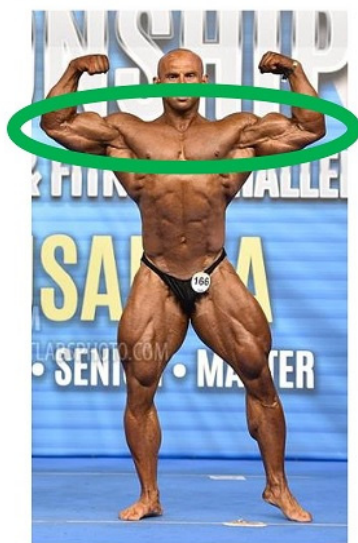


図 1-4 腕を最大限に強調したポーズ[6]

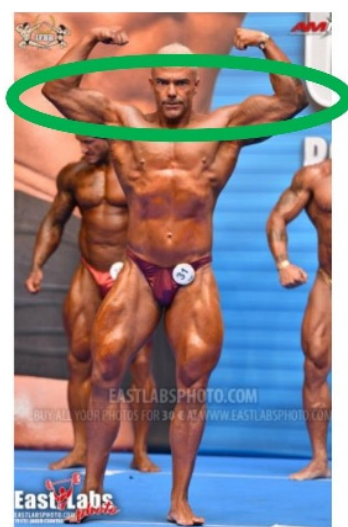


図 1-5 腕を強調できていないポーズ[6]

## 1.2 先行研究

本節では、動作分析システムを使用した姿勢改善の先行研究を紹介する。

### 1.2.1 ダイビングスキルの定着を促す MediaPipe を使用した動作比較システムの開発

#### の開発

加藤らは、ダイビング技術の習得を支援するための動作分析システムを提案した。提案されたシステムは、人物の骨格推定が可能なライブラリである MediaPipe を用いて熟練者と未熟練者の姿勢を検出する。その後、熟練者と未熟練者の各関節角度の差異を計算し、表示する（図 1-6）[7]。

実験では、システムを使用したグループと従来の指導法を受けたグループのスキル定着度の採点を行った。また、システム使用前後でスキル定着度の採点を行った。

結果として、システム使用グループと不使用グループ間のスキル定着度の平均値に有意差が認められ、大きな効果が認められた[7]。また、システム使用前後の比較でもスキル定着度の平均値に有意差が認められた。以上より、システム使用による姿勢改善への効果が高いことが分かった[7]。

この研究では撮影データを PC に移して分析するため、動作比較のリアルタイム性が欠けている。また、熟練者と未熟練者の関節角度の差異は表示されるが結果の良し悪しが直感的にわからないという問題点がある。

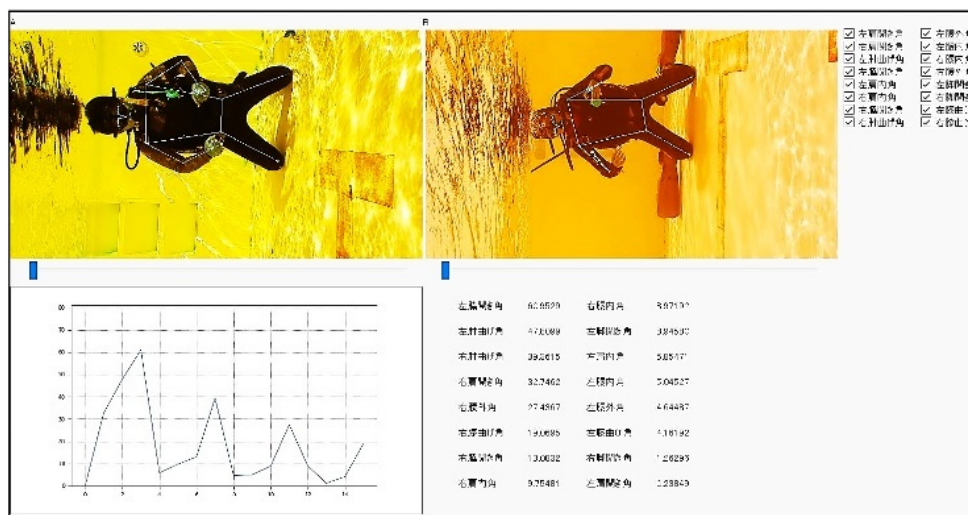


図 1-6 文献[7]が提案するシステムのユーザーインターフェース[7]



## 1.3 本研究の目的

1.1 節、1.2 節より、我々は以下の条件を満たす「関節角度による運動姿勢の評価システム」の開発を目的とした。

1. 関節角度からボディビルのポーズの良し悪しを評価すること
2. リアルタイムで評価を得ること
3. 評価結果が直感的にわかること

1の条件を満たすために、文献[7]と同じく Google 社が提供する機械学習ライブラリの MediaPipe を使用する。MediaPipe は、画像や動画、ストリーミングメディアに対して、顔や手の検出、姿勢推定が可能なライブラリである。

2の条件を満たすために、カメラから取得したデータを PC で直接分析する手法を確立する。

3の条件を満たすために、使用者が良いポーズと悪いポーズを直感的に理解できる表示方法を実現する。

## 第 2 章 ポーズ認識 (池田)

第 2 章ではポーズ認識の手法について説明する。

### 2.1 使用環境

本研究で開発したシステムの使用環境について説明する。

本システムでは、ノート PC 搭載の Web カメラを室内で使用することを想定している。

表 2-1 に使用 PC の仕様を示す。Windows 上でプログラミング言語 Python を用いて機械学習を行う環境を構築するためのディストリビューションである Anaconda をインストールした。さらに、Anaconda 上で開発環境 Spyder を利用し、MediaPipe と呼ばれる機械学習用ライブラリを用いて骨格点の認識を行う。

表 2-1 本研究で使用する PC の仕様

OS	Windows 11 Home
CPU	12th Gen Intel® Core™ i3-1215U
RAM	8.00 GB

## 2.1.1 MediaPipe について

骨格点の認識に用いる MediaPipe について説明する。

MediaPipe とは、画像や動画、ストリーミングメディアに対して、顔や手、骨格点の検出を可能とする機械学習用ライブラリである。MediaPipe には、多くの利用方法があるが、本研究では、姿勢(ポーズ)を認識する機能を用いる[8]。ポーズ認識により、図 2-1 及び表 2-2 に示した 33 箇所の骨格点が推定される。このポーズ認識を利用するため、Kazuhito00 氏が作成したサンプルファイルを利用する[9]。サンプルファイルを用いた認識結果を図 2-2 に示す。

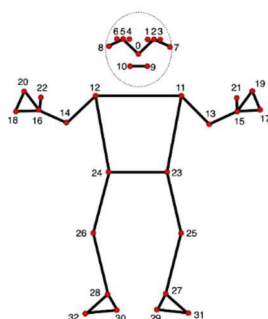


図 2-1 推定される骨格点の位置[8]

表 2-2 骨格点の名称[8]

0.	nose	17.	left_pinky
1.	left_eye_inner	18.	right_pinky
2.	left_eye	19.	left_index
3.	left_eye_outer	20.	right_index
4.	right_eye_inner	21.	left_thumb
5.	right_eye	22.	right_thumb
6.	right_eye_outer	23.	left_hip
7.	left_ear	24.	right_hip
8.	right_ear	25.	left_knee
9.	mouth_left	26.	right_knee
10.	mouth_right	27.	left_ankle
11.	left_shoulder	28.	right_ankle
12.	right_shoulder	29.	left_heel
13.	left_elbow	30.	right_heel
14.	right_elbow	31.	left_foot_index
15.	left_wrist	32.	right_foot_index
16.	right_wrist		



図 2-2 実際に 33 箇所の骨格点が推定された様子

## 2.2 2次元の関節角度算出手法

本節では2次元の関節角度算出手法について説明する。

### 2.2.1 ピクセルについて

図 2-3 は、33箇所 の骨格点のピクセルの位置を、その信頼性ととも に Excel に保存した様子を示している。MediaPipe で検出される各骨格点の座標は全て 2次元のピクセルの位置で表される。また、図 2-4 のように、ピクセルとは左上を原点[0,0]とした要素である。MediaPipe で用いられる座標はこのピクセルの位置を示しており、全て整数で表記される。

	x	y	信頼性
1	165	78	1
2	169	69	1
3	173	69	1
4	176	69	1
5	159	69	1
6	155	70	1
7	151	71	1
8	180	74	1
9	143	77	1
10	172	89	1
11	160	89	1
12	217	127	1
13	115	133	1
14	286	110	1
15	40	122	1
16	270	46	0.97
17	51	56	0.99
18	264	31	0.86
19	48	45	0.94
20	255	32	0.83
21	57	42	0.93
22	257	39	0.83
23	60	47	0.93
24	209	273	1
25	149	280	1
26	249	385	0.95
27	133	390	0.97
28	267	489	0.94
29	147	493	0.96
30	259	504	0.72
31	156	506	0.83
32	290	523	0.9
33	124	521	0.92

図 2-3 MediaPipe で検出される骨格点の座標

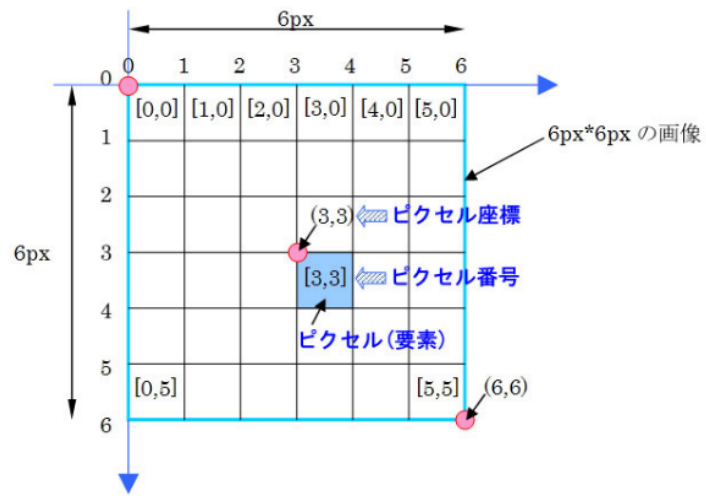


図 2-4 ピクセルについて[10]

## 2.2.2 関節角度算出手法

2つのベクトルのなす角の公式を利用することで関節角度を算出する。例を以下に示す。

2つの2次元ベクトルをそれぞれ  $\vec{a} = (x_1, y_1)$ ,  $\vec{b} = (x_2, y_2)$  とする。2つのベクトルのなす角の公式より、ベクトル  $\vec{a}$  と  $\vec{b}$  のなす角  $\theta$  は次式を満たす。

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$
$$\cos \theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}}$$

よって、求める角度は以下のように計算される。

$$\theta = \cos^{-1} \left( \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \right) [\text{rad}]$$
$$\theta = \cos^{-1} \left( \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}} \right) [\text{rad}]$$

最後に、単位を変換して、次式を得る。

$$\theta = \frac{180}{\pi} \cos^{-1} \left( \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \right) [^\circ]$$
$$\theta = \frac{180}{\pi} \cos^{-1} \left( \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}} \right) [^\circ]$$

これらの式を用いて、右肩・右肘・右手首の骨格点の座標から得られるベクトルを利用し、右肘の関節角度を求め、画面に赤字で表示させた結果を図 2-5 に示す。

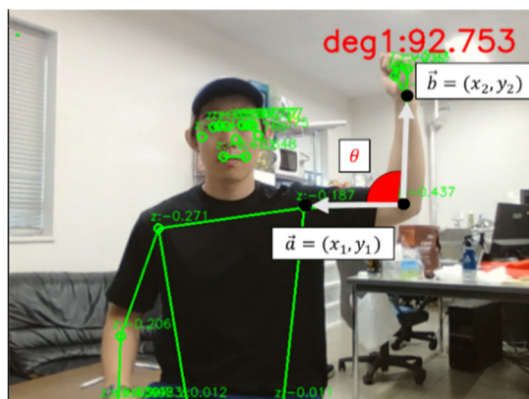


図 2-5 2次元で算出された右肘の関節角度

## 2.3 2次元の関節角度算出手法と平面角の変化に関する実験

2.2節で解説した2次元の関節角度算出手法は、図2-5に示されているように、腕が作る平面とカメラの向きが垂直であることを前提としている。そこで、この平面とカメラの向きのなす角度が変化したときに、どれだけ正しい関節角度が計算されるかを調べる実験を行う。ここで、腕が作る平面とカメラの向きがなす角度を平面角と呼び、図2-6のように垂直のときに $0^\circ$ となるように定義する。

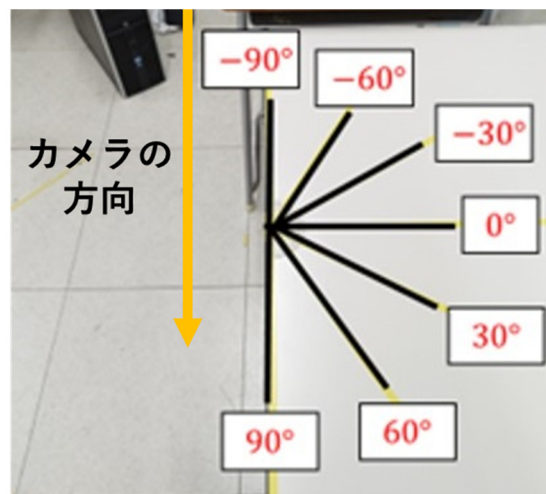


図2-6 腕が作る平面とカメラの方向の間の平面角の定義

### 2.3.1 実験方法

右肘の関節角度を概ね $90^\circ$ に固定した状態で正面のWebカメラに対して、平面角(図2-6)を $-90^\circ$ から $+30^\circ$ 刻み( $-90, -60, -30, 0, 30, 60, 90[^\circ]$ )で変化させ、算出された関節角度をグラフ化する。



### 2.3.2 実験結果

実験結果を図 2-7 に示す。

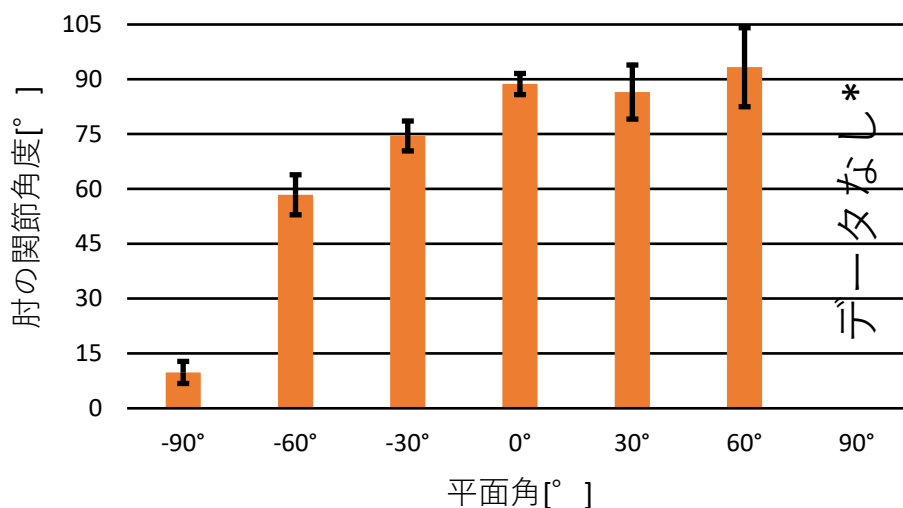


図 2-7 実験結果

実験結果より、平面角が 0° 付近のとき正しい角度である 90° が取得可能であることがわかる。一方で、骨格点の奥行きの違いが大きくなるにつれて、正しい角度の取得が不可能となった。これは、通常の Web カメラでは、奥行き(z座標)を取得できないためであると考えられる。また、90° の平面角では、骨格点の推定自体がうまくいかないため、データの取得が不可能であった。

これらのことから、カメラの向きに対し、身体が正面を向くような条件で本システムを利用することにした。

## 2.4 推定対象のポーズ

本節では、推定対象としたボディビルのポーズについて説明する。

本研究では、図 2-8 の「ダブルバイセップス」を推定対象のポーズとする。



図 2-8 ダブルバイセップス[6]

## 2.4.1 評価ポイント

一般的にダブルバイセップスの評価ポイントとして、

- 肘の角度
- 肘の位置
- 全身のバランス
- 大腿四頭筋

といった4点が挙げられる[11]。これらを MediaPipe による認識結果(図 2- 9)をもとに評価することが以下の目標である。

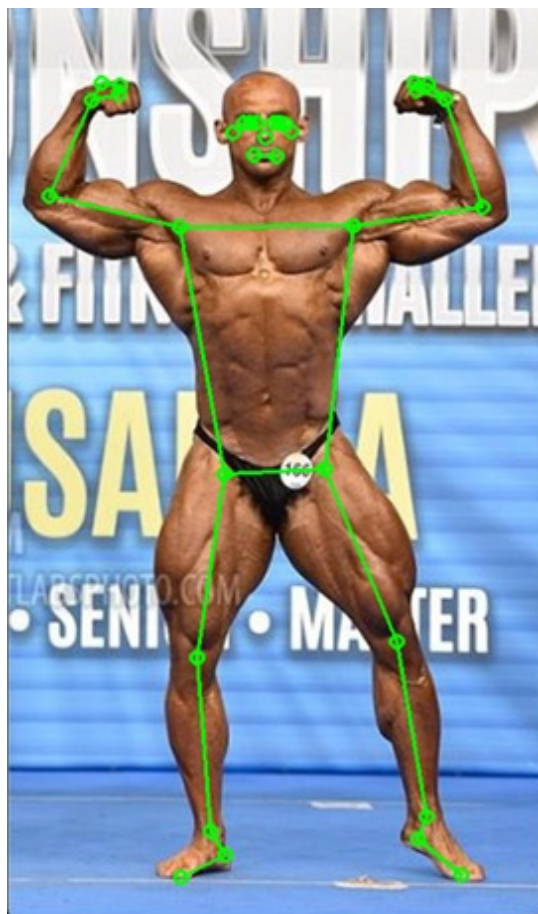


図 2- 9 MediaPipe での認識

## 2.5 ポーズの判定に用いる角度

本節ではポーズの判定に用いる角度について説明する。

2.4.1 項で解説した評価ポイントを基準に、用いる角度を表 2-3 の 8 箇所 に定義した。また、これらの角度を図 2- 10 に図示する。ここでは、 $\theta_5, \theta_6$  を算出するために必要な肩と腰のそれぞれ左右の中心、及びそれらを結んだ身体の中心線を MediaPipe に追加で定義している。

表 2-3 ポーズの判定に用いる角度

推定対象の箇所	角度
左肘	$\theta_1$
右肘	$\theta_2$
左脇	$\theta_3$
右脇	$\theta_4$
身体の中心線に対しての肩の傾き	$\theta_5$
身体の中心線の傾き	$\theta_6$
左膝	$\theta_7$
右膝	$\theta_8$

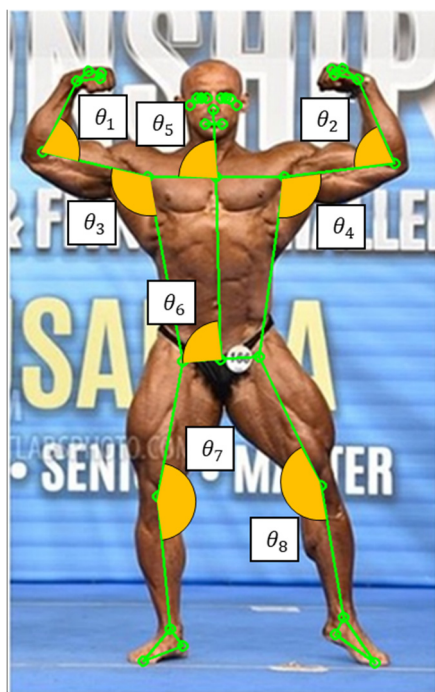


図 2- 10 ポーズの判定に用いる角度

## 2.6 2次元座標を用いた機械学習

本節では、本研究で用いる2次元座標を考慮した機械学習の手法について説明する。

### 2.6.1 機械学習の概要

機械学習とは、ある目的を達成するための手順を予め用意されたデータに基づき自動的に決定する技術である[12]。本研究で機械学習を用いる理由として、ポーズを評価するために適切な関節角度を基準とする際、それに関する学术论文が少ないことでデータが不足していることが挙げられる。そこで、機械学習により、プロのポーズから2.5節で定義した8箇所関節角度を算出し、学習することでこの問題の解決を図る。

## 2.6.2 機械学習の流れ

機械学習の全体の流れを以下の手順で行う。

- I. 良いポーズ・悪いポーズと定義したそれぞれの静止画に対して、骨格点 33 箇所の座標を Excel ファイル (csv(Comma Separated Values)形式) に記録する(図 2- 3)
- II. I. で記録したポーズの Excel ファイルを 2 種類(良・悪)のフォルダに分類する
- III. II. で分類されたポーズ(座標)から 2.5 節で定義した角度を計算し、良いポーズか悪いポーズかの判定を機械学習する。

また、学習のイメージ図を図 2- 11 に示す。以後、class0 とは良いポーズ、class1 とは悪いポーズのことであるとする。

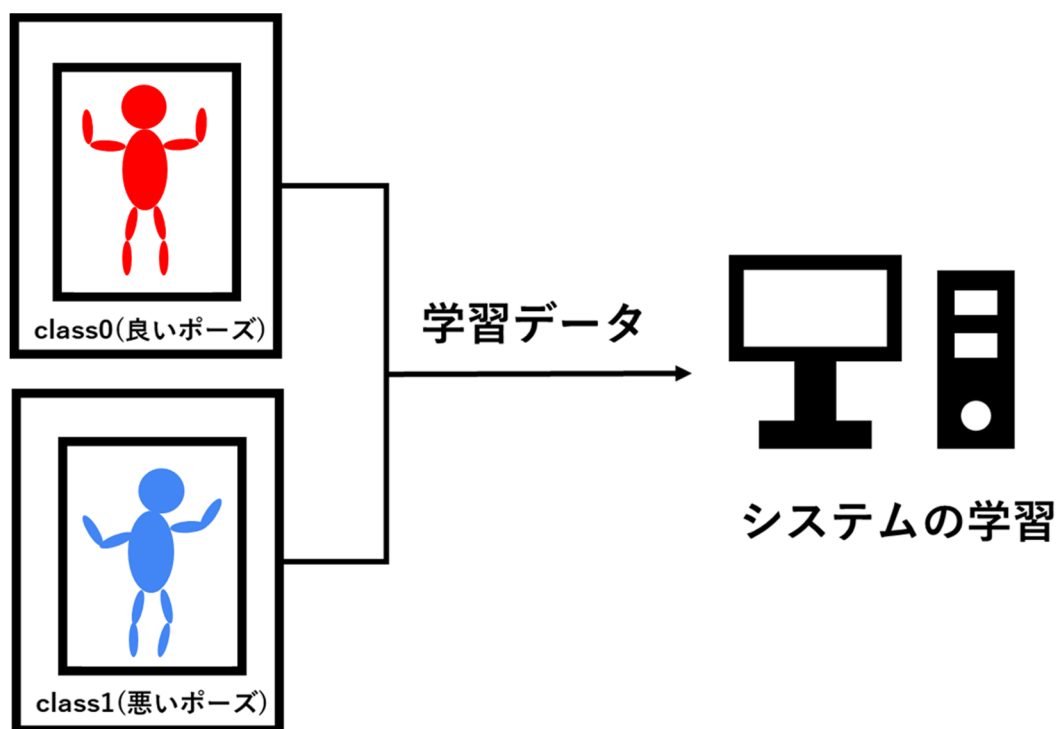


図 2- 11 学習のイメージ図

### 2.6.3 座標から角度への移行手法

機械学習に用いるのは、骨格点 33 箇所の 2 次元座標から算出された 8 箇所の関節角度である。本研究において、学習対象に座標ではなく角度を用いる理由は、静止画内の人物の立ち位置や体格によって座標が異なることでポーズを適切に分類できないことが想定されるためである。

ここでは、機械学習させるベクトルデータについて解説する。

最初に、特徴点の個数を骨格点 33 箇所の 2 次元座標 66 個から 8 個( $\theta_1 \sim \theta_8$ )のポーズの判定に用いる角度に変更する。

$$\mathbf{X} = [[x_1, y_1], [x_2, y_2], \dots, [x_{33}, y_{33}]]$$

↓

$$\mathbf{X} = [\theta_1, \theta_2, \dots, \theta_8]$$

8 個の角度を学習データの個数  $i$  だけ学習させるため、学習データは以下のように  $8 \times i$  個のデータとなる。

$$\mathbf{X} = [\theta_1, \theta_2, \dots, \theta_8]$$

↓

$$\mathbf{X} = \begin{bmatrix} [\theta_{1(1)}, \theta_{2(1)}, \dots, \theta_{8(1)}] \\ [\theta_{1(2)}, \theta_{2(2)}, \dots, \theta_{8(2)}] \\ \vdots \\ [\theta_{1(i)}, \theta_{2(i)}, \dots, \theta_{8(i)}] \end{bmatrix}$$

それに、対応する正解(ターゲット)  $\mathbf{y}$  は、以下ようになる。

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(i)} \end{bmatrix}$$

#### 2.6.4 良いポーズ・悪いポーズの定義

プロによるダブルバイセップスのポーズの静止画から8箇所の関節角度を求め、良いポーズの基準とする。また、ダブルバイセップスの評価ポイントを満たしていない悪いポーズは我々が撮影し用意するものとする。

良いポーズの具体例を図 2- 12 に、悪いポーズの具体例を図 2- 13、図 2- 14、図 2- 15、図 2- 16 にそれぞれ示す。また、表 2- 4 に 2.4.1 項の評価ポイントを基準にこれらを纏めた。

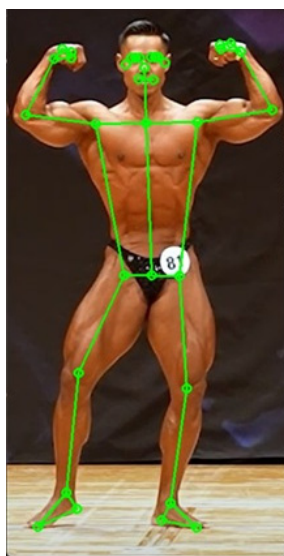


図 2- 12 良いポーズ



図 2- 13 悪いポーズ 1



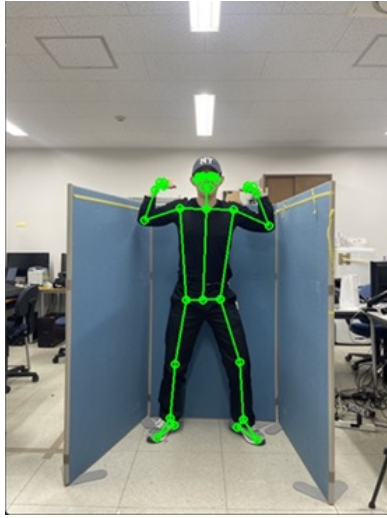


図 2-14 悪いポーズ 2

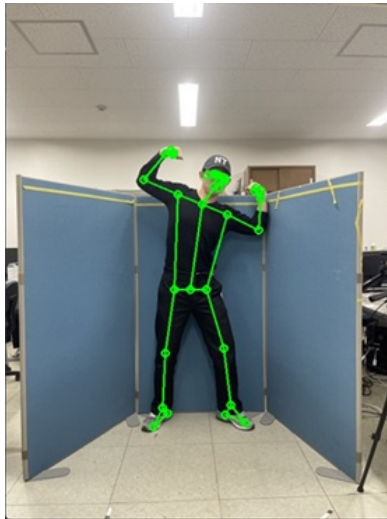


図 2-15 悪いポーズ 3

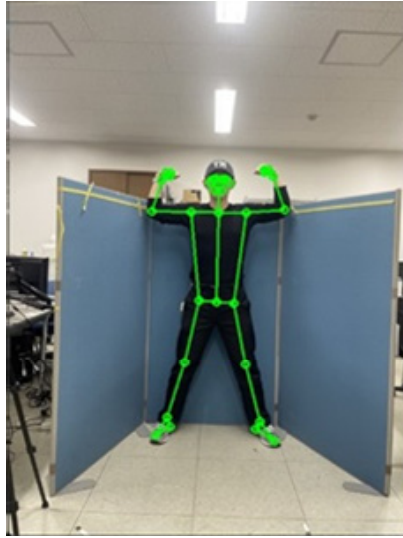


図 2-16 悪いポーズ 4

表 2-4 各ポーズの評価

	肘の角度	肘の位置	全身のバランス	大腿四頭筋
図 2-12	○	○	○	○
図 2-13	×	○	○	○
図 2-14	×	×	○	○
図 2-15	○	×	×	○
図 2-16	○	○	○	×

## 2.6.5 機械学習の条件

良いポーズ 100 枚、悪いポーズ 200 枚の合計 300 枚の静止画を用意し機械学習を行う。学習条件を表 2-5 に示す。

機械学習では、エポック数(epoch)と呼ばれる学習回数を表す数値を設定する。ここでは、1000 として設定した。また、バッチサイズ(batch\_size)とは 1 epoch 実行中にパラメータ調整をどのタイミングで行うかについて決定する値である。ここでは、50 として設定した。このとき、Excel ファイルの個数との商( $300 \div 50$ )より、6 といった数値が求まる。したがって、1 epoch 実行中に 6 回のパラメータ調整が行われることとなる。この数値はイテレーション(iteration)と呼ばれる。また、1 iteration におけるデータは、300 個の中から 50 個のデータを選ぶ組み合わせからランダムに選ばれる。

$$300C_{50}$$

最後に、epoch と iteration の積( $1000 \times 6$ )より、全体では 6000 回の学習(パラメータ調整)が行われることとなる。このように、学習するデータにランダム性をもたせることで、柔軟性のある学習データを作成可能である。

表 2-5 学習条件

エポック数	1000
バッチサイズ	50
イテレーション	6

## 2.6.6 機械学習の結果

2.6.5 項で解説した学習条件での学習結果を以下に示す。

2.6.4 項で定義した良いポーズ・悪いポーズの分類における正解率は99.67%であった。正解率とは、分類問題において正しく分類したかどうかのみで考慮される。全データをまとめた配列  $\mathbf{X}$ 、それに対する正解  $\mathbf{y}$ 、システムの実出力  $\text{result}$  に対し、正解率はプログラム上で次の式で算出される。

```
total = len(X)
```

```
success = sum(result==y)
```

```
correct rate=(100.0*success/total)
```

また、学習結果の損失関数の時間変化を図 2-17 に示す。損失関数とは、システムの実出力と正解との間の距離に類似した値をとる関数である。0 以上の値をとり、システム出力と正解が一致するときに 0 となる性質をもつ。図 2-17 をみると、学習の進行とともに 0 に近づいているため、システムの学習が成功していることがわかる。

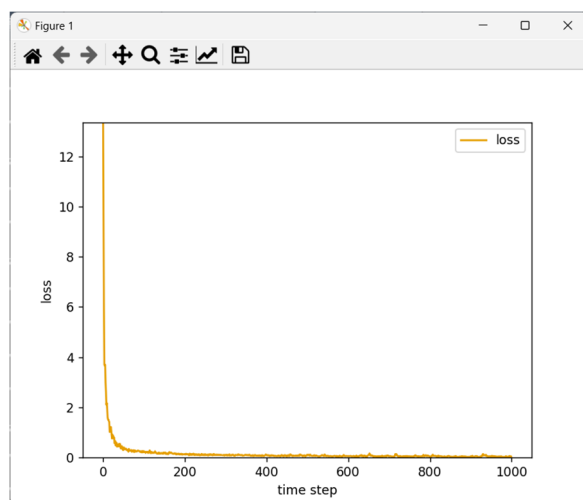


図 2-17 損失関数

## 2.7 ポーズ認識

本節では、前節までの内容を踏まえたポーズ認識の手法について説明する。

2.6.6 項で学習させた結果を用いてポーズ認識を行う。システムによる分類結果を画像に上書きしたものを図 2-18、図 2-19、図 2-20、図 2-21、図 2-22 に示す。全ての図において、class0 と class1 によるポーズの分類が正しくできていることがわかる。

また、学習データに含まれないポーズに対する分類結果を図 2-23 に示す。良いポーズ想定しているが、正しく class0 と分類されていることがわかる。

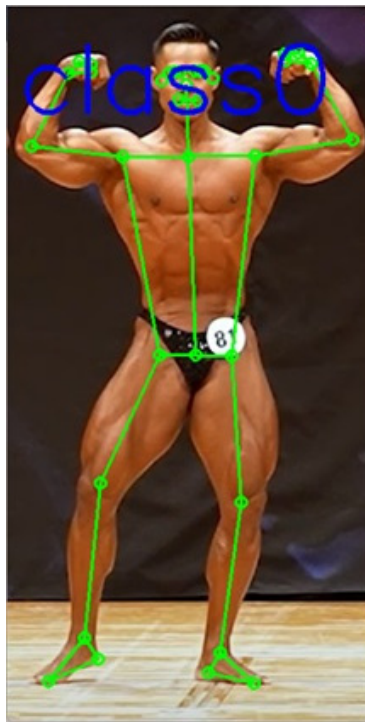


図 2-18 正しく分類された良いポーズ

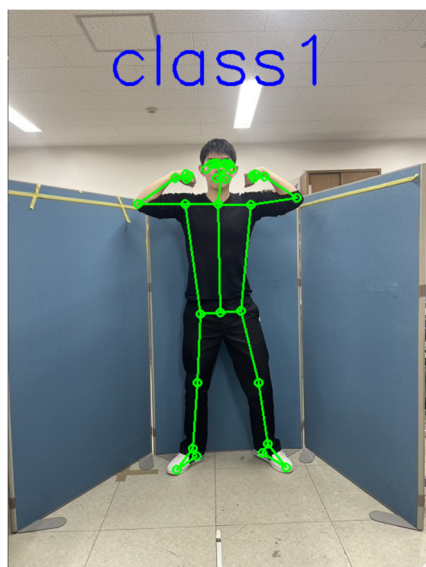


図 2-19 正しく分類された悪いポーズ 1

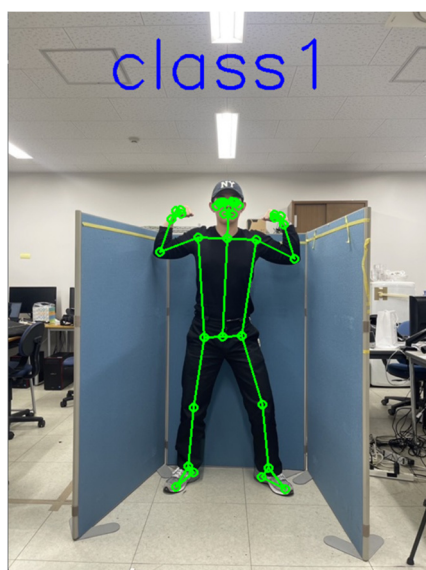


図 2-20 正しく分類された悪いポーズ 2

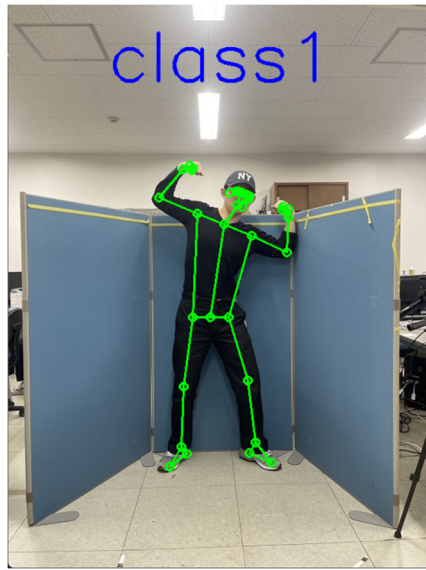


図 2-21 正しく分類された悪いポーズ 3



図 2-22 正しく分類された悪いポーズ 4

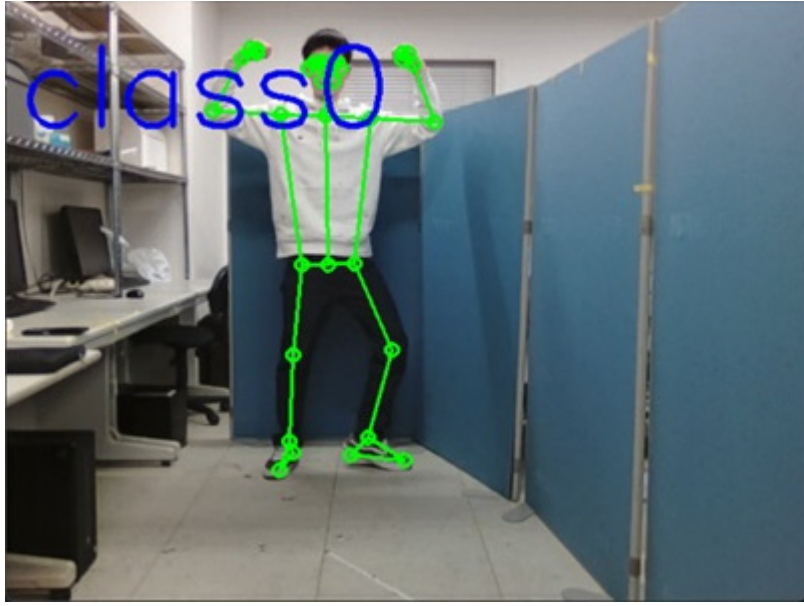


図 2-23 学習データに含まれないポーズに対する分類結果



## 第3章 関節角度による運動姿勢の評価システム精度実験 (青井)

### 3.1 実験1

本実験では、第2章で作成した関節角度による運動姿勢の評価システムの精度を確認する。

#### 3.1.1 実験方法

ポーズを分類したい画像を学習データに読み込ませ、ダブルバイセップスのポーズとして良いか悪いのか分類を行う。実験のイメージ図を図3-1に示す。

##### I. ポーズを分類したい画像

学習済みのシステムを評価するために学習データとは別の画像を class0、class1 に50枚ずつ使用する。

##### II. 学習済みのシステム

第2章で学習済みのシステムを用いる。

##### III. 分類結果

良いポーズであれば0、悪いポーズであれば1と画像上に表示される。画像の例を図3-2、図3-3に示す。

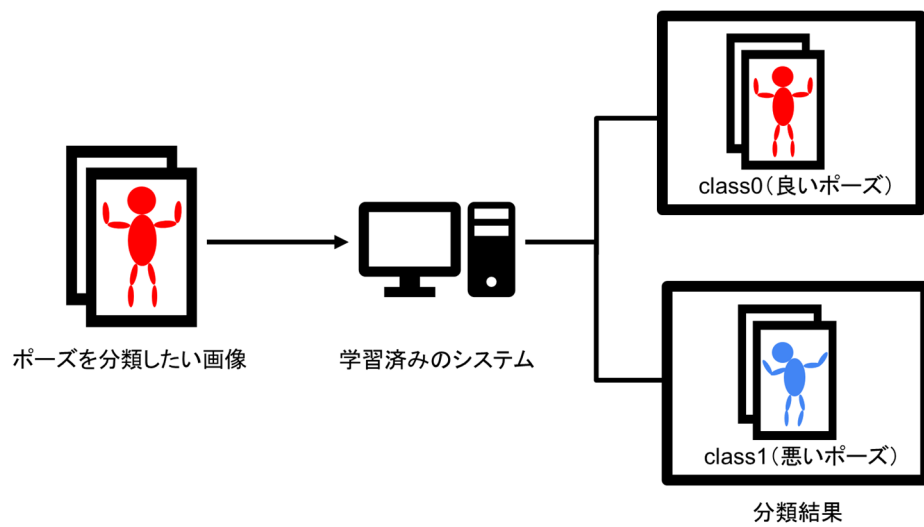


図 3-1 実験のイメージ図

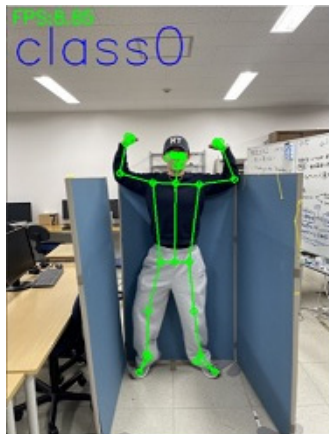


図 3-2 良いポーズの分類結果



図 3-3 悪いポーズの分類結果

### 3.1.2 実験結果 1

実験結果を図 3- 4 と表 3- 1 に示す。良いポーズの正解率は 100% となったが、悪いポーズの正解率が 74% と低く、良いポーズと悪いポーズの全体の正解率は 87% という結果になった。

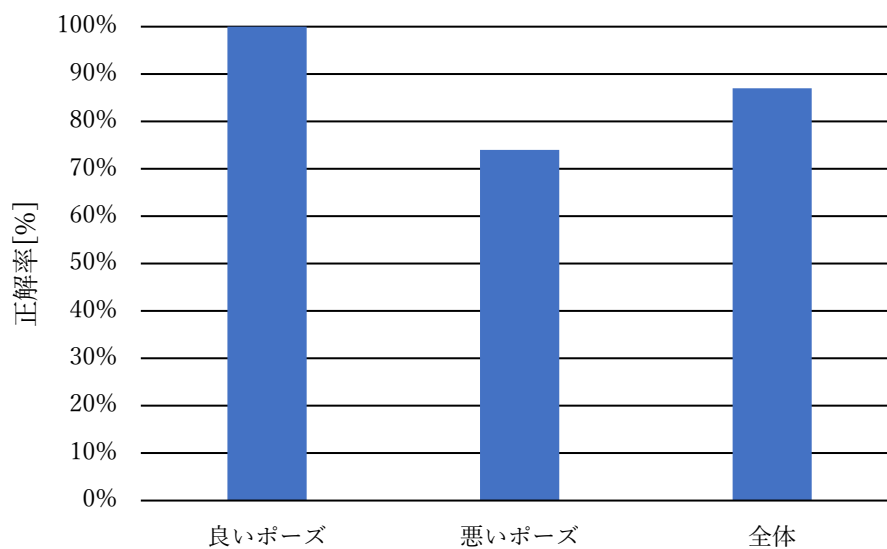


図 3- 4 実験結果 1

表 3- 1 分類結果 1

		予測	
		良いポーズ	悪いポーズ
結果	良いポーズ	50	13
	悪いポーズ	0	37

### 3.1.3 考察

全体の正解率が87%で精度が高いと言えない結果であった。

典型的には図3-5、図3-6のように悪いポーズが良いポーズと誤認識される問題があった。

良いポーズの判定画像が悪いポーズと分類（以下、偽陽性と呼称）されたり、悪いポーズの判定画像が良いと分類（以下、偽陰性と呼称）されたりする誤判定の結果は、学習データに問題があると考えられる。例えば、学習データの偏りやデータ数が少ないことなどの問題である。

実験1で使用した学習データの画像と偽陰性と判定されたポーズを照らし合わせたところ、偽陰性のポーズに似た学習データの画像が少なかったことが分かった。このことが原因で、悪いポーズの特徴量が少なく、実験1の結果になったと考えられる。



図3-5 良いポーズと分類された悪いポーズ1（偽陰性）



図3-6 良いポーズと分類された悪いポーズ2（偽陰性）

## 3.2 実験2：学習データの改善

実験1の結果を踏まえ、学習データの改善を行う。実験1で使用した悪いポーズの学習データ200枚に30枚の画像を追加し、新たな学習データとする。追加する30枚の画像は、偽陰性と判定されたポーズに近いポーズをした画像を使用した。

### 3.2.1 実験方法

新しい学習データを用いて第2章の方法で学習し、実験1と同様の流れで実験を行う。

### 3.2.2 実験結果 2

実験結果を図 3-7 と表 3-2 に示す。良いポーズの正解率は 92% となり、実験 1 の結果より低下した。しかし、悪いポーズの正解率が 90% となり実験 1 と比較すると正解率が改善した。また、全体の正解率も実験 1 と比較すると高くなった。実験 1 と実験 2 の全体の正解率を比較した図を図 3-8 に示す。

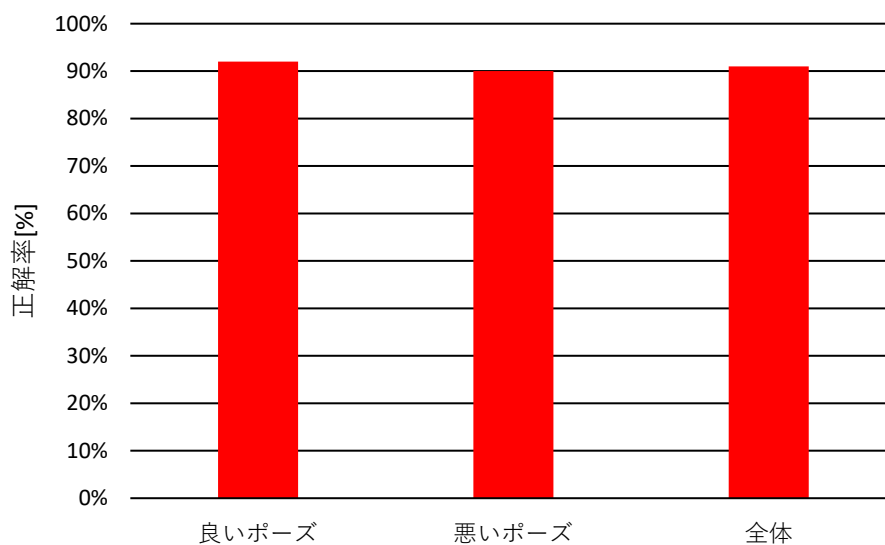


図 3-7 実験結果 2

表 3-2 分類結果 2

		予測	
		良いポーズ	悪いポーズ
結果	良いポーズ	46	5
	悪いポーズ	4	45

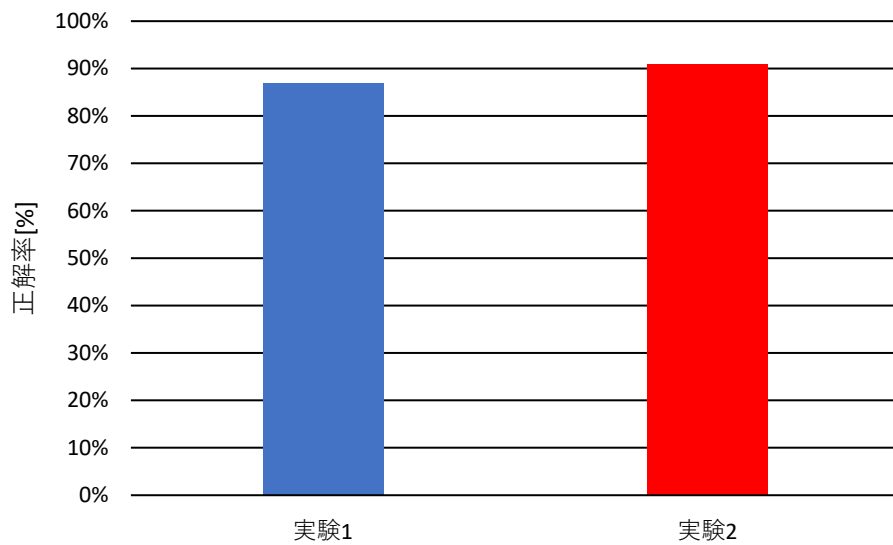


図 3-8 全体の比較

### 3.2.3 考察

学習データを改善したことで、実験1より全体の正解率を改善することができた。全体の正解率は90%となり、本研究の目的である関節角度から姿勢を評価することができている。

実用化に向けては、悪い姿勢のポーズの学習データの増強が必要であると考えている。

さらに、本実験では限定したポーズや条件下で実験を行ったため、本システムに対応しないポーズを改善することを目指す必要がある。

### 3.3 本システムに対応しないポーズ

本システムはカメラに対して奥行きのあるポーズは、正しい関節角度を算出することができないため、対応していない。実際に対応していない画像を図 3-9 に示す。

本システムで使用しているカメラは Web カメラであり、2 次元の情報のみしか取得することができないためである。

奥行きを取得するためには、例えば 3 次元の情報を取得可能なカメラの RealSense を使用する必要がある。RealSense を利用したシステムについては次章で説明する。

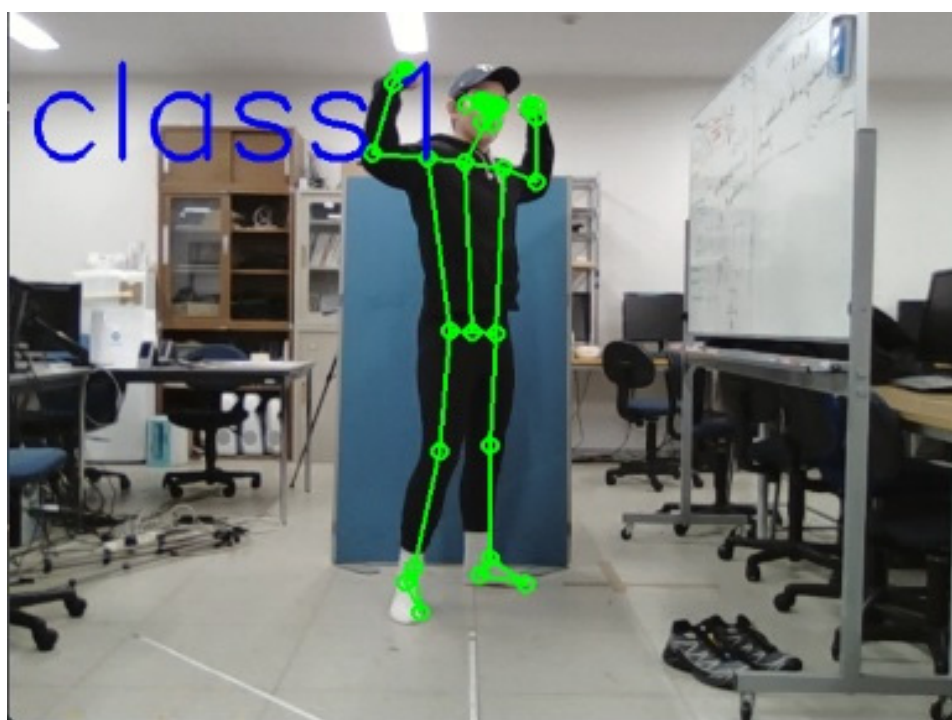


図 3-9 本システムに対応しないポーズ



## 第 4 章 RealSense (池田)

第 4 章では RealSense を用いたポーズ認識の手法について説明する。

### 4.1 RealSense について

本節では、RealSense について説明する。

RealSense とは、インテル社が開発するデプスカメラであり、Web カメラなどの従来のカメラでは認識不可能な奥行きを把握できる製品である。本研究では、RealSense D435(図 4-1)を用いて 3 次元でのポーズ認識を行う。使用する RealSense の仕様を表 4-1 に示す。また、RealSense では図 4-2 のような RGB 画像と、図 4-3 のような深度センサから得られる Depth 画像を利用可能である。本研究では、RealSense と表 2-1 で示した使用 PC を USB 接続して使用する。また、Web カメラ(2 次元)の認識で用いた MediaPipe を同様にポーズ認識用のライブラリとして使用する。



図 4-1 RealSense D435

表 4-1 本研究で使用する RealSense の仕様

製品名	Intel RealSense Depth Camera D435
デプス解像度	1280 × 720
RGB 解像度	1920 × 1080
最大デプス FPS	90
デプス範囲	0.2[m]~10[m]
デプス対角画角	90[° ]
寸法	90[mm] × 25[mm] × 25[mm]



图 4-2 RGB 画像

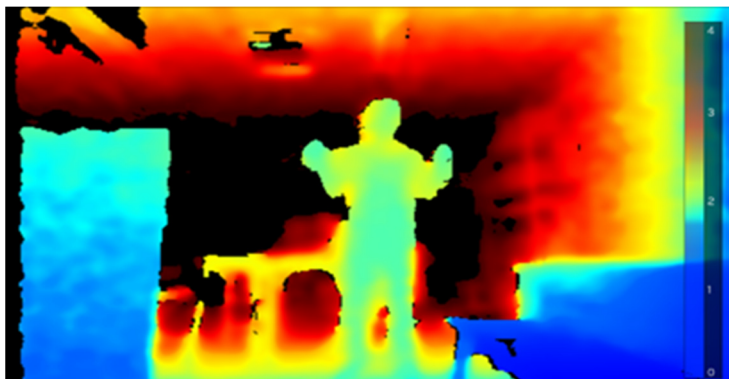


图 4-3 Depth 画像

## 4.2 3次元の関節角度算出手法

本節では RealSense を用いた 3次元の関節角度算出手法について説明する。

### 4.2.1 スケールの統一

RealSense から取得される数値は、ピクセル( $[x, y]$ )表記の座標とメートル( $z[m]$ )表記の距離であり、この状態ではスケールが統一されていない。ここで、正しい関節角度を取得するために、ピクセルをメートルに変換し、スケールを統一する必要がある。実際には、「pyrealsense2」と呼ばれる Python モジュールに含まれる「rs2\_deproject\_pixel\_to\_point()」の関数を用いた。この関数は次のように実行される[13]。

```
rs.rs2_deproject_pixel_to_point(depth_intrin, [x, y], z)
```

また、「depth\_intrin」は深度情報の組み込み関数を指している。これにより、RealSense から得られる数値のスケールは以下のようにメートルに統一される。

$(x, y, z)[m]$

ピクセルからメートルへ変換した後の 2次元座標の原点と向きを図 4-4 に、z軸の定義と向きを図 4-5 にそれぞれ図示する。

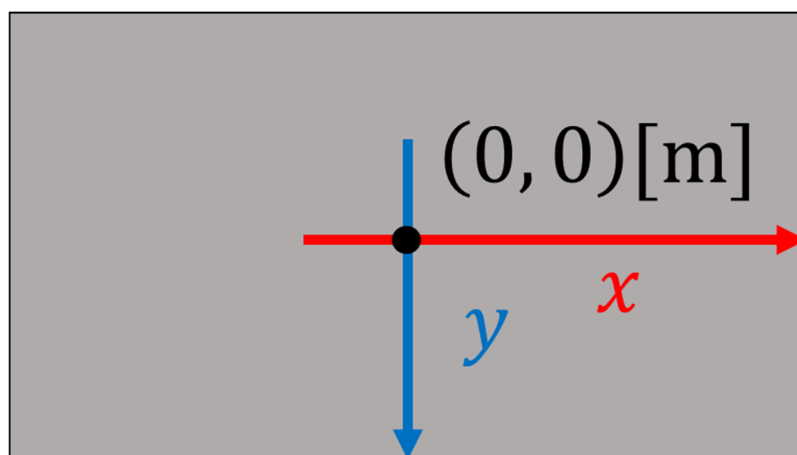


図 4-4 メートル変換後の 2次元座標の原点と向き

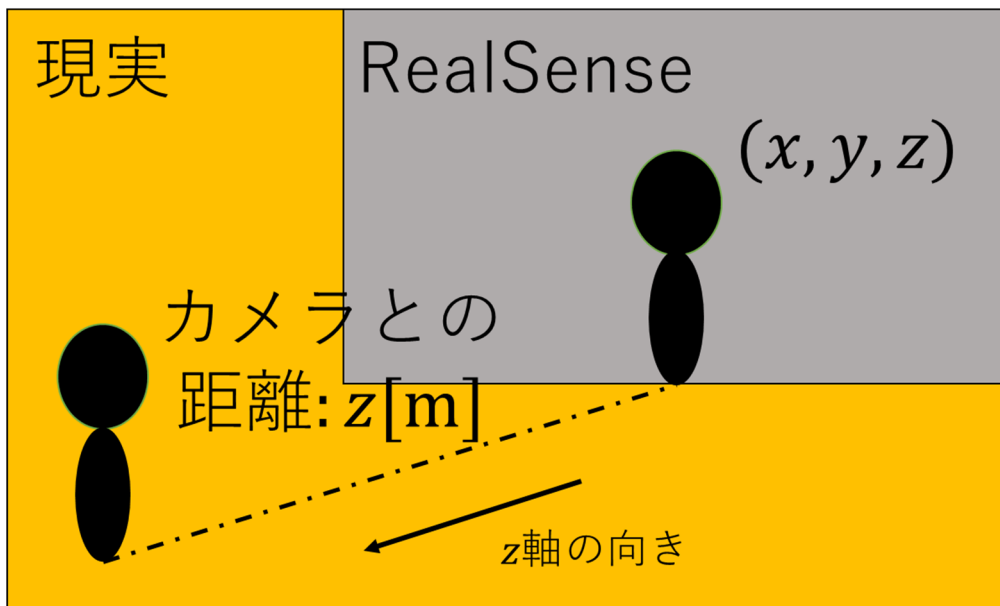


図 4-5  $z$  軸の定義と向き

## 4.2.2 関節角度算出手法

3次元における関節角度算出手法を説明する。また、ここでの座標は4.2.1項で解説した通り、スケールをメートルに統一したものとする。Webカメラ(2次元)の認識と同様に2つのベクトルのなす角の公式を利用することで関節角度を算出する。

2つの3次元ベクトルをそれぞれ  $\vec{a} = (x_1, y_1, z_1)$ ,  $\vec{b} = (x_2, y_2, z_2)$  とする。2つのベクトルのなす角の公式より、ベクトル  $\vec{a}$  と  $\vec{b}$  のなす角  $\theta$  は次式を満たす。

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$
$$\cos \theta = \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}}$$

よって、求める角度は以下のように計算される。

$$\theta = \cos^{-1} \left( \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \right) [\text{rad}]$$
$$\theta = \cos^{-1} \left( \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}} \right) [\text{rad}]$$

最後に、単位を変換して、次式を得る。

$$\theta = \frac{180}{\pi} \cos^{-1} \left( \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \right) [^\circ]$$
$$\theta = \frac{180}{\pi} \cos^{-1} \left( \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \sqrt{x_2^2 + y_2^2 + z_2^2}} \right) [^\circ]$$

これらの式を用いて、右肩・右肘・右手首の骨格点の座標から得られるベクトルを利用し、右肘の関節角度を求め、画面に赤字で表示させた結果を図4-6に示す。

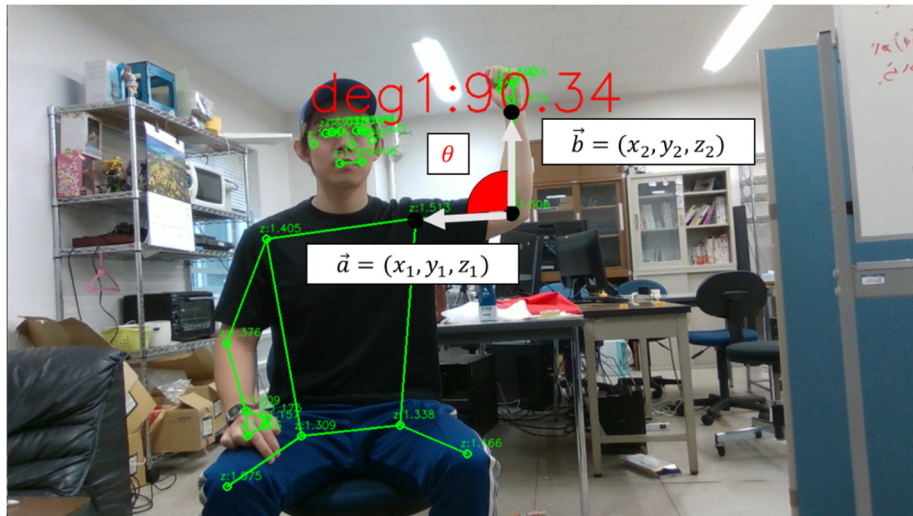


図 4-6 3次元で算出された右肘の関節角度

### 4.3 3次元の関節角度算出手法と平面角の変化に関する実験

#### 4.3.1 実験方法

2.3節で行った平面角の変化に関する実験を RealSense に対しても行う。実験方法は全て 2.3節と同様とする。

### 4.3.2 実験結果

実験結果を図 4-7 に示す。

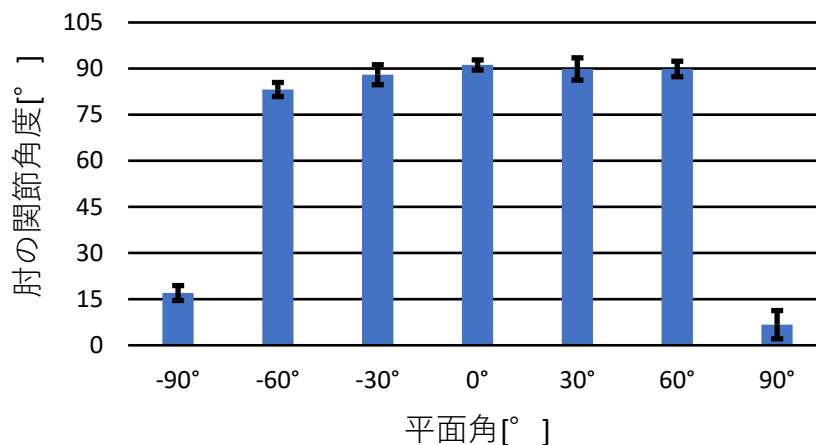


図 4-7 実験結果

実験結果より、腕が作る平面とカメラの向きがなす角(平面角)を変化させても、肘の関節角度は広い範囲で 90° に近い値を求められていることがわかる。ただし、平面角が -90° 及び 90° のときは正しい関節角度を求められなかった。2.3.2 項の Web カメラ(2次元)での実験結果(図 2-7)と比較すると、正しい関節角度を取得可能な平面角が増加していることがわかる。したがって、Web カメラを用いたときよりも信頼性が向上したといえる。一方で、-90, 90[°] のとき正しい角度を求められていない。この理由として、RealSense から見て対象の骨格点が重なり、距離  $z[m]$  が正しく認識できないことが挙げられる。実際に、平面角が -90° で起きたケースを図示したのが図 4-8 である。

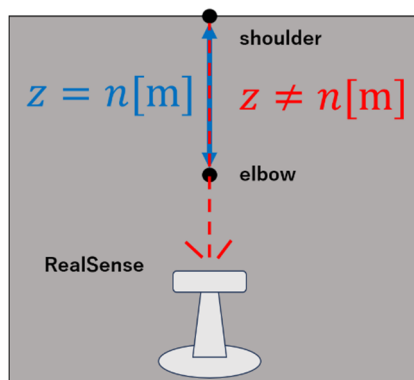


図 4-8 距離が正しく認識できないケース

## 4.4 3次元座標を用いた機械学習

本節では、3次元における機械学習の手法を説明する。

全体の流れとしては、2.6.2項で解説したWebカメラ(2次元)での機械学習と概ね同様である。機械学習の対象を3次元に移行することで、奥行き(z座標)が取得可能となり、図3-9のように奥行きのある画像に対してもポーズの判定が実現可能であると考えられる。ただし、ポーズの判定に3次元座標が必要となるため、図2-8のようなプロのポーズの静止画を学習データとすることはできない。すなわち、良いポーズの学習データも我々がRealSenseで撮影して用意する必要がある。



#### 4.4.1 Excel ファイルに記録するデータ

3次元の機械学習における、Excel ファイルに記録するデータについて説明する。

RealSense を用いて、ある時点での骨格点 33 箇所の座標を Excel ファイル (csv(Comma Separated Values)形式) に記録する。このとき、Web カメラでの Excel ファイルはデータ数が図 2-3 より  $33 \times 3$  であったが、RealSense を用いることで奥行き情報と 4.2.1 項で解説した統一後のスケールを追加で書き込むことにした。そのため、Excel ファイルに書き込む情報を図 4-9 のように変更する。変更後の Excel ファイルのデータ数は  $33 \times 6$  となる。

	A	B	C	D	E	F
1	$x[px]$	$y[px]$	$z[mm]$	$x[mm]$	$y[mm]$	信頼性
2	660	207	2.89	0.08	-0.51	1
3	665	207	2.89	0.09	-0.51	1
4	668	208	2.9	0.1	-0.51	1
5	647	206	2.87	0.04	-0.51	1
6	643	206	2.87	0.02	-0.51	1
7	638	207	2.89	0.01	-0.51	1
8	673	215	2.93	0.12	-0.49	1
9	634	213	2.9	0	-0.49	1
10	661	230	2.89	0.08	-0.44	1
11	644	230	2.93	0.03	-0.44	1
12	706	289	2.95	0.23	-0.26	1
13	600	280	2.93	-0.11	-0.28	1
14	713	372	2.95	0.25	0.01	0.97
15	580	365	2.93	-0.17	-0.01	0.96
16	718	449	2.9	0.26	0.25	0.97
17	574	440	2.87	-0.19	0.22	0.94
18	723	473	2.9	0.28	0.33	0.95
19	571	463	2.86	-0.2	0.29	0.9
20	716	471	2.89	0.25	0.32	0.95
21	577	461	2.86	-0.18	0.28	0.91
22	712	465	2.89	0.24	0.3	0.94
23	580	455	2.89	-0.17	0.27	0.9
24	679	453	2.78	0.13	0.25	1
25	616	449	2.89	-0.06	0.25	1
26	676	571	2.94	0.13	0.64	0.99
27	600	565	2.89	-0.11	0.61	1
28	674	671	2.9	0.12	0.95	0.99
29	593	666	2.88	-0.13	0.92	0.99
30	664	684	2.84	0.09	0.97	0.91
31	602	680	2.93	-0.1	0.98	0.92
32	701	704	2.87	0.2	1.04	0.98
33	573	701	2.81	-0.19	1.01	0.99

図 4-9 3次元の機械学習で用いる Excel ファイル

#### 4.4.2 奥行きを考慮した良いポーズ・悪いポーズの定義

3次元の機械学習における、奥行きを考慮した良いポーズ・悪いポーズの定義を説明する。ここでは、4.3.2項の実験結果をもとに、平面角 $-90, 90[^\circ]$ でのポーズを除いたポーズを判定の対象とする。

2.4.1項で解説したダブルバイセップスの評価ポイントを全て満たしているポーズを良いポーズ、それ以外のポーズを悪いポーズとする。また、全てのポーズは我々が RealSense を用いて記録したものとする。

良いポーズの具体例を図4-10、図4-11に示す。一方で悪いポーズの具体例を図4-12、図4-13に示す。特に、図4-11と図4-13のポーズでは、奥行き情報を利用できることを前提としている。



図4-10 良いポーズ1



図4-11 良いポーズ2



図 4-12 悪いポーズ 1



図 4-13 悪いポーズ 2

### 4.4.3 機械学習の条件

3次元の機械学習における学習条件を説明する。

様々な平面角を対象とした良いポーズ 200 枚、悪いポーズ 300 枚の合計 500 枚のそれぞれ異なるポーズを記録した Excel ファイルを用意し機械学習を行う。また、学習条件を表 4-2 に示す。

表 4-2 学習条件

エポック数	1000
バッチサイズ	100
イテレーション	5

### 4.4.4 機械学習の結果

4.4.3 項で解説した学習条件での学習結果を以下に示す。

4.4.2 項で定義した良いポーズ・悪いポーズの分類における正解率(2.6.6 項)は99.60%であった。また、学習結果の損失関数(2.6.6 項)の時間変化を図 4-14 に示す。これらの結果より、システムの学習が成功していることがわかる。

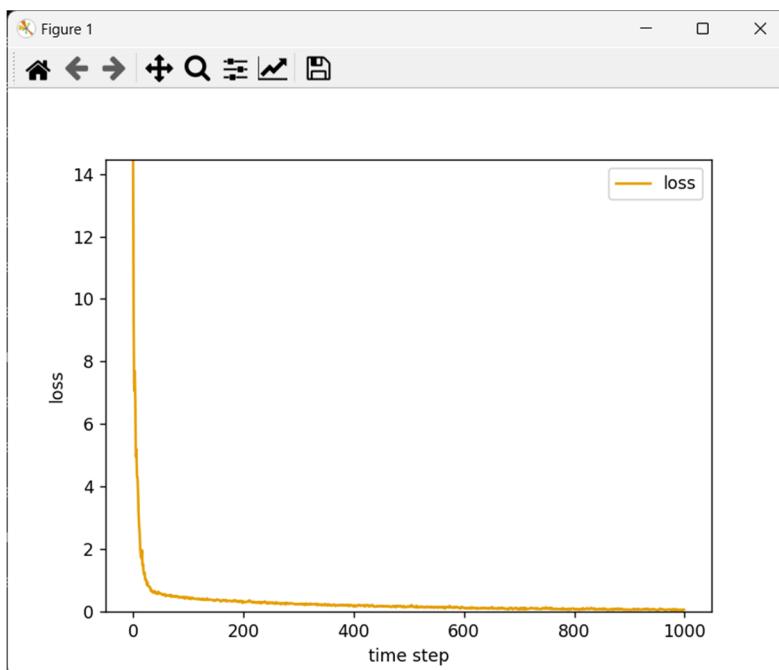


図 4-14 損失関数

## 4.5 奥行きを考慮したポーズ認識

本節では、前節までの内容を踏まえた3次元座標を用いたポーズ認識について説明する。

4.4.4 項で学習させた結果と RealSense を用いてポーズ認識を行う。システムによる分類結果を画像に上書きしたものを図 4- 15、図 4- 16、図 4- 17、図 4- 18 にそれぞれ示す。全ての図において、class0 と class1 によるポーズの分類が正しくできていることがわかる。また、Web カメラ(2次元)でのポーズ認識では対応していなかった、奥行きのあるポーズに対しても正しい結果を返している。したがって、我々が提案した RealSense を用いた3次元でのポーズ認識の手法が有用であることがわかった。

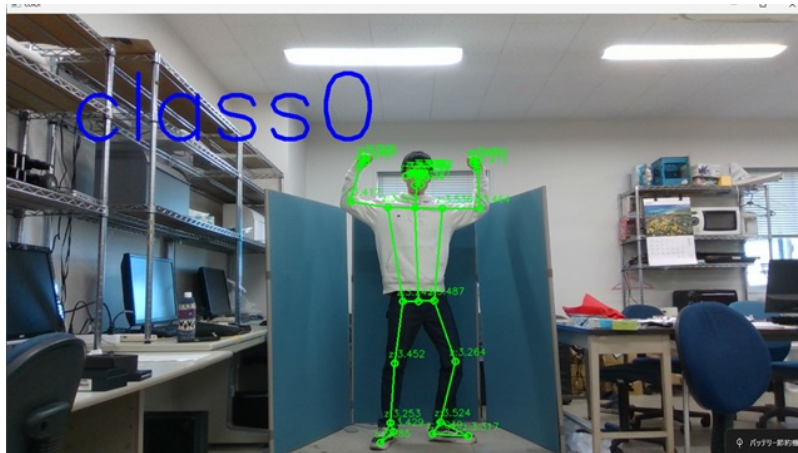


図 4- 15 正しく分類された良いポーズ 1

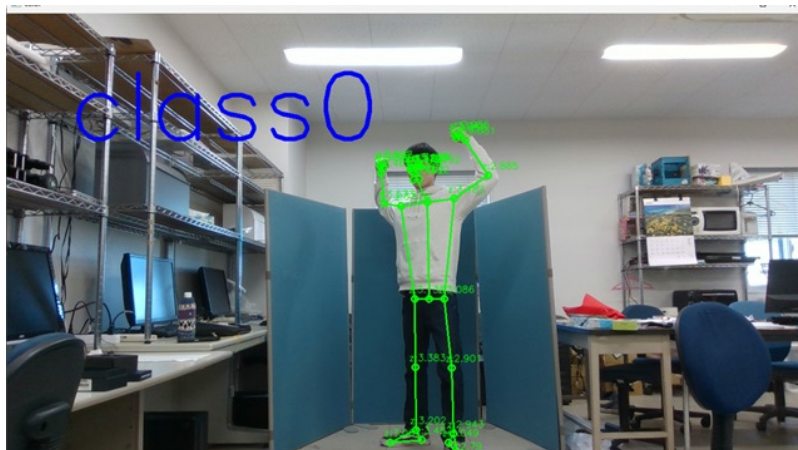


図 4- 16 正しく分類された良いポーズ 2

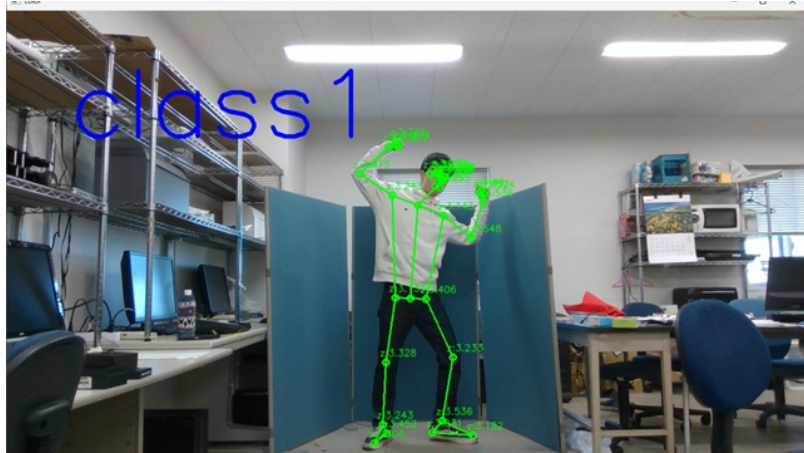


図 4-17 正しく分類された悪いポーズ 1

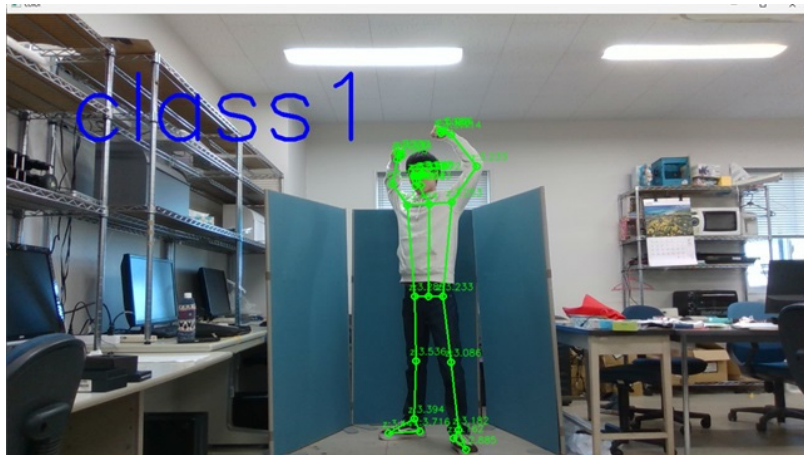


図 4-18 正しく分類された悪いポーズ 2

## 第5章 結論 (青井)

本研究では、下記の条件を目標に関節角度によるボディビルのダブルバイセップスの姿勢の評価システムの構築を目指してきた。

1. 関節角度からボディビルのポーズの良し悪しを評価すること
2. リアルタイムで評価を得ること
3. 評価結果が直感的にわかること

以上の条件を満たすよう、我々は関節角度による運動姿勢の評価システムの構築を行った。

条件1に対して、MediaPipeと機械学習を利用してシステムを構築した。構築したシステムでポーズの良し悪しの分類を行ったところ、90%の正解率が得られた。そのため、関節角度から姿勢を評価することができたと言える。また、RealSenseを用いることで奥行きのあるポーズに対してもポーズの評価が可能となった。

条件2に対して、PCで利用可能なカメラとMediaPipeから得られる機械学習の結果を組み合わせることで、リアルタイムで評価を行うことができた。

条件3に対して、良いポーズのclass0と悪いポーズのclass1の分類結果を画像上に表示した。class0かclass1を表示させることでわかりやすい、表示画面を構築できた。

以上より、3つの条件を満たす関節角度による運動姿勢の評価システムの構築を実現できた。しかし、実用化に向けては悪いポーズの学習データの増加や、悪いポーズの部位を表示できるようなシステムを構築する必要がある。

## 参考文献

[1] 村谷 博美,濱田 やえみ,江田 佳子,楠林 あかね,太田 美枝子,辻 利恵,幸地 英理子,木村 奈都美,米田 美佳,道本 典明,”コロナ禍における大学生の健康意識と生活習慣,健康度の自己評価,”人間科学,6 巻,p.1-12,(2024)

[https://www.jstage.jst.go.jp/article/hsksu/6/0/6\\_6.0\\_1/\\_html/-char/ja](https://www.jstage.jst.go.jp/article/hsksu/6/0/6_6.0_1/_html/-char/ja)

(閲覧日：2025 年 1 月 29 日)

[2] 長期データ | 特定サービス産業動態統計調査 | 経済産業省

[https://www.meti.go.jp/statistics/tyo/tokusabido/result/result\\_1.html](https://www.meti.go.jp/statistics/tyo/tokusabido/result/result_1.html)

(閲覧日：2025 年 1 月 29 日)

[3] 2017 年度 事業報告書 | 公益社団法人 日本ボディビル・フィットネス連盟

[https://www.jbbf.jp/download/Sokai/H30\\_Sokai/jigyohoukoku2017.pdf](https://www.jbbf.jp/download/Sokai/H30_Sokai/jigyohoukoku2017.pdf)

(閲覧日：2025 年 1 月 29 日)

[4] 2018 年度 事業報告書 | 公益社団法人 日本ボディビル・フィットネス連盟

[https://www.jbbf.jp/download/Sokai/R1\\_Sokai/2018%E5%B9%B4%E5%BA%A6%E4%B4%A8%E6%A5%AD%E5%A0%B1%E5%91%8A%E6%9B%B8.pdf](https://www.jbbf.jp/download/Sokai/R1_Sokai/2018%E5%B9%B4%E5%BA%A6%E4%B4%A8%E6%A5%AD%E5%A0%B1%E5%91%8A%E6%9B%B8.pdf)

(閲覧日：2025 年 1 月 29 日)

[5] 2024 年度 基本競技ルール講習 | 公益社団法人 日本ボディビル・フィットネス連盟

[https://www.jbbf.jp/attachments/e8c5232bf5dd6baba6eaae36585071841cb60692/store/5535344cdc73e1e39d58a94dccb48c8af3f0dede282d6e91c25552c8de7/2024JBBF%E5%9F%BA%E6%9C%AC%E8%AC%9B%E7%BF%92\\_%E7%AB%B6%E6%8A%80%E3%83%AB%E3%83%BC%E3%83%AB0227\\_OK.pdf](https://www.jbbf.jp/attachments/e8c5232bf5dd6baba6eaae36585071841cb60692/store/5535344cdc73e1e39d58a94dccb48c8af3f0dede282d6e91c25552c8de7/2024JBBF%E5%9F%BA%E6%9C%AC%E8%AC%9B%E7%BF%92_%E7%AB%B6%E6%8A%80%E3%83%AB%E3%83%BC%E3%83%AB0227_OK.pdf)

(閲覧日：2025 年 1 月 29 日)

[6] 2024 年度版 JBBF 競技ルールマニュアル | 公益社団法人 日本ボディビル・フィットネス連盟 競技ルール委員会

[https://www.jbbf.jp/download/Committee/2024%94N%93x%94%C5\\_JBBF%8B%A3%8BZ%83%8B%81%5B%83%8B%83%7D%83j%83%85%83A%83%8B20240620.pdf](https://www.jbbf.jp/download/Committee/2024%94N%93x%94%C5_JBBF%8B%A3%8BZ%83%8B%81%5B%83%8B%83%7D%83j%83%85%83A%83%8B20240620.pdf)

(閲覧日：2025 年 2 月 4 日)



[7] 加藤 司,安富祖 仁,“ダイビングスキル定着を促す MediaPipe を使用した動作比較システムの開発,”電気学会論文誌 D,143 卷,2 号,p.118-124,(2023)

[https://www.jstage.jst.go.jp/article/ieejias/143/2/143\\_118/\\_article/-char/ja/](https://www.jstage.jst.go.jp/article/ieejias/143/2/143_118/_article/-char/ja/)

(閲覧日：2025 年 1 月 29 日)

[8] Google, 2024/10/13, 「姿勢ランドマーク検出ガイド」, Google AI for Developers,

[https://ai.google.dev/edge/mediapipe/solutions/vision/pose\\_landmarker?hl=ja](https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker?hl=ja)

(閲覧日：2025 年 1 月 4 日)

[9] Kazuhito00, 2021/12/14, 「v0.8.9」, GitHub

<https://github.com/Kazuhito00/mediapipe-python-sample/releases/tag/v0.8.9>

(閲覧日：2023 年 6 月 13 日)

[10] ガンマソフト株式会社, 2011/01/11, 「ピクセル座標とは：グーグルマップのしくみを探る(2)」, maple.net,

<https://www.mapli.net/blogs/google-maps-mechanism/how-google-map-works2/>

(閲覧日：2024 年 7 月 2 日)

[11] 杉元 勝盛, 2024/01/12, 「ダブルバイセップスのやり方は？アピールする筋肉や魅力的に見せるための 3 つのポイントを徹底解説」, T-BALANCE,

<https://t-balance-gym.com/fcul/double-biceps#3>

(閲覧日：2024 年 10 月 23 日)

[12] 金丸 隆志(2019), 「RaspberryPi ではじめる機械学習～基礎からディープラーニングまで～」, 株式会社講談社.

[13] zivsha, 2018/02/26, 「Pyrealsense2: Question about 3D reconstruction problem #1231」, GitHub

<https://github.com/IntelRealSense/librealsense/issues/1231>

(閲覧日：2024 年 10 月 29 日)

## 謝辞

最後に、この場をお借りして本研究を進めるにあたり、2年間多大なご指導を頂きました金丸隆志教授、協力していただいた研究室の先輩、学科の同級生に班一同、心より感謝いたします。ありがとうございました。