

# 深層学習を用いた人物の関節点時系列データ からの動作推定の一提案

下山 朗弘\* 藪木 登\*\*

## A Proposal for Human Motion Estimation from Time-series Skeletal Joint Data Using Deep Learning

SHIMOYAMA Akihiro, YABUKI Noboru

This study proposes a human motion estimation system using AI for welfare applications, aiming to facilitate real-time recognition and detection of potentially hazardous movements. Unlike conventional sensor-based methods, the system utilizes standard color cameras and deep learning models to reduce user burden.

Skeletal joint data is extracted from video input using the OpenPose library. These data are then used to train motion estimation models based on Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM), using the STAIR Actions dataset, which includes categorized daily activities relevant to caregiving.

Experimental results show that LSTM outperforms RNN in estimation accuracy due to its ability to handle long-term dependencies. However, LSTM is more sensitive to missing data, especially when body parts are occluded. Category-specific analysis revealed that motion categories involving occlusions had lower accuracy, likely due to limitations in joint detection.

Finally, the trained model was applied to live camera input, confirming the system's capability for real-time motion estimation and individual tracking.

*Key Words: Neural Network, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Deep Learning*

### 1. 緒 言

現在、高齢化が進む日本において全体の高齢者が占める割合や一人暮らしの高齢者が増加している。また、世界でも少子高齢化は問題となっており、中国では高齢者人口の増加が深刻になってきている。また、近年 AI 技術は急速に発展し、福祉分野においても AI 技術が注目され活用されている。そこで AI を用いた人物の動作推定システムを福祉に用いることで、対象者を認識し動きを推定することが容易になる。さらにシステムを用いてリアルタイムな動作記録や危険な動きを検知して介護ロボットや通知システムへの活用などが考えられる。

人物の動作推定を行う方法としてはセンサーを用いた方法がある。例えば、スマートフォンを身につけて内蔵の加速度センサーから取得した加速度

データからの行動推定<sup>1)</sup>や室内に設置したビーコンや対象者が身につけるスマートタグから得られるセンシングデータと動画画像解析で取得した骨格データを用いた行動認識<sup>2)</sup>などがある。しかし、これらの方法では対象者がセンサーなどを身につけ、あるいは、室内の各所にビーコンやセンサーなどを設置してデータを取得する必要がある、装置の装着や設置が利用者の負担となってしまう。

本研究では、室内に設置したカメラの入力映像から人物の関節点データを取得し、深層学習モデルを用いてリアルタイムで処理が可能な動作推定システムの一手法を提案する。本提案より、加速度センサーなどの特殊センサーなどを使用せずに一般的なカラーカメラのみを用いた人物の動作推定を行うことができるようになり、利用者の負担を軽減することができる考える。

そのためにまず、動画画像から人物の関節点データを取得する。その人物の関節点データの取得には姿勢推定ライブラリである OpenPose<sup>3)</sup>を用いる。これにより、関節点データを動画画像から抽出する作業を

原稿受付 令和7年9月10日

\*専攻科 電子・情報システム工学専攻 令和7年3月修了

\*\*総合理工学科 教授

削減することができる。

次に、動画像から抽出した人物の関節点データから動作推定を行うために動作推定モデルの学習を行う。深層学習モデルには、時系列データを扱うことができる Recurrent Neural Network (RNN)<sup>4)</sup>と Long Short-Term Memory (LSTM)<sup>5)</sup>を使用し、深層学習モデルの比較を行う。この時、学習時のデータセットには福祉分野での活用を想定しているため、STAIR Actions<sup>6)</sup>を使用する。STAIR Actionsには日常生活における動作カテゴリ別に分類された動画像が含まれている。

最後に、動作推定システムに学習済みのモデルを適用して実際のカメラからの入力映像に対してリアルタイムに動作推定を行うことができるか確認を行う。

## 2. 関連技術

### 2.1 STAIR Actions

本研究でのモデルの学習に使用するデータセットである STAIR Actions について説明する<sup>6)</sup>。図1に動作カテゴリリストを示す。STAIR Actionsとは約10万本の日常の人間の動作シーンの動画像で

<b>Kitchen related</b>	
drinking	
eating meal	
eating snack	
washing dish	
throwing trash	
washing hands	
opening refrig door	
pouring tea or coffee	
cutting food	
cooking	
<b>Washroom related</b>	
setting hair	
drying hair with blower	
making up	
manicuring	
gargling	
brushing teeth	
washing face	
shaving	
<b>Object manipulation</b>	
wearing glass	
playing with toy	
playing board game	
using computer	
listening to music with headphones	
playing computer game	
taking photo	
using smartphone	
using tablet	
operating remote control	
watching TV	
telephoning	
gardening	
playing guitar	
playing piano	
blowing flute	
standing on chair or table or stepladder	
throwing	
opening or closing container	
smoking	
ironing	
knitting or stitching	
polishing shoe	
wearing shoes	
sewing	
hanging out or capture laundry	
folding laundry	
wearing tie	
putting off cloth	
putting on cloth	
housecleaning	
wiping window	
drawing picture	
doing origami	
reading newspaper	
studying	
reading book	
writing	
<b>Multipayer action</b>	
changing baby diaper	
bottle-feeding baby	
piggybacking someone	
holding someone	
feeding baby	
assisting in getting up	
assisting in walking	
teaching	
nodding	
shaking head	
speaking	
hearing	
pointing with finger	
caressing head	
kissing	
doing high five	
hugging	
stroking animal	
shaking hands	
bowing	
giving massage	
passing something	
doing paper-rock-scissors	
fighting	
<b>Solo action</b>	
walking with stick	
walking	
going up or down stairs	
jumping on sofa or bed	
baby crying	
baby crawling	
exercising	
dancing	
running around	
clapping hands	
sitting down	
standing up	
sleeping on bed	
lying on floor	
leaving room	
entering room	
being angry	
being surprised	
crying	
smiling	

図1 STAIR Actions データセットのカテゴリ一覧

構成されたデータセットである。100種類の動作カテゴリが存在し、それぞれの動作カテゴリにつき、約1000本の動画像で構成されている。含まれている動画像は料理や読書のような日常生活における動作の動画像となっている。

### 2.2 OpenPose

動画像からの姿勢推定に使用する姿勢推定ライブラリである OpenPose について説明する<sup>3)</sup>。OpenPose とはカーネギーメロン大学などによって開発された深層学習を用いて人の関節点などのキーポイント情報をリアルタイムに抽出する姿勢推定ライブラリである。人の関節点の座標値情報の取得は従来の技術であれば、対象の人物にセンサーなどを取り付けたリ、Kinect などの三次元情報を取得することができるカメラなどを利用したりして取得していた。しかし、OpenPose では加速度センサーや三次元カメラなどの特殊な機材を使わずに、一般的なカラーカメラによる画像や動画像から関節点の座標値を抽出することができる。STAIR Actions の動画像に対して関節点の座標値抽出を行い、人物の25個の関節点を動画像内から推定し、座標値を出力する。図2に OpenPose で推定する各関節点と部位の対応図<sup>8)</sup>を示す。

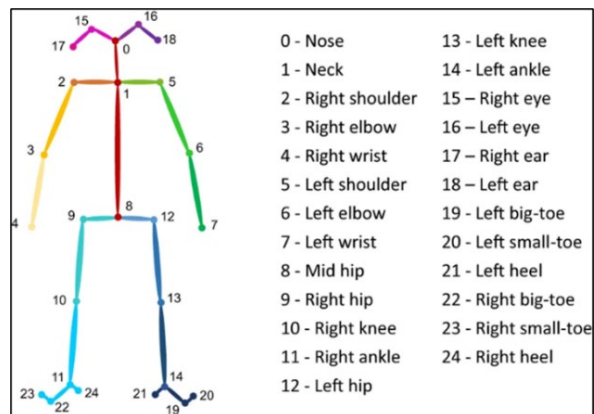


図2 各関節点の対応図<sup>8)</sup>

## 3. 動作推定システムの提案

### 3.1 動作推定システムの概要

提案する動作推定システムの概要<sup>9)</sup>について説明する。図3に提案システムの概要図を示す。本研

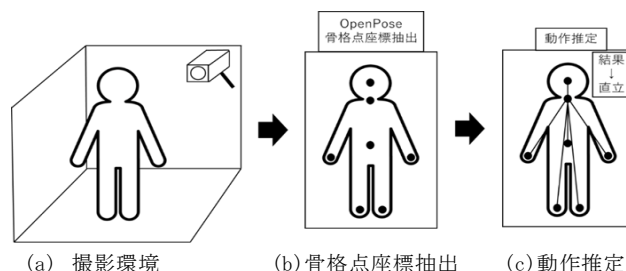


図3 提案システムの概要図

究では屋内での利用を前提にしている．図 3(a)のように室内に設置されたカメラの入力映像から OpenPose を用いて関節点を検出し，関節点の座標値を取得する(図(b))．次に取得した座標値データを学習済みの深層学習モデルを用いて動作推定を行い，推定結果を出力する(図(c))．

### 3. 2 動作推定ネットワークの構造

動作推定のネットワーク構造<sup>9)</sup>について説明する．図 4 にネットワーク構造を示す．データは図の下側から上側に向けて移動する．入力データは  $t$  時間のフレーム  $F$  の関節点データがモデルレイヤに入力される．モデルレイヤには，動作を推定するために時系列データである関節点の座標値変化を扱うため，RNN モデルまたは LSTM モデルを使用する．モデルレイヤからの出力は次フレームの入力とともに次のモデルレイヤに入力される．そして最後のモデルレイヤからの出力を全結合層に入力し，その出力を Softmax 層に入力する．Softmax 層に入力することで全結合層の出力値を 0.0~1.0 の範囲の確率値に変換する．そして変換後の値で最も推定確率の高いカテゴリを推定結果として出力する．なお，図 4 では，最終フレーム以外の全結合層は利用していない．

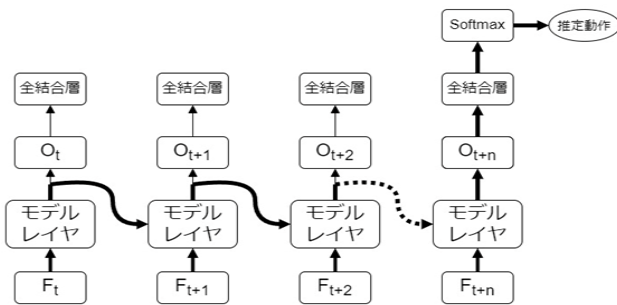


図 4 動作推定のネットワーク構造

## 4. 実験方法

### 4. 1 動作推定の深層学習モデルの学習

まず，動作推定モデルの学習プログラムについて説明する．プログラムはプログラミング言語に Python を使用し，PyTorch という機械学習のオープンソースライブラリを使用した．深層学習モデルである RNN と LSTM は PyTorch のクラスとして定義されており，これを用いて図 4 のネットワーク構造を実装した．

次に，学習に用いるデータセットについて説明する．使用した動作カテゴリは 78 カテゴリとなっている．STAIR Actions の動作カテゴリは図 1 に示したように 100 カテゴリに分類されているが，一部の動作カテゴリが非公開となっているため，本研究では公開されている 78 個の動作カテゴリを使用している．

さらに，この 78 カテゴリを動画像内に存在する人数ごとに選別したデータセットを構築した．動画像内に一人のみのカテゴリは 50 カテゴリ，二人存在するカテゴリは 28 カテゴリとなっている．表 1 に各カテゴリのデータ数を示す．学習データとテストデータは各カテゴリで 8 : 2 の割合とした．

深層モデルの学習では OpenPose の出力である関節点の座標値を使用する．学習時に OpenPose を実行すると GPU のメモリを消費するため，事前にデータセットの各動画像に対して OpenPose を実行し，次の 4.2 で説明する変換を行って関節点の座標値データを抽出してデータセットを構築したのち，学習を行った．

表 2 に実験条件を示す．構築した三つのデータセットを用いてそれぞれ RNN と LSTM の場合の学習を行う．モデルに入力するフレーム数は 60 フレームとしている．STAIR Actions の各動画像は 30FPS の約 5 秒となっており，総フレーム数は約 150 となるが，動画像ごとにフレーム数がバラバラで一定の秒数で切り出すと動作の一部が抜けるため，本実験では各動画像の総フレーム数から等間隔に 60 フレームを抜き出して入力データとしている．

RNN と LSTM の隠れ層は 512 とし，過学習を抑制するためのドロップアウトは 0.5，バッチサイズは 128 としている．バッチサイズとは 1 回の学習において一度に処理を行うデータ数を意味する．学習回数は 3000 回とし，一回の学習を終えるごとにテストデータでモデルの推定精度と Loss の計測を行う．なお，Loss とは損失関数によるモデルの予測値と正解値の差である．

表 1 学習用データセットのデータ数

	全カテゴリ	一人のみ	二人
カテゴリ数	78	50	28
学習データ数	37,129	23,440	13,689
テストデータ数	9,296	5,861	3,435

表 2 各カテゴリにおける実験条件

データセット	全カテゴリ		一人のみ		二人	
	RNN	LSTM	RNN	LSTM	RNN	LSTM
深層学習モデル						
カテゴリ数	78		50		28	
入力フレーム数	60		60		60	
隠れ層	512		512		512	
ドロップアウト	0.5		0.5		0.5	
バッチサイズ	128		128		128	
学習回数	3000		3000		3000	

### 4. 2 データセットの座標変換処理

データセットの座標変換処理について説明する．学習時には関節点の座標値データを使用するが，STAIR Actions データセットは動画像データとなっているため，座標値データへの変換を行う．図 5

に座標変換処理の全体の流れを示す。図の上側から、まず、データセットを読み込み、各動画をフレームに分割する。その後、分割したフレームに対して OpenPose の関節点抽出を行う。OpenPose の出力には検出した人物ごとに 25 個の関節点の X 座標と Y 座標、推定の信頼度が出力される。しかし、フレームごとに実行するためフレーム間の統一された ID などの情報がなく、複数の関節点データが抽出された場合、人物ごとにまとめることができない。そこで、PersonID(PID)を付与するプログラムを作成する。

図 6 に PID 付与のプログラムを示す。このプログラムでは PID と最後に検出された関節点データを保持する PID リストを作成してフレーム間で統一した ID を付与する。まず、OpenPose で抽出した関節点が入力されると、PID リスト内に要素がなければ、新規の PID を発行して関節点とともにリストに追加する。PID リスト内に要素が存在する場合 PID リストに保持されている各 PID の直前の関節点データと入力された関節点データを式(1)を用いて各関節点のユークリッド距離を計算する。i は各関節点を表しており、(X2,Y2)は入力された関節点座標、(X1,Y1)は直前の関節点座標を表している。

$$\sum_{i=1}^{25} \sqrt{(X_{2i} - X_{1i})^2 + (Y_{2i} - Y_{1i})^2} \quad (1)$$

本実験ではフレームレートが 30FPS となっており、同一の人物が映っていた場合、前後のフレーム間で座標値は大きく変わらないと考えられる。そのため、前後のフレームで関節点座標のユークリッド距離を求め、最小値の PID が同一の人物であるとして PID リストの更新を行う。もし、すべての PID の関節点座標とのユークリッド距離が閾値より大きい場合は新たに検出された人物として新規の PID を発行する。

PID を付与した後は関節点座標の変換<sup>9)</sup>を行ってから、PID ごとに蓄積する。座標変換を図 7 に示す。座標変換を行う理由としては、OpenPose から出力される関節点の座標は画面左上からの絶対座標となっているため、この座標を用いて学習を行うと画角の変化に対して座標の値が大きく影響を受ける。そのため、本実験では座標を図 7 に示すように画面左上からの絶対座標から首の関節点を基準にした相対座標に変換して使用する。その後、使用するデータセットの各動画の縦横比率がバラバラなため、画像の縦幅値と横幅値で座標データの x 値と y 値をそれぞれ割り、座標データの正規化を行う。なお、関節点が画像に映っていない場合は、x, y の各値が 0 になっており、この場合、関節点は座標変

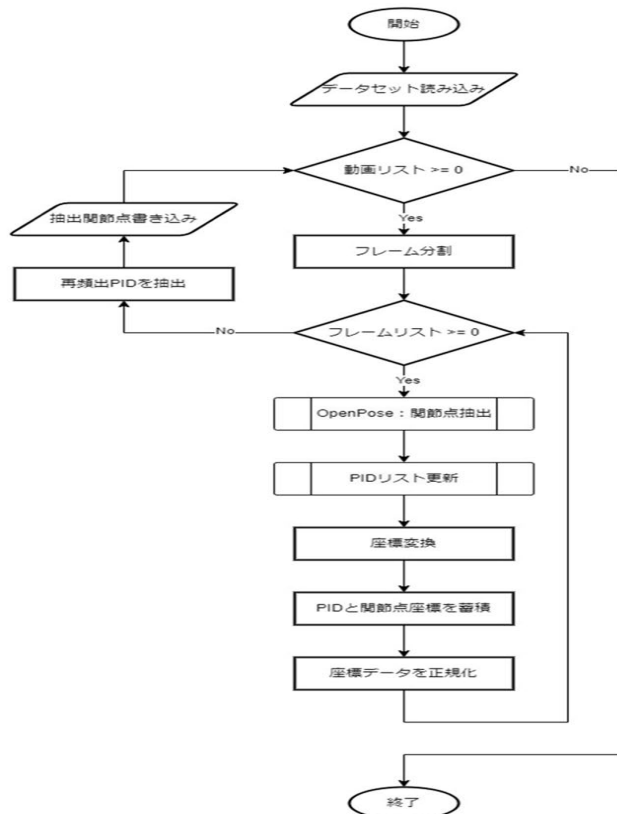


図 5 データセット処理の全体の流れ

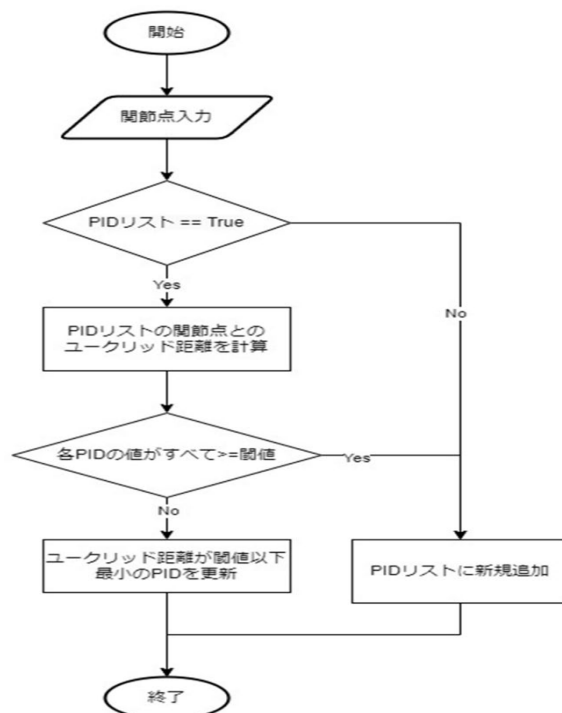


図 6 PID 付与プログラムの流れ

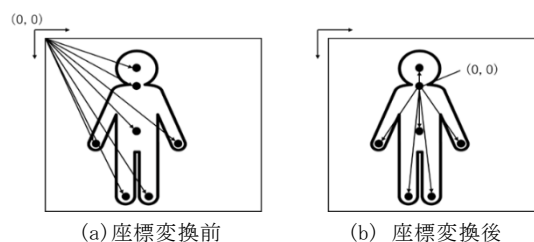


図 7 関節点の座標変換

換を行わず、そのまま各値を0としている。

図8に1フレーム分のOpenPose出力時と座標変換と正規化後の人物の関節点出力の例を示す。

すべてのフレームを処理した後は、関節点データのファイルへの書き込みを行う。しかし、PIDリストの中には人物ではない物体などをOpenPoseが人物だと誤検出した値も含まれている。この例では図左側の蛇口を誤検出している。それらの値は継続的ではなく、数フレームのみ検出されるため、PIDリストの中で全フレーム中、最も検出されたPIDの関節点データを抽出してファイルに書き込む。



図8 1フレーム分の関節点データの出力例

## 5. 実験結果

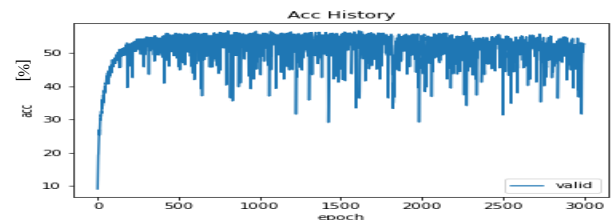
### 5.1 RNNモデルの学習状況推移

深層学習モデルにRNNを使用して学習したときの、モデルの推定精度を図9に、Lossの推移図を図10に示す。それぞれの推定精度(acc[%])を見ると学習回数が500回付近ですでに頭打ちとなっており、その後はドロップアウトを適用しているため、精度の大きな低下と上昇を繰り返していることがわかる。

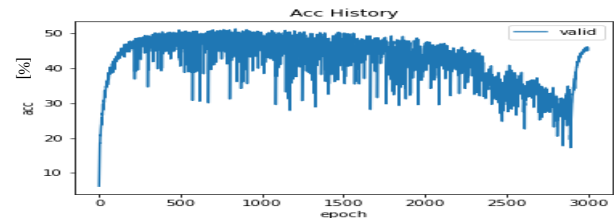
図9(a)全カテゴリーでは、データ数が多いためか、他のデータセットと比べ、推定精度は下がっていないことがわかる。また、図10(a)Lossの推移図を見ると、学習回数が250回付近で下がりきっており、その後は上昇して横ばいとなっている。

図9(b)一人のみの場合は推定精度のグラフから頭打ちとなってから徐々に低下し2800回付近まで低下していることや、図10(b)Lossの推移図を見ても、2500回から3000回の間で推定精度のグラフと反比例していることがわかる。図9(c)二人の場合を見ると推定精度のグラフが頭打ちになってから時々、緩やかな減少と上昇を繰り返している箇所がある。これはデータ数が最も少ないためだと思われる。

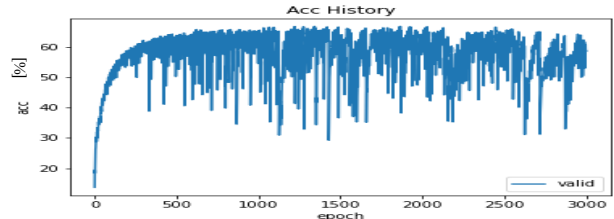
表3に推定精度とLossの最高値の比較表を示す。



(a) データセット：全カテゴリー

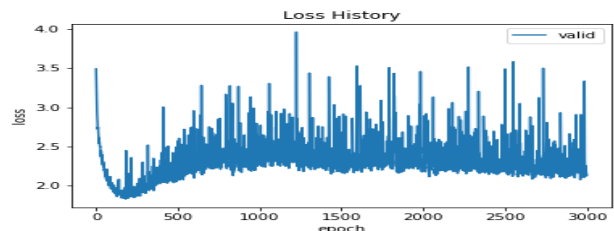


(b) データセット：一人のみ

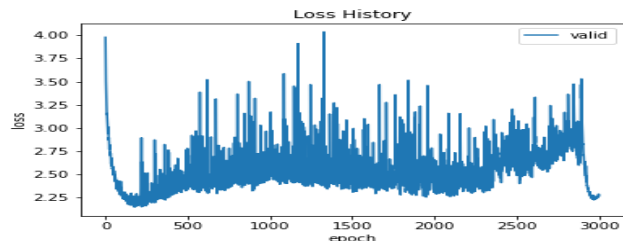


(c) データセット：二人

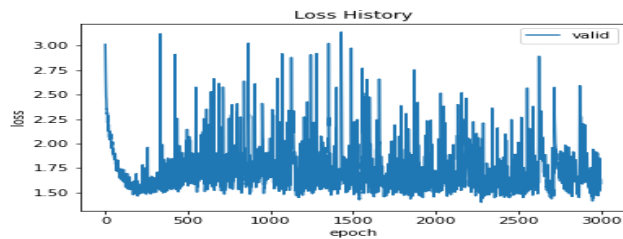
図9 RNNの推定精度



(a) データセット：全カテゴリー



(b) データセット：一人のみ



(c) データセット：二人

図10 Lossの推移図

表3 推定精度の比較表

データセット	全カテゴリー	一人のみ	二人
推定精度	51.320%	56.744%	66.928%
Loss	2.144	1.829	1.401

表 1 と表 2 からデータ数が少ないデータセットほど推定精度が高くなっており、Loss が小さくなっていることがわかる。

推定精度と Loss のデータだけでは判断できないが、過学習の影響も否定できないと考えられる。

### 5.2 LSTM モデルの学習状況推移

次に LSTM を使用して学習したときのモデルの推定精度を図 11 に、Loss の推移図を図 12 に示す。それぞれの推定精度を見ると学習回数が 150 回付近で上昇しきっており、その後はドロップアウトの影響のために大きく精度が低下している箇所があるが全体的に横ばいとなっている。一人のみと二人の場合を見ると上昇しきってから数百回学習している間は RNN と同じようなグラフとなっているが、その後は一定周期で推定精度が大きく低下と素早い推定精度の回復を繰り返している。RNN と比べて精度低下の回数が少なく安定している。

LSTM は、RNN の時と比べ緩やかな精度の低下は、推定精度のグラフからは見られない。しかし、Loss の推移図を見ると異なる動きとなっている。通常だと推定精度が上昇すると Loss は低下し反比例しているようなグラフになるが、それぞれのデータセットで共通して推定精度が上昇中は反比例しているが、その後は上昇している。考えられる原因としてはデータセットに含まれる各カテゴリーが持つ動画像データ数がバラバラで多いカテゴリーと少ないカテゴリーが存在していることや OpenPose で検出できなかった関節点の欠損値の影響が RNN と比べて大きく出たのではないかと考えられる。

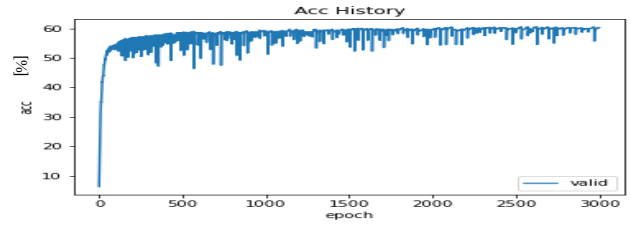
LSTM は RNN と比べ長い時系列データを扱う能力が高いとされているため、データセットに含まれる欠損値が RNN では情報が長期間にわたって保持されなため、欠損値の影響が少ないが、LSTM だと長期間にわたって情報を保持する特徴によって欠損値の影響が大きく出たと思われる。

前節最後でも述べたが、推定精度と Loss のデータだけでは判断できないが、過学習の影響も否定できないと考えられる。

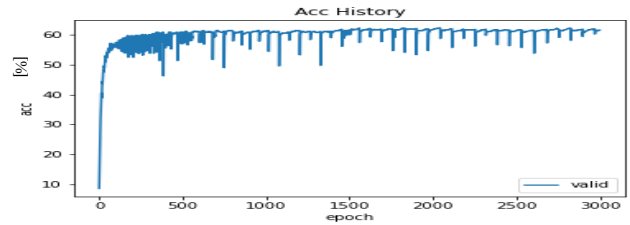
表 4 に推定精度と Loss の最高値の比較表を示す。この表から RNN と同様データ数が少ないデータセットほど推定精度が高くなっており、Loss が小さくなっている。

### 5.3 カテゴリーごとの推定精度

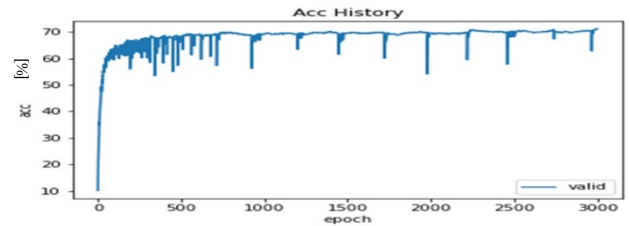
次に各カテゴリーの推定精度を測定した。表 5、6 にそれぞれ RNN, LSTM を用いた場合の各データセットの推定精度上位 5 カテゴリーと下位 5 カテゴリーを示す。同じデータセットを用いて学習を行った場合 RNN と LSTM のどちらも推定精度の高いカテゴリーと低いカテゴリーは同じようなカテゴリーに分類されており、深層学習モデルの違い



(a) データセット：全カテゴリー

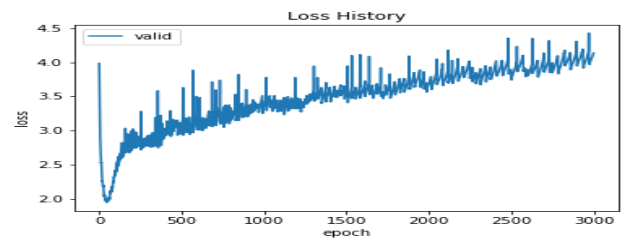


(b) データセット：一人のみ

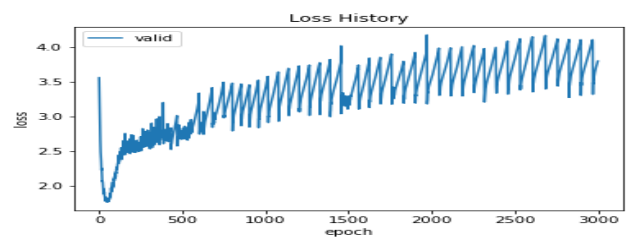


(c) データセット：二人

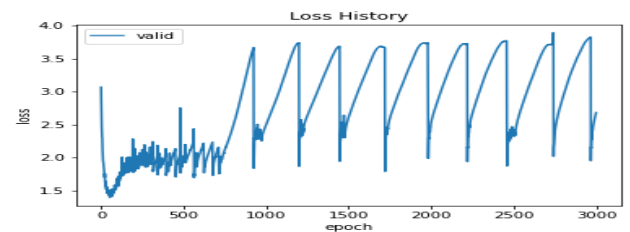
図 11 LSTM の推定精度



(a) データセット：全カテゴリー



(b) データセット：一人のみ



(c) データセット：二人

図 12 Loss の推移図

表 4 推定精度の比較表

データセット	全カテゴリー	一人のみ	二人
推定精度	60.586 %	62.307 %	71.237 %
Loss	1.950	1.762	1.395

表 5 RNN でのカテゴリー推定精度

データセット	上位		下位	
	5 カテゴリー		5 カテゴリー	
全カテゴリー	lying_on_floor [78.3%] baby_crawling [76.7%] wearing_shoes [73.3%] wadhing_dis [69.4%] jumping_on_sofa_or_bed [68.7%]	eating_meal [13.7%] throwin [11.3%] using_computer [4.2%] reading_book [3.4%] telephoning [0.0%]		
一人のみ	lying_on_floor [87.4%] washing_face [80.3%] bowing [79.8%] washing_dish [78.1%] wearing_shoes [76.4%]	reading_newspaper [22.6%] throwing [22.5%] cutting_food [22.0%] reading_book [13.8%] using_computer [8.3%]		
二人	jumping_on_sofa_or_bed [82.0%] assisting_in_getting_up [81.4%] baby_crawling [79.5%] maniburing [78.1%] polishing_shoe [76.1%]	listening_to_music_with_headphones [45.6%] shaking_head [43.4%] hugging [42.5%] caressing_head [39.3%] telephoning [0.0%]		

表 6 LSTM でのカテゴリー推定精度

データセット	上位		下位	
	5 カテゴリー		5 カテゴリー	
全カテゴリー	washing_face [84.3%] bowing [78.6%] lying_on_floor [77.6%] walking_with_stick [75.0%] washing_dish [73.2%]	throwing [16.9%] eating_snack [16.7%] eating_meal [15.1%] reading_book [9.2%] telephoning [0.0%]		
一人のみ	wadhing_face [81.1%] lying_on_floor [79.0%] sitting_down [78.1%] wearing_shoes [76.4%] washing_dish [75.4%]	eating_snack [19.2%] drinking [16.7%] cutting_food [14.6%] using_computer [12.5%] reading_book [6.9%]		
二人	jumping_on_sofa_or_bed [86.0%] baby_crawling [80.8%] holding_someone_on_back [78.8%] assisting_in_getting_up [76.8%] assisting_in_walking [75.2%]	shaking_head [33.7%] hugging [32.5%] passing_something [29.7%] listening_to_music_with_headphones [27.8%] holding_someone [26.2%]		

によるカテゴリーの推定精度に大きな違いはない。各カテゴリーの推定精度において、上位カテゴリーの下位カテゴリーで大きく推定精度に差が発生している。下位カテゴリーにおいて、reading\_book (読書の動画像) や using\_computer (パソコンの使用), eating\_meal (食事) などのカテゴリーの推定精度が低くなっている。この理由としてこれらのカテゴリーの動画像に映る人物は机や物によって体の一部が隠れている動画像が多く含まれているためだと思われる。物体によって人体が遮られている場合、OpenPose でうまく関節点の抽出ができない場合が多く、欠損値として出力されるため、これらのカテゴリーが上位のカテゴリーと比べ学習データに多くの欠損値が含まれているため推定精度が低くなったと思われる。

今後は物体に遮られた場合の推定向上に取り組む必要がある。考えられる対策としては人物の関節点の抽出とともに空間の物体認識を行い、検出した人物に近い物体の情報を推定の特徴量として利用すれば精度を上げられるのではないかと考えられる。

例えば、読書をしている人物が椅子に座り机によって下半身が遮られているような場合、現状だと上

半身の関節点のみで推定を行っているが、物体認識によって机と本を検出し、人物の関節点の近くに机があり、手の関節点と本が近くにあるというように特徴量を使うことで推定精度を上げられるのではないかと考えられる。

## 6. カメラ入力における動作推定システムの構築

### 6.1 システムの構成について

図 3 をベースにカメラからの入力映像による動作推定システムの流れを図 13 に示す。まず、カメラから入力された動画像は静止画像として分割し、OpenPose で関節点データを抽出する。抽出したデータは PID 付与の処理を行い、PID リストに同じ PID で関節点を保存していく。

ここで、図 3 のシステムを長時間動作させるため、図 6 において、「ユークリッド距離が閾値以下最小の PID を更新」の処理の次に、2 つの処理を追加した。「ライフカウントが 0 の PID 削除」、「スタックカウントの更新」である。ライフカウントとは、OpenPose の誤検出や画像外へ移動した人物の PID をカウントするものであり、前者は、その数が連続して既定数以上になると対象がないとして、その関係する PID 数を削除する。また、スタックカウントは、保存した関節点データの数を表し、後者は、この数が既定した数値を超えると、保存した関節点データを学習済みモデルに送る。

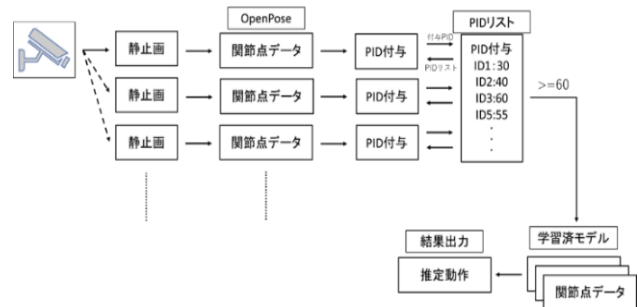


図 13 動作推定システムの流れ

### 6.2 動作確認

実際に作成したシステムを用いて動作確認を行った。図 14 に、例として、椅子に座り、本を持っている様子を撮影し、動作推定を行ったときの 1 フレームの画像結果を示す。学習済みモデルには一人のみのデータセットで学習した LSTM モデルを使用した。動作推定結果は図下側に現在の PID とその推定動作が出力される。推定結果としては standing\_up や wearing\_shoes など椅子に座った姿勢やそれに近い姿勢のカテゴリーと推定することが多かった。

5.2 のカテゴリーの推定精度でも指摘したとおり、読書という動作カテゴリーの推定精度が低いこと



図 14 実行画面

が原因だと思われる。しかし、OpenPose で検出した関節点データの PID 管理がうまく動作しており、実際にカメラ入力からの映像に対して動作推定を行えることが確認できた。

## 7. まとめ

本研究ではカメラからの入力映像から人物の関節点データを取得し、深層学習モデルを用いてリアルタイムな処理が可能な動作推定システムの一手法を提案した。

まず、姿勢推定ライブラリである OpenPose を用いて、動画像から人物の関節点データを取得する。取得した関節点データから動作推定を行うために深層学習モデルの学習に取り組んだ。深層学習モデルには RNN と LSTM を用いてネットワークを構成し、学習用のデータセットを作成して学習を行った。

各モデルの学習後のテストデータに対する動作の推定精度は、RNN と比べて LSTM の方が高い結果となった。LSTM は RNN と比べて長い系列データを扱う能力が高く、フレーム間の関節点データの変化量をより効果的に学習できたためだと思われる。しかし、この長時間にわたって情報を保持する特徴によってデータセットに含まれる欠損値が Loss の推移に大きく影響したと思われる。

動作カテゴリー別の推定精度を計測した結果は、モデルによって大きな推定精度の差は見られなかったが、カテゴリーの推定においては、大きく差が出る結果となった。理由としては、カテゴリーに含まれる動画内の人物が物体によって遮られているようなカテゴリーの推定精度が低くなる傾向から、OpenPose によって関節点を抽出できない場合が多く、欠損値が多く含まれてしまうからと思われる。

この問題に対して今後は物体に遮られた場合の推定精度向上に取り組む必要がある。例えば、OpenPose の関節点抽出とともに YoLo などを用いて物体検出を行い、各関節点との距離と種類を特徴量

として用いることが考えられる。

最後に、本提案システムに学習済みモデルを用いて、実際にカメラからの入力映像に対して動作推定を行った。入力映像に対して OpenPose で関節点抽出を行い、PID で検出人物を管理することで動画内の人物に対してリアルタイムな動作推定ができ、提案したシステムの有効性を確認した。

今後の課題としては、学習済みモデルの推定精度の向上があげられる。学習時のパラメータの調整や物体に遮られた場合の対策として空間の物体情報の利用、学習に利用している STAIR Actions のデータセットのカテゴリーの選別、福祉や介護の分野に特化した動作カテゴリーの自作データセットの作成などが今後の課題となる。

AI を用いた動作推定システムを福祉に用いることで、対象者を認識し動きを推定しリアルタイムな動作記録や危険な動きを検知して介護ロボットや通知システムへの活用などが考えられる。

## 参 考 文 献

- 1) 植田智明, 杉村博, 松本一教, 一色正男, “センサデータからの人間の行動推定”, 2013 年度人工知能学会全国大会, 2013.
- 2) 小村皓大, 堀川三好, 岡本東, “センシングデータと骨格データのマルチモーダル学習による作業者の動作推定”, 日本経営工学会論文誌, pp. 31-39, 74 巻, 2 号, 2023.
- 3) Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.172-186, vol.43, 2021.
- 4) Jeffrey L. Elman, “Finding structure in time”, Cognitive Science, pp.179-211, vol.14, No.2, 1990.
- 5) Sepp Hochreiter, Jürgen Schmidhuber, “Long short-term memory”, Neural Computation, pp.1735-1780, Vol.9, No.8, 1997.
- 6) 吉川友也, 竹内彰一, “家庭やオフィス内の動作認識用大規模動画データセットの構築”, 平成 29 年度人工知能学会全国大会 (JSAI2017), 2017.
- 7) Yuya Yoshikawa, Jiaqing Lin, Akikazu Takeuchi, “STAIR Actions: A Video Dataset of Everyday Home Actions”, arXiv preprint arXiv:1804.04326, Apr.2018.
- 8) Mingming Zhang, Yanan Zhou, Xinye Xu, Ziwei Ren, Yihan Zhang, Shenglan Liu, “Multi-view emotional expressions dataset using 2D pose estimation”, Sci Data 10, 649, 2023.
- 9) 下山朗弘, 藪木登, 築谷隆雄, “深層学習を用いた日常生活における人物の動作推定”, 2024 年度 (第 75 回) 電気・情報関連学会中国支部連合大会, R24-24-04, 2024