

スマートウォッチを用いたトランポリン上の動作検出

小林慧¹ 長谷川達人¹

概要:本研究では、スマートウォッチに搭載されている加速度センサを用いてトランポリン上で行った動作を検出する手法を開発する。スマートウォッチからトランポリン上での動作を検出することで家庭用トランポリンを用いてトランポリン運動を促進するシステムの開発につながる。本研究では、Convolutional Neural Network (CNN) を用いた手法を提案した。被験者 8 名に対し、Leave-One-Subject-Out Cross Validation (LOSO-CV) で性能評価した結果、提案手法において 78.8%の推定精度が得られ、ベースライン手法と比較した場合でも最良の推定精度であった。また、実際にスマートウォッチ上でモデルを動作させた場合の推論時間やバッテリー消費量を評価し、提案手法がオンデバイスの推論においても有効であることを示した。

Detection of Motion on a Trampoline with a Smartwatch

SATOSHI KOBAYASHI¹ TATSUHITO HASEGAWA¹

1. はじめに

ゲームと運動の融合により継続的に運動することができ、リングフィットアドベンチャー^{*1}の売上が好調であるように、自宅での運動手段の需要は高まっている。また、生活習慣病の予防には日常生活中での適度な運動が必要であると言われている^{*2}。そのため、自分の体力に見合った運動を継続的に行うための手段は重要である。

トランポリン上で行うエアロビクスはトランポビクスと呼ばれ、トレーニングとして注目されている^{*3}。硬い床の上でのエアロビクスは激しい動きにより足腰を損傷する可能性があるが、トランポリン上で行うことで足腰への負担を軽減できる。また、トランポリンを利用した運動は平地で行う同等の運動に比べて、運動量は同等である一方で主観評価においては楽な運動であると捉えられる傾向があることが報告されている [1]。しかし、単なる健康維持を目的

とした運動では継続するための意欲を失いやすく習慣になりにくい [2]。特に、トランポリン運動は単調になりやすく継続性が乏しいと考えられる。トランポリン運動をエンターテインメント化したシステム^{*4*}や、トランポリン運動を促進するシステムを開発している研究は存在している [3] [4] [5]。しかし、これらは施設に向く必要や専用のデバイスを使用しているため専用機器の調達に手間やコストがかかり、手軽に利用することは難しい。

近年、スマートフォンやスマートウォッチの普及により様々なセンサが手軽に利用できるようになった。スマートフォンやスマートウォッチのようなスマートデバイスから得られるセンサデータを用いた研究は多数存在し、特にセンサデータから人間の行動を推定する研究が多くなされている [6] [7] [8] [9] [10]。

そこで本研究では、スマートウォッチと家庭用トランポリンを用いて、トランポリン運動を促進するシステムの開発を目指す。図 1 に提案するシステムの概要を示す。腕に装着したスマートウォッチから得られるモーションセンサデータを用いて、トランポリン上での動作を推定する。スマートウォッチは、スマートフォンと比べて装着位置が

¹ 福井大学大学院工学研究科知識社会基礎工学専攻
Graduate School of Engineering, University of Fukui

^{*1} 任天堂. リングフィットアドベンチャー. <https://www.nintendo.co.jp/ring>. (Accessed on 2021/04/27).

^{*2} 厚生労働省. 生活習慣病予防. https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/kenkou_iryuu/kenkou/seikatsu/seikatusyuukan.html. (Accessed on 2021/04/27).

^{*3} 笹川スポーツ財団. トランポビクス. https://www.ssf.or.jp/ssf_eyes/dictionary/trampobics.html. (Accessed on 2021/04/27).

^{*4} IVRC History Archive. HOP AMP. <http://ivrc.net/archive/hop-amp-2007>. (Accessed on 2021/04/27).

^{*5} ココノエ. スペースホッパーズ. <https://9ye.jp/works/spacehoppers>. (Accessed on 2021/04/27).

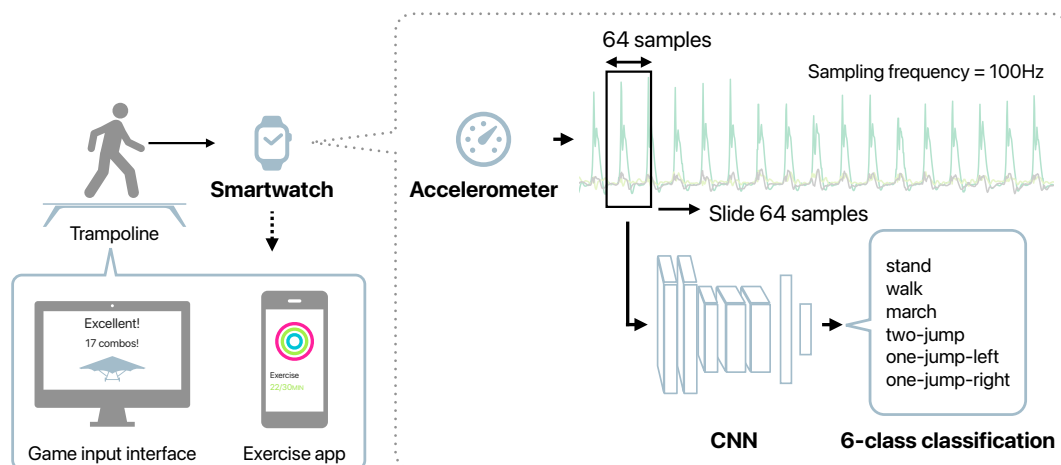


図 1: 提案するシステムの概要

決まっており固定しやすいため、トランポリン運動の妨げにならない。スマートウォッチから得られる推定結果をゲームのインターフェースとして利用したり、詳細な情報とともにエクササイズを記録したりできる。トランポリン上で行った動作でゲームを操作できるようになれば、トランポリン運動に対するモチベーションが上がり、持続性向上が見込まれる。特に本研究では、提案システムの中核である、スマートウォッチの加速度センサからトランポリン上で行った動作を推定する手法を開発し、実験により有効性を示す。

2. 関連研究と本研究の立ち位置

2.1 トランポリン運動を促進するシステム

福地ら [3] は、トランポリン運動にカメラでの自撮り機能を追加することで運動を促進するシステムを開発している。このシステムでは、トランポリンで跳んでいる状態の写真を自動で撮影するために、測距センサをトランポリン下に設置し跳躍間隔から頂点到達時間を推定している。森ら [4] は、トランポリン上での動作に合わせて VR 上の映像を変化させるシステムを開発している。このシステムでは、非接触の赤外測距センサによりトランポリンの膜面の 2 次元情報を計測している。これにより、トランポリン上での歩行運動、バランス運動、ジャンプ運動を検出している。向ら [5] は、トランポリンのゲームのインターフェースとして用いるための動作検出法を開発している。測域センサと赤外測距センサを用いた検出手法を提案し、トレジャーハンティングゲームで有効性を検証している。

トランポリンをエンタテインメント化したシステムとして、HOP AMP がある*4。これは、トランポリンの膜面下にある機械式のセンサからユーザのジャンプ状態を計測し、ジャンプの状態からトランポリンの膜面に投影している映像を変化させるようなシステムである。また、トランポリンを使ったアトラクションに、スペースホッパーがあ

る*5。これは、トランポリン上でジャンプすることでレースを勝ち抜くゲームになっている。トランポリンの下にあるセンサから着地した位置を計測しており、トランポリンの中心で高く跳ぶと加速するようなゲームになっている。

2.2 モーションセンサを用いた行動認識

加速度センサなどのモーションセンサから人間の行動を認識する研究は多く存在する。行動の認識には機械学習がよく使われており、深層学習を適用した研究も多くなされている [7] [8] [9]。Hasegawa ら [10] は、画像認識分野でよく用いられている Convolutional Neural Network (CNN) のモデル構造が行動認識においても有効に働くのかを検証している。また、OPPORTUNITY データセット [11] [12] や HASC データセット [13] などの公開されたデータセットがあるという背景もあり、センサデータを用いた行動認識分野の研究は盛んに行われている。

2.3 オンデバイス深層学習

深層学習モデルは一般に層の数を増やすことでパラメータ数が増加し、モデルの表現力が向上することが知られている。しかし、層の数を増やすことで計算コストが増加し多くの計算リソースが必要になる。画像分類の分野では MobileNet [14] や EfficientNet [15] などのパラメータ数を抑えながら高精度を達成するモデル構造が提案されている。特に MobileNet は、スマートフォンなどの計算リソースに限りのある環境で動作するモデルとして開発されている。

深層学習をアプリケーションとして提供する方法として、サーバで推論させる方法 (クラウド*6*7) とスマートフォンなどのデバイス上で推論させる方法 (オンデバイス) がある。クラウドの場合、サーバ上の潤沢な計算機資源を使っ

*6 Amazon.com Inc. Machine Learning - AWS. <https://aws.amazon.com/machine-learning>. (Accessed on 2021/04/27).

*7 Google. AI Platform. <https://cloud.google.com/prediction>. (Accessed on 2021/04/27).

て推論することが可能になるため、一般的にはサーバにモデルを実装することが多い。オンデバイスの場合、ユーザのプライバシー保護につながることやインターネットへの接続の必要性がないこと、サーバのコストがかからないことが利点として挙げられる [16]。しかし、スマートフォンはサーバと比べて計算リソースが限られているため、計算コストが小さいモデルが必要になる。

2017年以降にスマートフォンに深層学習モデルを実装するためのフレームワークとして TensorFlow Lite^{*8}や PyTorch Mobile^{*9}, Core ML^{*10}などがリリースされている。これらのフレームワークにより、推論のためのサーバを用意することなく深層学習モデルを用いたアプリの開発が可能となっている。

行動認識の分野においてもスマートフォンやエッジデバイスで動作する深層学習モデルの開発が行われている。Tengら [17] は、小さなフィルタを使った CNN モデルに対して、層ごとに個別の損失関数を用いてモデルを学習する手法を提案している。PyTorch Mobile を用いて提案モデルを Android スマートフォンに実装し、オンデバイス推論にかかる時間を評価している。Xuら [18] は、Inception モジュール [19] と Gated Recurrent Unit (GRU) を組み合わせた InnoHAR を提案し、小型の組み込みプラットフォームである MinnowBoard Turbot^{*11} にモデルを組み込んで推論時間を評価している。また、Agarwalら [20] は、エッジデバイスで動作する行動認識のための軽量な深層学習モデルを提案している。

2.4 本研究の立ち位置

以上の研究に対し、本研究の独自性は、以下の3点である。

- スマートウォッチを用いたトランポリン上での動作を検出する手法を提案する。
- スマートウォッチから得られる加速度データからどの程度まで詳細にトランポリン上の動作検出が実現できるかを検証する。
- 提案手法をスマートウォッチ上で動作するアプリケーションとして実装し、推論時間とバッテリー消費量の観点から有効性を評価する。

推定対象はトランポリン運動の効果について検証している論文を参考にし、家庭用トランポリンで実施可能な動作

である、両足立ち (ST; stand) [21], 歩行 (WL; walk) [22], 跳ねながら歩く (MR; march) [21], 両足ジャンプ (TJ; two-jump) [23] [22] [24], 片足ジャンプ (左) (LJ; one-jump-left), 片足ジャンプ (右) (RJ; one-jump-right) [23] [25] [26] の6種類とした。

3. スマートウォッチを用いたトランポリン上での動作検出手法

提案手法の概要を図1に示す。本システムは、腕に装着したスマートウォッチから得られる加速度データを入力としている。本研究では、加速度データを一定長のウィンドウに分割し CNN を用いて推定する手法を提案する。今回は、ウィンドウサイズを 64, スライド幅を 64 としてウィンドウに分割する。サンプリング周波数が 100Hz の場合、0.64 秒の観測データからトランポリン上で行った動作を予測するモデルとなる。

本研究で使用した CNN モデルを表1に示す。Conv1D は畳み込み層, MaxPooling1D はプーリング層, GAP は Global Average Pooling を示す。出力層の活性化関数には Softmax 関数を用いている。Hasegawaらの研究 [10] において、VGG16 [27] が行動認識において高い推定精度を実現するモデルであることが示されていることから、本研究では VGG16 をベースにパラメータ調整を行った。また、パラメータ数を削減するために全結合層の代わりに Global Average Pooling (GAP) [28] を用いた。パラメータ数を削減することはモデルサイズを小さくさせるだけでなく、過学習の抑制にもつながるため今回採用した。

4. 認識精度に関する有効性評価実験

4.1 データ計測

提案手法の推定精度を評価するために、データ計測実験を行った。スマートウォッチを左手首に装着した状態でトランポリン上での動作を行う。両足立ち、歩行、跳ねながら歩く、両足ジャンプ、片足ジャンプ (左)、片足ジャンプ (右) の順で 10 秒ずつ動作を行い、これを 1 セットとする。これを 1 人 5 セット行った。各セットの間は 1 分間の休憩を挟んでいる。被験者は、表2に示す 20 代の健常男性 8 名である。身長や体重の情報は自己申告であり、実際に計測した値ではない。

使用したスマートウォッチは Apple Watch SE である。データ計測にあたって、加速度データを集めるアプリケーションを開発した。このアプリケーションでは、Core Motion フレームワーク^{*12}を使用している。Core Motion では加速度データの単位を G としているため、画面を上にして机上においたとき、Z 軸には -1.0 が観測される。つ

^{*8} Google. TensorFlow Lite. <https://www.tensorflow.org/lite>. (Accessed on 2021/04/27).

^{*9} Facebook. PyTorch Mobile. <https://pytorch.org/mobile>. (Accessed on 2021/04/27).

^{*10} Apple Inc. Core ML Overview. <https://developer.apple.com/machine-learning/core-ml>. (Accessed on 2021/04/27).

^{*11} MinnowBoard Wiki. MinnowBoard Turbot. http://minnowboard.outof.biz/MinnowBoard_Turbot.html. (Accessed on 2021/04/27).

^{*12} Apple Inc. CMMotionManager. <https://developer.apple.com/documentation/coremotion/cmmotionmanager>. (Accessed on 2021/04/27).

表 1: 使用した CNN モデルの構造

Layer Type	Number of Filter	Shape of Output	Output Channels
Conv1D	16	192	16
Conv1D	16	192	16
MaxPooling1D	2	96	16
Conv1D	32	96	32
Conv1D	32	96	32
MaxPooling1D	2	48	32
Conv1D	64	48	64
Conv1D	64	48	64
Conv1D	64	48	64
MaxPooling1D	2	24	128
Conv1D	128	24	128
Conv1D	128	24	128
Conv1D	128	24	128
MaxPooling1D	2	12	128
Conv1D	128	12	128
Conv1D	128	12	128
Conv1D	128	12	128
MaxPooling1D	2	6	128
GAP	-	-	128
Softmax	-	-	6

表 2: 被験者 8 名の詳細

ID	Age	Gender	Dominant	Height [cm]	Weight [kg]
A	23	男	右	163.0	52.0
B	22	男	右	169.5	55.0
C	23	男	右	161.0	54.0
D	24	男	右	173.0	55.0
E	22	男	右	165.0	75.0
F	23	男	右	170.5	60.0
G	22	男	右	170.0	76.5
H	23	男	右	170.0	55.0

まり, $1 [G] = 9.8 [m/s^2]$ である. サンプル周波数は 100Hz とした. データ収集実験では, 同時に Apple Watch のジャイロセンサのデータとズボンの右前ポケットに格納した iPhone の加速度センサ, ジャイロセンサ, 磁気センサのデータを収集しているが, 本研究では使用していない.

4.2 ベースライン

従来手法のベースラインとして, 表 3 に示す人手で設計した特徴量 (HCF; Hand-Crafted Features) と Random Forest [29] を用いた手法と比較する. 表 3 の特徴量を 3 軸加速度データに対して抽出し Random Forest の入力とする. 今回は, スマートフォンの加速度センサからスマートフォンの所持位置推定をする研究 [30] や路面の種類を推定する研究 [31] で使用されている特徴量を採用した. 周波数成分に関しては, 加速度データをウィンドウ単位で Fast Fourier Transform (FFT) 処理し, 全周波数領域, 低周波数領域, 中周波数領域, 高周波数領域で同様の特徴量を算

表 3: 特徴量名称一覧

Domain	Feature Name
Time	平均
	絶対値の平均
	標準偏差
	絶対値の標準偏差
	最小値
	最大値
	二乗平均平方根
	第 1 四分位数
	中央値
	第 3 四分位数
	四分位範囲
	軸間の相関係数
	軸間の絶対値の相関係数
	フレーム内の初期値
	フレーム内の最終値
	動きの激しさ
	歪度
	尖度
	ゼロ交差率
	Frequency
最大値のときの周波数	
2 番目に大きい値	
2 番目に大きい値のときの周波数	
標準偏差	
第 1 四分位数	
中央値	
第 3 四分位数	
四分位範囲	
軸間の相関係数	

出している. 低周波数領域は 0~4.2Hz, 中周波数領域は 4.2~8.4Hz, 高周波数領域は 8.4~12.6Hz とした.

また, ベースラインの CNN モデルとして関連研究 [9] で提案されているシンプルな CNN モデル (以降, Li2018) と全結合層を用いたオリジナルの VGG16 (以降, VGG16) [10] を用いる.

4.3 性能評価

モデルの評価で使用する評価指標は, クラス間のデータ数が大きく偏っていないことから, 正解率 (Accuracy) を用いる. 評価手法については, Leave-One-Subject-Out Cross Validation (LOSO-CV) を用いる. LOSO-CV では, 1 人の被験者をテストデータとし, 残りの被験者を学習データとする. それをすべての被験者に対して行う交差検証である.

CNN モデルは Adam [32] で最適化を行う. 学習率は 1.0×10^{-3} とした. 損失関数は, 多クラス交差エントロピー (Categorical cross entropy) とし, ミニバッチサイズを 20, エポック数を 100 として学習を行った. CNN モ

デルは TensorFlow^{*13}を用いて作成し、性能を評価する。モデルの学習は Apple M1 と 16GB の RAM を搭載した MacBook Pro (13-inch, M1, 2020) を使用した。

4.4 実験結果

4.4.1 推定精度

表 4 に被験者ごとのトランポリン上の動作検出の Accuracy を示す。太字で表した推定精度は、同一被験者内において達成された最高推定精度を示している。また、表最下部の Avg は全被験者の推定精度の平均である。表 4 より、全被験者の平均では提案手法が最も高い推定精度を達成した。全結合層から GAP に置き換えたことで VGG16 よりも精度が向上している。しかし、被験者によっては提案手法以外の手法で最高精度を達成している。被験者 D は HCF と Random forest を用いた手法で最高精度を達成しており、CNN モデルが有効でなかったと考えられる。このことから、個人によって有効な特徴量に差異があることが分かる。そのため、全ての被験者に対して提案手法が有効であるとは言えない。しかし、平均で最高精度を達成しており、最高精度となる被験者が半数以上を占めている。このことから、トランポリン上での動作検出に提案手法は多くの場合有効であると考えられる。

4.4.2 詳細な分類可能性の検証

表 5 と表 6 に RF を用いた場合と提案手法を用いた場合の Confusion Matrix をそれぞれ示す。Recall とは、あるラベルと予測したデータのうち実際にそのラベルである割合のことである。Precision とは、あるラベルのデータのうち、そのラベルと予測された割合のことである。F-measure (F 値) は Recall と Precision の調和平均である。F 値を見ると、どちらの手法でも両足立ちと歩行は高精度で推定が可能であることが分かる。また、提案手法を用いた場合では、RF と比べてジャンプ 3 種を誤分類するケースが減少していることが分かる。特に、片足ジャンプの左右を誤分類するケースが減少している。このことから提案手法ではジャンプの分類において有効であったと考えられる。しかし、

表 4: 被験者ごとのトランポリン上の動作検出の Accuracy

Subject	RF	Li2018 [9]	VGG16 [10]	Ours
A	74.4	73.3	76.2	82.9
B	77.3	80.9	71.1	80.7
C	57.6	66.7	59.3	67.1
D	86.7	72.4	68.0	76.0
E	75.6	82.0	75.6	85.1
F	58.0	63.6	63.3	68.0
G	63.3	82.2	82.4	88.0
H	82.4	76.7	86.4	82.4
Avg	71.9	74.7	72.8	78.8

^{*13} Google. TensorFlow. <https://www.tensorflow.org>. (Accessed on 2021/04/27).

どちらの手法でも跳ねながら歩くと片足ジャンプを誤分類するケースが多く見られる。このことから、これらの推定対象を分類することは難しいと考えられる。これらの対象を分類することが難しい理由として、跳ねながら歩くという動作は左右の片足ジャンプを組み合わせた動作であるため、今回使用したウィンドウ幅では判別がつきにくかったことが考えられる。

4.4.3 提案手法の実運用に関する有効性の検証

提案手法が実運用において有効であるか議論する。他人のデータのみで学習した場合、提案手法で得られた推定精度は平均で 78.8%であった。被験者によっては推定精度が 70%程度しか得られていないため、実運用上でユーザビリティの低下が懸念される。行動認識では自己のデータが学習データに含まれている場合、推定精度が向上することが知られている。また、Core ML や TensorFlow Lite にはオンデバイスで個人特化する機能が存在している^{*14*15}。そこで、3 セット分のユーザのデータが学習に使用できる状況を想定した推定精度を検証する。

図 2 に 3 セット分のユーザのデータを学習に使用した場合の被験者ごとの推定精度を箱ひげ図で示す。各モデルの説明は以下の通りである。

- None: 他人のデータのみで訓練したモデル
- Fine-tune: 他人のデータで訓練したモデルをさらに 3 セット分の自己のデータで訓練したモデル
- Fine-tune (Dense only): 他人のデータで訓練したモデルの全結合層のみをさらに 3 セット分の自己のデータで訓練したモデル
- Included training data: 3 セット分の自己のデータを他人のデータと合わせて学習データとして訓練したモデル

Fine-tune (Dense only) は、全結合層のみの学習をサポートしている Core ML や TensorFlow Lite の個人特化を模したモデルである。全結合層のみを追加学習した Fine-tune (Dense only) では、None と同等の推定精度しか得られていない。そのため、現在の Core ML などを用いたオンデバイスの個人特化は本課題では有効であるとは言えない。しかし、将来的に畳み込み層の学習を含めた個人特化がサポートされた場合、Fine-tune の推定精度が得られる。また、個人特化をするためのデータをサーバに送り学習データに加えて再学習することで Included training data の推定精度が得られる。

^{*14} Apple Inc. Personalizing a Model with On-Device Updates - Apple Developer Documentation. https://developer.apple.com/documentation/coreml/core_ml_api/personalizing_a_model_with_on-device_updates. (Accessed on 2021/05/03).

^{*15} TensorFlow Blog. Example on-device model personalization with TensorFlow Lite. <https://blog.tensorflow.org/2019/12/example-on-device-model-personalization.html>. (Accessed on 2021/05/03).

表 5: RF による Confusion Matrix

Pre. \ Cor.	ST	WK	MR	TJ	LJ	RJ	Precision
ST	591	4	1	4	0	0	98.5
WL	7	551	39	3	0	1	91.7
MR	1	42	360	71	121	120	50.3
TJ	1	0	32	456	50	55	76.8
LJ	0	2	101	39	328	121	55.5
RJ	0	1	67	27	101	303	60.7
Recall	98.5	91.8	60.0	76.0	54.7	50.5	71.9
F-measure	98.5	91.8	54.8	76.4	55.1	55.1	

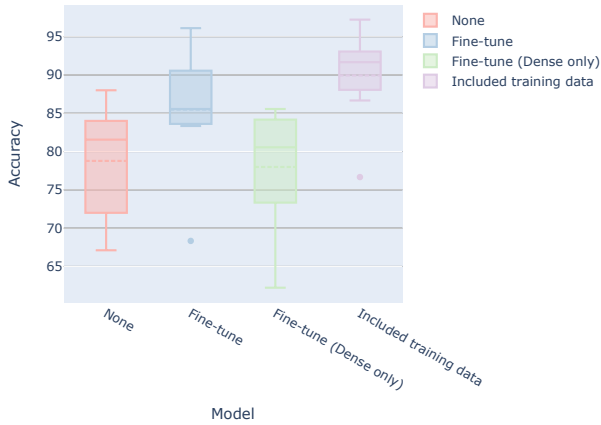


図 2: 自己のデータを学習に使用した場合の推定精度

図 3 に 3 セット分のユーザのデータを学習に使用した場合の被験者ごとの F 値を箱ひげ図で示す。他人のデータのみで訓練したモデルである None の場合、跳ねながら歩くと左右の片足ジャンプの F 値が平均 64.6% である。そのため、それらの推定対象の誤検出がユーザビリティの低下につながる。一方、Included training data の場合、MR, LJ, RJ の F 値は平均で 86.0% を達成しており、それらの推定対象の誤検知は None より少ないことが分かる。そのことから、ユーザのデータを学習データに加えて再学習できる場合、ユーザビリティが低下することは少ないと考えられる。Fine-tune の場合、None と比べて MR, LJ, RJ の F 値は向上しているが平均 76.7% である。LJ と RJ を区別することが難しいことから、それらを 1 つのラベルにすると推定精度は 88.0% になる。また、MR, LJ, RJ を 1 つのラベルとすると推定精度は 95.0% になる。そのことから、これら 3 つの推定対象を片足で跳ねるなどの 1 つのラベルとすることで実環境で誤検知の少ないシステムを実現できると考えられる。

5. オンデバイス動作に関する有効性評価実験

5.1 スマートウォッチでの動作検証方法

提案手法をスマートウォッチ上にオンデバイスで動作するように実装し、実環境下における提案手法の有効性を検証する。スマートウォッチとしては Apple Watch を使用し、モデルの実装には Core ML を用いた。Core ML とは、

表 6: 提案手法による Confusion Matrix

Pre. \ Cor.	ST	WL	MR	TJ	LJ	RJ	Precision
ST	581	3	0	3	0	0	99.0
WL	8	578	65	3	2	5	87.4
MR	1	16	379	13	95	83	64.6
TJ	4	1	9	479	32	30	86.5
LJ	0	2	88	64	422	91	63.3
RJ	0	0	59	38	50	391	72.7
Recall	97.8	96.3	63.2	79.8	70.3	65.2	78.8
F-measure	98.4	91.7	63.9	83.0	66.6	68.7	

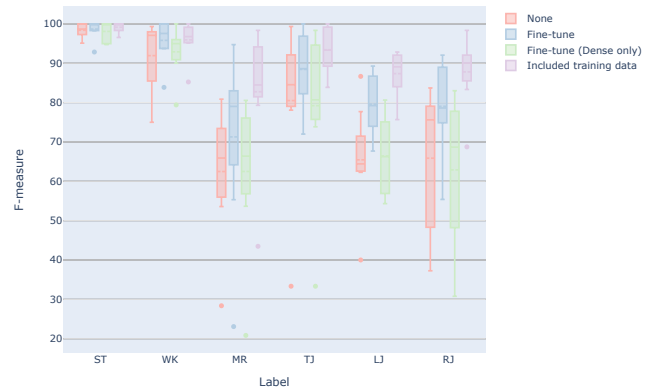


図 3: 自己のデータを学習に使用した場合の各推定対象の F 値

Apple が開発した機械学習フレームワークである。Apple のハードウェアに最適化されており、機械学習モデルによる推論処理を完全にオンデバイスで実行することができる。Core ML では mlmodel ファイルを Xcode^{*16}に取り込むだけでアプリケーションで使用するクラスのソースコードが自動生成される。アプリケーションでは、そのクラスを使用することでモデルに推論させることができる。また、TensorFlow などの Python 向けのライブラリからモデルを変換して使用することができる。Python 向けライブラリのモデルは Core ML Tools^{*17}を用いて mlmodel 形式へ変換する。

Apple Watch 上でモデルの動作を検証するために作成した watchOS App について説明する。作成したモデルを Apple Watch で使用するための流れを図 4 に示す。まず、TensorFlow で作成したモデルを Core ML Tools を用いて mlmodel 形式に変換する。Xcode に mlmodel ファイルを取り込み、自動生成されたクラスをアプリケーション内で使用する。作成したアプリケーションでは、Core Motion フレームワークを用いて加速度データを 0.01 秒間隔 (100 Hz) で収集し、3 軸 64 サンプルが収集されたらモデルで推論を行う。作成した watchOS App のスクリーンショットを図 5 に示す。ボタンをタップすると加速度データの計測が開始され、ウィンドウサイズを満たすデータが収集され

*16 Apple Inc. Xcode 12. <https://developer.apple.com/xcode/>. (Accessed on 2021/04/27).

*17 Apple Inc. Core ML Tools. <https://coremltools.readme.io/>. (Accessed on 2021/04/27).

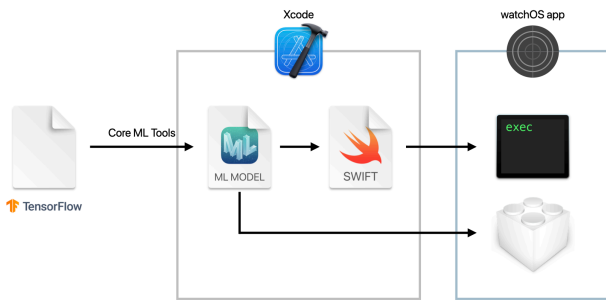


図 4: watchOS 上でモデルを使用する流れ

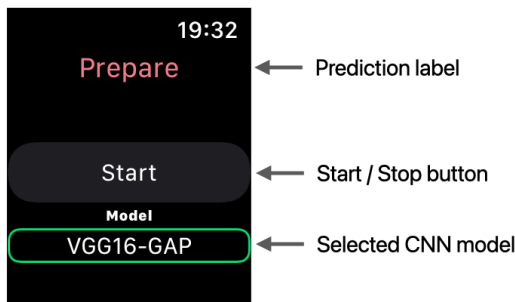


図 5: 作成した watchOS App

たらモデルで推論が行われる。

本研究では、1回の推論にかかる時間と消費電力でスマートウォッチ上でのモデルの動作を評価する。動作評価の実験について説明する。作成した watchOS App による加速度データの収集・推論を1時間行う。アプリでは推論時間と各推論後のバッテリー残量を記録している。バッテリー残量は100%に充電された状態から実験を開始する。推論にかかる時間の平均と加速度データの収集とモデルの推論を1時間動作させた場合のバッテリー消費量をモデルの評価とする。検証は、watchOS 7.3.2 を搭載した Apple Watch SE で実施した。

5.1.1 結果

表7に、各 CNN モデルの mlmodel ファイルのサイズ、平均推論時間、バッテリー消費量を示す。モデルサイズは提案手法の CNN モデルが最も小さいという結果であった。これは、畳み込み層のフィルタサイズが小さいことと GAP によってパラメータ数が抑えられていることによるものである。一方、特徴抽出部は同じ構造である VGG16 が最もモデルサイズが大きい。これは、全結合層を使用していることによりパラメータ数が提案手法よりも多いためである。また、推論時間は Li2018 が最も小さい。これは層数が少ないため推論に必要な処理が少ないためであると考えられる。しかし、どのモデルにおいても推論に必要な加速度データを収集するのにかかる時間 (64 ミリ秒) よりも小さいため、Apple Watch 上で推論することは現実的であると言える。また、加速度データの収集と推論におけるバッテリー消費量は、1時間使用した場合で最大でも7%程度しか減少しない。1日に行うトランポリン運動の時間を

表 7: モデルサイズと Apple Watch 上でのパフォーマンス

Model	Size [MB]	Time [ms/window]	Battery [%/h]
Li2018	4.0	0.81	7
VGG16	8.6	1.78	5
Ours	1.3	1.25	5

多く見積もって1時間であるとしても、バッテリー容量に制限のある Apple Watch でも普段使用できる範囲であると考えられる。しかし、これらのパフォーマンスの結果は Apple Watch と Core ML を用いて得られたものであることに留意する必要がある。Core ML 自体は Apple のデバイスに最適化されたローレベルの基盤である Accelerate, BNNS, Metal Performance Shaders の上に構築されている^{*18}。したがって、他のプラットフォームでは今回得られたパフォーマンスを実現できない可能性があるため、他のプラットフォームでも検証することが今後の課題である。

6. まとめ

本研究では、スマートウォッチから得られる加速度データからトランポリン上で行った動作を検出する手法を開発し評価した。提案手法として、行動認識で有効な CNN モデルであった VGG16 をベースとした手法を提案し、その有効性を検証した。GAP を用いることでベースの VGG16 より6%高い精度を達成した。また、どこまで詳細に動作を分類できるかを検証したところ、跳ねながら歩くとジャンプ3種の分類が難しく、特に跳ねながら歩くと片足ジャンプを誤分類しやすいことが分かった。加えて、実際に Apple Watch 上で CNN モデルを動作させた場合の性能を評価したところ、提案手法はモデルサイズが小さく推論にかかる時間やバッテリー消費量も現実的であることが分かった。今後は、本研究で使用しなかったセンサの併用や動的なウィンドウ幅を用いた手法などを検討し、それにより高精度な分類が可能かを検証していきたい。また、Apple Watch 以外のプラットフォームでの性能評価もしていきたい。

謝辞 本研究は公益財団法人科学技術融合振興財団の補助金助成によるものである。ここに謝意を示す。

参考文献

- [1] 三浦孝仁, 鈴木俊哉, 松本美和子, 宅野栄子. ミニ・トランポリンを用いたエアロビック・ジャンプ運動法の検討. 体力科学, Vol. 35, No. 6, p. 561, 1986.
- [2] 山地啓司. 体力向上のための運動プログラム実施中の途中脱落率とプログラム実施率. 体育の科学, Vol. 38, pp. 607-612, 1998.
- [3] 福地健太郎, 助台良之, 大野悠人, 三輪聡哉, 大場洋哉. 自分撮りによる競創を取り入れたトランポリン運動の促進シス

^{*18} Apple Inc. Core ML - Apple Developer Documentation. <https://developer.apple.com/documentation/coreml>. (Accessed on 2021/04/29).

- テム. 情報処理学会論文誌, Vol. 58, No. 5, pp. 1003–1013, 2017.
- [4] 森博志, 白鳥和人, 星野准一. トランポリンインタフェースを用いたウェルネスエンタテインメントシステム. 日本バーチャルリアリティ学会論文誌, Vol. 15, No. 3, pp. 369–378, 2010.
- [5] 向健次, 中津留義樹, 星野准一. 健康運動支援のためのトランポリンインタフェースの動作検出法. 情報処理学会インタラクシオン 2012, pp. 193–198, 2012.
- [6] J.R. Kwapisz, G.M. Weiss, and S.A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, Vol. 12, pp. 78–82, 2010.
- [7] J. Yang. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on interactive multimedia for consumer electronics*, pp. 1–10. ACM, 2009.
- [8] H. Gjoreski, J. Bizjak, M. Gjoreski, and M. Gams. Comparing Deep and Classical Machine Learning Methods for Human Activity Recognition using Wrist Accelerometer. In *Proceedings of 25th International Joint Conference on Artificial Intelligence*, pp. 1–7, 2016.
- [9] F. Li, K. Shirahama, M. A. Nisar, and L. Köping. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. *Sensors*, Vol. 18, No. 679, pp. 1–22, 2019.
- [10] T. Hasegawa and M. Koshino. Representation learning by convolutional neural network in activity recognition on smartphone sensing. In *Proceedings of the 2nd International Conference on Computational Intelligence and Intelligent Systems*, pp. 99–104. ACM, 2019.
- [11] D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkel, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. Millán. Collecting complex activity datasets in highly rich networked sensor environments. In *Proceedings of 7th International Conference on Networked Sensing System*, pp. 233–240, 2010.
- [12] P. Lukowicz, G. Pirkel, D. Bannach, F. Wagner, A. Calatroni, K. Foerster, T. Holleczeck, M. Rossi, D. Roggen, G. Tröster, J. Doppler, C. Holzmaan, A. Riener, A. Ferscha, and R. Chavarriaga. Recording a Complex, Multi Modal Activity Data Set for Context Recognition. In *Proceedings of 23th International Conference on Architecture of Computing Systems 2010*, pp. 1–6, 2010.
- [13] N. Kawaguchi, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Nura, S. Inoue, Y. Kawahara, Y. Sumi, and N. Nishio. HASC Challenge: gathering large scale human activity corpus for the real-world activity understandings. In *Proceedings of the 2nd augmented human international conference*, pp. 1–5. ACM, 2011.
- [14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [15] M. Tan and Q. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114, 2019.
- [16] M. Xu, J. Liu, Y. Liu, F. X. Lin, Y. Liu, and X. Liu. A First Look at Deep Learning Apps on Smartphones. In *Proceedings of WWW '19: The World Wide Web Conference*, pp. 2125–2136, 2019.
- [17] Y. Tang, Q. Teng, L. Zhang, F. Min, and J. He. Layer-Wise Training Convolutional Neural Networks With Smaller Filters for Human Activity Recognition Using Wearable Sensors. *IEEE Sensors Journal*, Vol. 21, No. 1, pp. 581–592, 2021.
- [18] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan. ImmoHAR: A Deep Neural Network for Complex Human Activity Recognition. *IEEE Access*, Vol. 7, pp. 9893–9902, 2019.
- [19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [20] P. Agarwal and M. Alam. A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. *Procedia Computer Science*, Vol. 167, pp. 2364–2373, 2020.
- [21] W. Kanchanasamut and P. Pensrl. Effects of weight-bearing exercise on a mini-trampoline on foot mobility, plantar pressure and sensation of diabetic neuropathic feet; a preliminary study. *Diabetic Foot & Ankle*, Vol. 8, No. 1, 2017.
- [22] 松村夫美子. トランポピクス (実技研修, 平成3年度大学体育指導者研修会 (中央研修会)). 大学体育, Vol. 18, No. 2, pp. 76–79, 1991.
- [23] B. Aalizadeh, H. Mohammadzadeh, A. Khazani, and A. Dadras. Effect of a Trampoline Exercise on the Anthropometric Measures and Motor Performance of Adolescent Students. *International Journal of Preventive Medicine*, Vol. 7, , 2016.
- [24] 高橋珠実, 石田滉弥, 梨本侑季, 塩原茂, 新井淑弘. トランポリン運動が男子大学生の体幹および下肢筋力に与える影響. 群馬大学教育実践研究, Vol. 36, pp. 101–106, 2019.
- [25] P. Giagazoglou, D. Kokaridas, M. Sidiropoulou, A. Patsiaouras, C. Karra, and K. Neofotistou. Effects of a trampoline exercise intervention on motor performance and balance ability of children with intellectual disabilities. *Research in Developmental Disabilities*, Vol. 34, No. 9, pp. 2701–2707, 2013.
- [26] F. A. Aragão, K. Karamanidis, M. A. Vaz, and A. Arampatzis. Mini-trampoline exercise related to mechanisms of dynamic stability improves the ability to regain balance in elderly. *Journal of Electromyography and Kinesiology*, Vol. 21, No. 1, pp. 512–518, 2011.
- [27] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of International Conference on Learning Representations*, 2015.
- [28] M. Lin, Q. Chen, and S. Yan. Network In Network. In *Proceedings of International Conference on Learning Representation*, 2014.
- [29] L. Breiman. Random Forests. *Machine Learning*, Vol. 45, No. 1, pp. 5–32, 2001.
- [30] 長谷川達人, 越野亮. 深層学習を用いた歩行時におけるスマートフォンの所持位置推定. 情報処理学会論文誌, Vol. 57, No. 10, pp. 2186–2196, 2016.
- [31] S. Kobayashi and T. Hasegawa. Smartphone-Based Estimation of Sidewalk Surface Type via Deep Learning. *Sensors and Materials*, Vol. 33, No. 1, pp. 35–51, 2021.
- [32] D. P. Kingma and J. L. Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.