エッジ，クラウド間分散処理に向けた動作識別手法の検討

高崎智香子† 竹房あつ子†† 中田 秀基††† 小口 正人†

† お茶の水女子大学 〒 112–8610 東京都文京区大塚 2–1–1
†† 国立情報学研究所 〒 101–8430 東京都千代田区一ツ橋 2–1–2
††† 産業技術総合研究所 〒 305–8560 茨城県つくば市梅園 1–1–1
E-mail: †chikako@ogl.is.ocha.ac.jp, ††takefusa@nii.ac.jp, †††hide-nakada@aist.go.jp †oguchi@is.ocha.ac.jp

あらまし センサ機器やクラウドコンピューティングの普及により，一般家庭で取得，蓄積した動画像が子供やお年寄りの見守りサービスや防犯対策，セキュリティに活用されるようになってきた．しかし，家庭のセンサで取得した動画像をリアルタイムに機械学習を用いて解析するには，データ転送量と解析計算量が課題となる．我々は，センサ側で姿勢推定ライブラリ OpenPose を使用して動画像から関節の特徴量データを抽出して転送し，クラウドでその特徴量データのみを用いて機械学習による動作識別を行うことで，処理遅延やプライバシの問題に対処する分散処理手法を提案している．実験では，STAIR Actions データセットの 3 カテゴリの動作識別が 80%以上の精度で行えること，センサからクラウドへのデータ転送量を大幅に削減できることを確認した．しかし，3 カテゴリの識別では汎用性が低く，過学習の改善も課題となっていた．本研究では，より多様な動作識別を活用する実アプリケーションへの応用を目指し，同データセットの全 100 カテゴリを用いた識別を行った．実験から，LSTM を用いることで最も高い精度を得ることができることを確認した．

キーワード 機械学習，深層学習，分散処理，OpenPose，Keras

A Study on Action Recognition Method using NN in Distributed System over Edge and Cloud

Chikako TAKASAKI†, Atsuko TAKEFUSA††, Hidemoto NAKADA†††, and Masato OGUCHI†

† Ochanomizu University 2-1-1 Otsuka,Bunkyo-Ku, Tokyo 112–8610, Japan
†† National Institute of Informatics 2-1-1 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
††† National Institute of Advanced Industrial Science and Technology (AIST) 1-1-1 Umezono, Tsukuba, Ibaraki 305–8560, Japan
E-mail: †chikako@ogl.is.ocha.ac.jp, ††takefusa@nii.ac.jp, †††hide-nakada@aist.go.jp †oguchi@is.ocha.ac.jp

1. はじめに

センサ機器やクラウドコンピューティングの普及により，一般家庭でライフログを取得，蓄積し，子供やお年寄り，ペットの見守りサービスや防犯対策，セキュリティに活用されるようになってきた．M. Mohammad ら [1] は，スマートホームやスマートシティ，ヘルスケア，農業分野などの Internet of Things (IoT) デバイス (センサ) から生成される大量のストリームデータを機械学習をで分析する様々な分野のアプリケーションを紹介し，大規模ストリームデータの分析とアプリケーションの目的の達成には機械学習を用いることが有望だと述べている．このように家庭のセンサで取得した動画像をリアルタイムに機械学習を用いて解析するにはデータサイズと解析計算量が大きいため，サーバやストレージを用いてデータの分析や蓄積を行う必要がある．

動画像解析のように大規模なストリームデータを扱う高負荷な処理を行うには高性能なサーバやストレージが必要となり，それらをセンサを配備している環境に設置するのは難しい．クラウドでは潤沢な計算資源を利用することができるが，センサとクラウド間のネットワーク帯域が限られており，アプリケーションが期待する処理遅延で解析を行うのは非常に困難である．そのため，計算の一部をセンサ側，もしくはセンサ側に近いエッジデバイスで処理をするエッジコンピューティング [2] やフォグコンピューティング [3] と呼ばれる分散処理が有効で

ある．

センサ側で前処理を行う利点は，処理遅延の他にもプライバシの確保，通信コストの削減の点も挙げられる．家庭内で撮影された個人が特定できるような動画像をそのままクラウドで収集，蓄積すると，アプリケーション利用者のプライバシを侵害する恐れがある．また，センサとクラウド間の通信にモバイル網を利用している場合は，転送量に従って課金されるため，できるだけ転送量を削減する必要がある．よって，センサ側での前処理により動画像から特徴量を抽出してデータ量を削減した後，クラウドでその特徴量データを収集して解析することでこれらの問題に対処できる．しかしながら，前処理によりもとの動画像データに含まれていた情報量が大幅に失われてしまうため，特徴量のみでどの程度の精度で学習や推論ができるのか明らかでない．

我々は，センサ側で姿勢推定ライブラリ OpenPose [4] [5] [6] [7] を使用して動画像から関節の特徴量データを抽出し，クラウドでその特徴量データのみを用いて機械学習による動作識別を行うことで，処理遅延やプライバシの問題に対処する分散処理手法を提案している [8]．STAIR Actions [9] データセットのうち，3 カテゴリを用いた識別を行い， 80%以上の精度で識別可能であること，センサからクラウドへのデータ転送量を大幅に削減できることを確認した．しかし，3 カテゴリの識別では汎用性が低く，過学習の改善も課題となっていた．

本研究では，より多様な動作識別を活用した実アプリケーションへ応用を目指し，同データセットの全 100 カテゴリを用いた識別を行う．また，リアルタイム動作識別の実現可能性を調査するため，センサ・クラウドにおける処理時間を測定し，STAIR Actions データセットを 3D ResNet [10] でファインチューニングしたモデルを用いた識別実験と比較する．ResNet は，画像解析に多用される Convolutional Neural Network(CNN) を深く重ねることを可能にしたモデルであり，ResNet を動画へ応用したモデルが 3D ResNet である．動作識別モデルには，深層学習フレームワーク Keras [11] で構築した Neural Network(NN) モデルと Long short-term memory(LSTM) モデルを用いた．LSTM を用いることで最も高い精度を得ることができ，センサ，クラウド間分散処理によりセンサからクラウドへのデータ転送量を大幅に削減できることを確認した．

2. 背　　景

本研究では，図 1 のようなシステムを想定している．一般家庭に設置されたカメラで動画像を取得し，前処理で特徴量を抽出し，その特徴量をクラウドに収集して機械学習処理により動画像に含まれる動作を識別する．クラウド側では動画像や静止画を用いず，センサ側で抽出した特徴量データのみを使用して十分に動作を解析できるのか，どの機械学習手法を用いると高い精度が得られるのかを調査する．本研究では，特徴量の抽出に OpenPose，学習モデルの構築に Keras を用いた．OpenPose と Keras について以降で説明する．

2.1 OpenPose

OpenPose は，深層学習を用いて人の関節等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，カーネギーメロン大学などによって開発された．動画像や画像に含まれる人物の身体，顔，手の 135 のキーポイントを検出することが可能である．加速度センサなどの特殊センサを使わずに，カメラによる画像や動画像のみで解析できることが特徴である．また，GPU を使用することで，画像や動画像に複数の人が含まれている場合でもリアルタイムに解析できる．

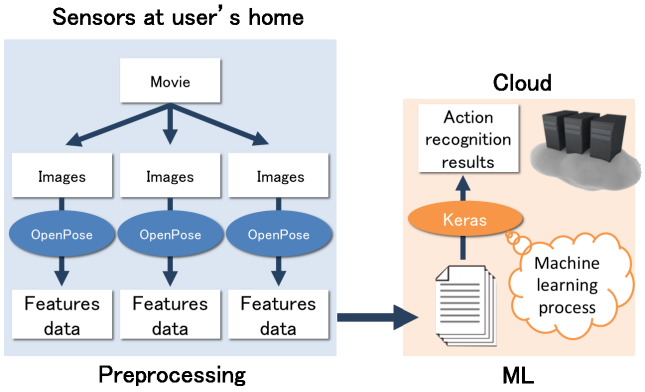

図 1　提案する分散動画像解析システム

2.2 Keras

Keras はプロジェクト Open-ended Neuro-Electronic Intelligent Robot Operating System (ONEIROS) の研究の一環として開発された，NN を実装するためのライブラリである．迅速な実験を可能にすることに重点を置いており，ユーザーフレンドリ，モジュール性，および拡張性により，容易に素早くプロトタイプの作成が可能である．また，CPU と GPU 上でシームレスに動作するため，高速な演算が可能である．Keras により簡単にモデルを記述することができる．

3. 本研究で使用する機械学習手法

OpenPose を用いて画像から抽出したキーポイントの座標データを使用し，複数の機械学習手法を用いて動作識別精度を比較する．実験では，複数静止画から取得した特徴量を用いた NN を用いた動作識別を行った上で，時系列データとしてデータを学習させるために LSTM を用いた実験を行った．データセットには，日常の動作 100 カテゴリの動画像を約 1000 ずつ集めた STAIR Actions [9] の動画像を利用する．

3.1 機械学習手法

実験では，以下の 2 つの手法で動作識別を行い，予測上位 5 カテゴリに正解カテゴリが含まれる精度 (top 5 accuracy) を測定した．

1)　NN モデル
2)　LSTM モデル

1) NN は人の神経細胞を模したモデルであり，完全結合の NN を用いた．また，NN モデルでは性能を改善するために，Dropout と Batch Normalization を以下の 3 パターンで導入し，識別精度を測定した．

1a)　NN + Dropout
1b)　NN + Batch Normalization (BN)
1c)　NN + Dropout, Batch Normalization

Dropout とは，各層のノードの一部を無効化して学習を行

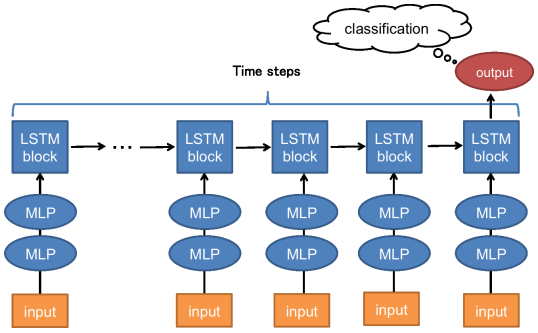

図 2　LSTM モデルの構成

い，ネットワークの自由度を強制的に小さくして汎化性能を上げることで過学習を防ぐ手法であり，本実験ではノードの 2 割を無効化して学習を行った．Batch Normalization とは，入力されるバッチの平均と分散を計算して正規化を行い，スケールとシフトで調整をすることで学習の精度と速度を向上させる手法である．

また，特徴量の前後関係をより長時間考慮して実験を行うために，RNN(Recurrent Neural Network) の拡張である 2) LSTM を用いて実験した．まず，RNN とは再帰型 NN と呼ばれるモデルであり，文章など連続的な情報を利用できるという利点がある．前の時間に計算された情報を記憶しておき，後の計算でこれらの情報を使用して学習を行うことができるが，長期記憶ができないという欠点がある．LSTM はこの欠点を解消し，データの長期依存を学習可能にした手法である．

本研究では，現段階では 10〜30 ステップでの学習を行うため，RNN による学習で十分依存関係を考慮できる可能性があるが，今後，動画像から取得する画像の枚数について考慮していく予定であるため，より長期の依存関係を学習可能な LSTM を使用した．LSTM モデルの構成を図 2 に示す．各画像から取得した特徴量を 2 層の全結合の NN(MLP) で学習した後，その結果を LSTM に時間ステップごとの入力として与え，最後のステップの出力を用いて動作のカテゴリ分類を行う．実験では，時間ステップ数に合わせた静止画を取得し，OpenPose によって抽出したキーポイントデータを各ステップの入力として与えた．また，過学習を防止するために Dropout を導入した．LSTM の Dropout には，入力の線形変換で無効化するノードの割合を表す Dropout と，再帰の線形変換で無効化するノードの割合を表す Recurrent Dropout があり，ノードの無効化率を 2 割に設定し，以下の 3 パターンで精度を測定した．

2a)　LSTM + Dropout
2b)　LSTM + Recurrent Dropout
2c)　LSTM + Dropout, Recurrent Dropout

3.2　使用データ

STAIR Actions データセットの各動画像から，等間隔に複数の静止画を抽出した．その後，各静止画に対して OpenPose を用いて 25 のキーポイントの画像上の x，y 座標を取得して特徴量 50 のデータを取得し，データセットを作成した．1) NN で使用するデータセットでは，各動画像から 0.1 秒間隔と 0.3 秒間隔の 2 パターンで 10 枚の静止画を取得した．各静止画の特徴量を時系列順に並べて，合計 500 の特徴量を 1 入力データとして使用した．2) LSTM で使用するデータセットでは，各モデルの時間ステップ数に合わせて 10，20，30 枚の静止画を取得し，各静止画の特徴量を各ステップの入力として与えた．10 枚の静止画を取得する際は，1) NN と同様に 0.1 秒間隔と 0.3 秒間隔の 2 パターンでデータセットを作成し，20 枚と 30 枚の場合には 0.1 秒間隔で静止画を取得した．OpenPose によって抽出したキーポイントの例を図 3 に，この画像から取得した座標値データの一部を図 4 に示す．各データセットで使用した静止画の枚数と取得間隔，データ数は表 1 の通りで，このうち 7 割を学習データ，3 割を正解データとして使用した．

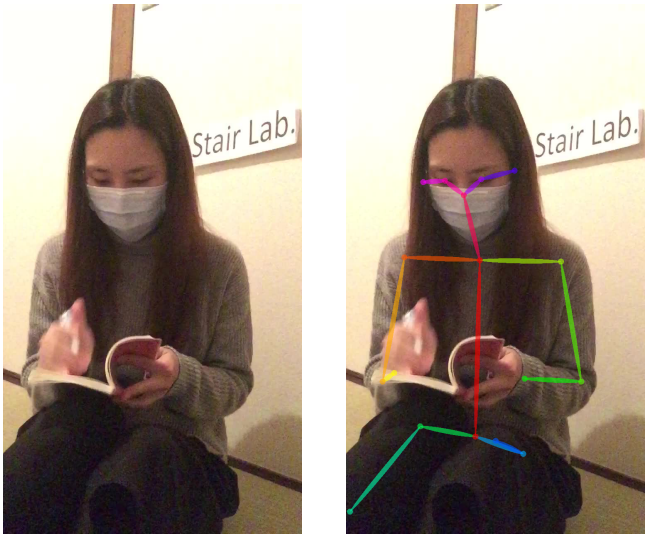

図 3　OpenPose によって取得したキーポイント


```
"people":[{"pose_keypoints_2d":[
283.201,461.222,0.818301,321.728,618.22,
0.751273,140.097,611.302,0.643794,
87.7155,911.437,0.400931,115.677,890.546,
0.379135,517.083,621.779,0.651862,
565.945,911.436,0.742212,429.756,904.388,
0.770721,311.119,1044.03,0.349768,
178.451,1019.65,0.359588,9.12998,1225.6,
0.103076,0,0,0,426.329,1085.95,0.265937,
360.019,1054.57,0.412622,0,0,0,237.761,
426.189,0.885139,325.128,422.815,
0.826925,185.5,429.66,0.365905,405.442,
401.993,0.704888,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0]}]
```


図 4　OpenPose によって取得した座標値の一部

4.　実　　験

NN と LSTM の複数手法を用いて 100 カテゴリの動作の識別精度を比較した．識別精度には，予測上位 5 カテゴリに正解カテゴリが含まれる精度 (top 5 accuracy) を用いた．また，3 カテゴリ識別で課題となっていた過学習の抑制効果を調査するために，Dropout を導入して実験を行った．次に，センサ・ク

表 1　データ数

機械学習手法	画像枚数	間隔	データ数
(1a) NN	10	0.1 sec	87923
(1b) NN	10	0.3 sec	96807
(2a) LSTM w/ 10 time steps	10	0.1 sec	87923
(2b) LSTM w/ 10 time steps	10	0.3 sec	96807
(2c) LSTM w/ 20 time steps	20	0.1 sec	128039
(2d) LSTM w/ 30 time steps	30	0.1 sec	85553

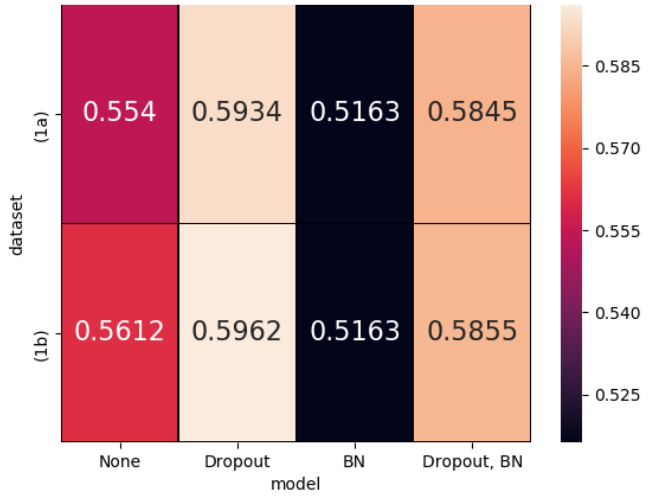

図 5　NN による動作識別のテスト精度

ラウドにおける処理時間を測定し，STAIR Actions データセットを 3D ResNet でファインチューニングしたモデルを用いた識別実験と比較することで，リアルタイムな動作識別の実現可能性を調査した．

4.1　識別精度の比較

図 5 に epoch 数を 500 に設定した際の NN による識別精度を，図 6 に epoch 数を 1200 に設定した際の LSTM による識別精度を示す．これらの図において，縦軸は学習で使用したデータセットを，横軸は学習モデルに導入した Dropout と Batch Normalization の種類を表す．図 5 から，NN では BN の導入で識別精度が悪化しているが，Dropout の導入により精度が改善されていることがわかる．また，図 6 の LSTM の結果においても Dropout の導入による精度の改善がみられた．LSTM では，時間ステップ数を 20 に設定した際の精度が良くなる傾向にあり，(2c) の時間ステップ数 30 のモデルに dropout のみを導入した結果が最も良い精度を示していた．時間ステップ数 20 の LSTM モデルで使用したデータセット (2c) はデータ数が最も多いため，識別精度が高くなったと考えられる．NN と LSTM の識別精度を比較すると，LSTM の方が良い結果が得られ，LSTM が時系列の学習に適していることが示された．

次に，Dropout による過学習の抑制効果を確認するために，最も識別精度が良くなる傾向にあった時間ステップ数 20 の LSTM モデルを用いて，学習の損失と識別精度を調査した．2)LSTM，2a)LSTM + Dropout，2b)LSTM + Recurrent Dropout，2c)LSTM + Dropout, Recurrent Dropout による学習時の損失と識別精度を図 7，図 8，図 9，図 10 に示す．これらの図において，左図は epoch 数ごとの損失の値，右図は

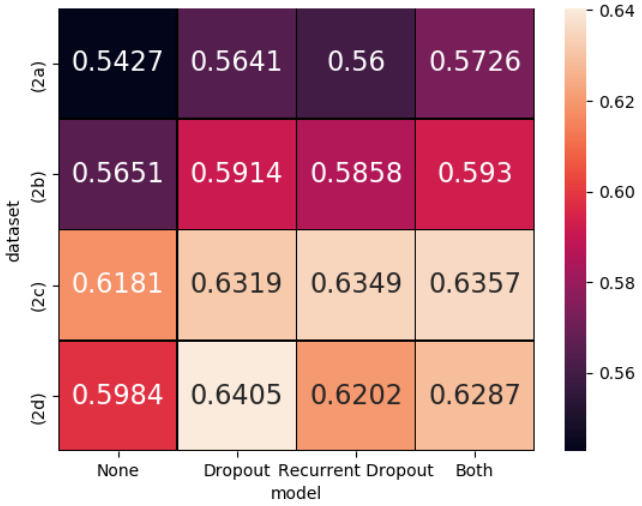

図 6　LSTM による動作識別のテスト精度

epoch 数ごとの識別精度の値を示している．横軸は epoch 数，縦軸はそれぞれ損失と識別精度の値を示しており，青色のグラフはトレーニング時の値，オレンジ色のグラフはバリデーション時の値を示している．この実験で，800 epoch では収束が十分ではなかったため 1200 epoch に設定した．図 7 から，2) ではトレーニングの損失が減少しているのに対してバリデーションの損失は増加しており，過学習の傾向が確認できる．一方，図 8，9，10 から，2a)〜2c) ではトレーニングとバリデーションの損失の差が小さく，過学習を抑制できていることがわかる．また，1200 epoch の学習後に測定したバリデーション精度は，2) では識別精度が 0.618 であったが，2a)，2b)，2c) では識別精度がそれぞれ 0.638，0.642，0.635 となり，Dropout の導入によっていずれも識別精度の改善が確認できた．3 カテゴリの識別では過学習の改善が課題となっていたが，100 カテゴリを用いて動作のバリエーションを増やすことにより，人間の動作をより汎用的に学習できていると考えられる．

4.2　学習処理時間の比較

センサ側での OpenPose の処理時間とクラウド側の推論処理時間を比較する．実験には，AI 橋渡しクラウド（AI Bridging Cloud Infrastructure，ABCI）を使用しており，用いた計算機の性能を表 2 に示す．本研究の提案システムにおいて，各データセットごとの 1000 データあたりの機械学習の推論にかかる時間を表 3 に示す．各動画の全フレームを OpenPose を用いてキーポイント抽出する時間は平均 11.13 秒，画像 1 枚あたりの平均は 0.089 秒であった．表 3 から，1000 データあたりの推論にかかる時間は 0.03 秒から 0.20 秒程度であり，OpenPose の処理時間が機械学習の推論にかかる時間より長く，センサ側での処理が重くなってしまっている．

4.3　3D ResNet との比較による提案システムの評価

次に，提案システムの識別精度や処理時間の評価を行うため，本研究で使用した STAIR Actions データセットを 3D ResNet によってファインチューニングしたモデルを用いた識別実験と比較した．CNN は，畳み込みを行うことで領域での特徴量を抽出できるため画像解析に有効であるとされるが，深く重ねるこ

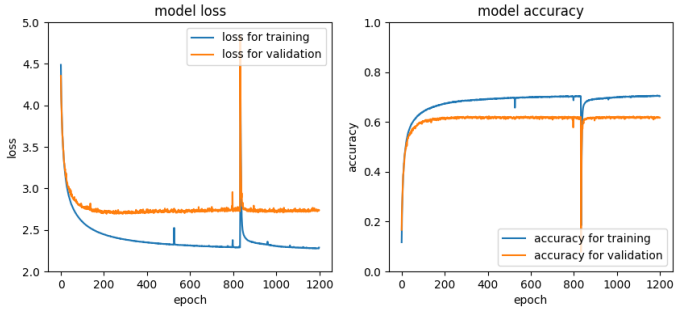

図 7　2)LSTM による学習の損失 (左図) と識別精度 (右図)

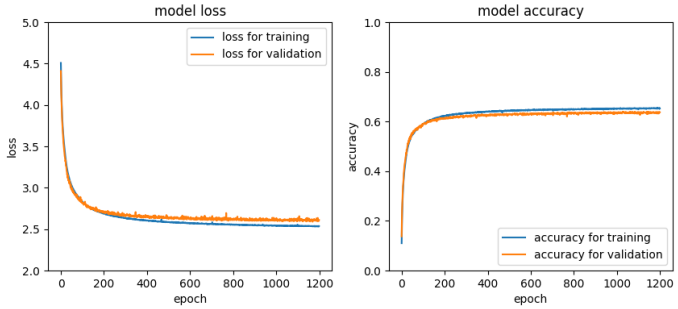

図 8　2a)LSTM + Dropout による学習の損失 (左図) と識別精度 (右図)

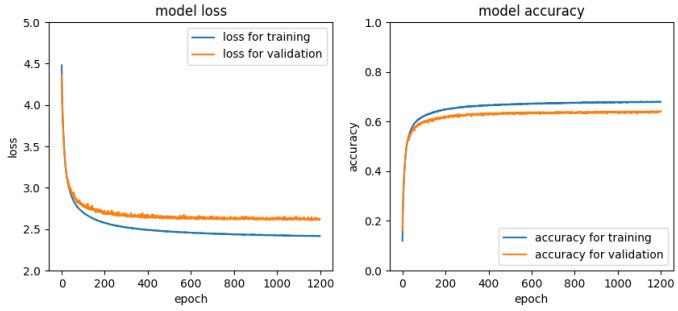

図 9　2b)LSTM + Recurrent Dropout による学習の損失 (左図) と識別精度 (右図)

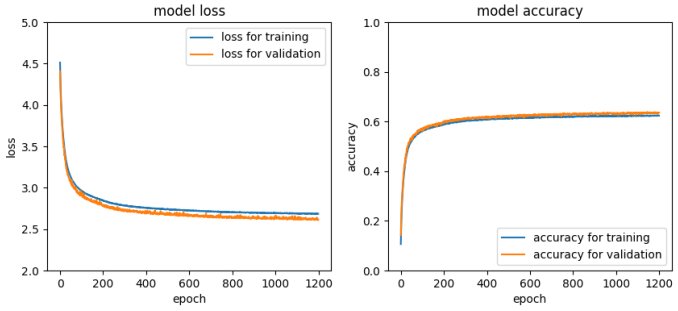

図 10　2c)LSTM + Dropout, Recurrent Dropout による学習の損失 (左図) と識別精度 (右図)

とで勾配が消失するという問題が生じる．この問題を解決することで，CNN を非常に深く重ねることを可能にしたモデルが ResNet であり，時間軸を追加することで動画へ応用したモデルが 3D ResNet である．このモデルでは，動画の全フレームを 16 フレームずつのセットに分割し，各セットの推論結果の平均

表 2　実験で使用した ABCI の計算ノード性能

OS	CentOS Linux release 7.5.1804 (Core)
CPU	Intel Xeon Gold 6148 CPU 2.6 GHz
	14 Cores (28 Threads)
GPU	NVIDIA Tesla V100 for NVLink 16GiB HBM2
Memory	384 GiB DDR4 2666 MHz RDIMM (ECC)

表 3　クラウドにおける推論にかかる時間の比較

機械学習手法	処理時間 (1000 データあたり)
(1a) NN	0.031 sec
(1b) NN	0.029 sec
(2a) LSTM w/ 10 time steps	0.091 sec
(2b) LSTM w/ 10 time steps	0.086 sec
(2c) LSTM w/ 20 time steps	0.143 sec
(2d) LSTM w/ 30 time steps	0.202 sec

を用いてカテゴリ分類を行っており，1 動画の識別にかかる平均処理時間は 7.93 秒，識別精度は 0.849 であった．提案システムと比較すると，3D ResNet を用いた識別は短時間でより高精度な識別が可能であることがわかった．しかし，OpenPose で抽出したポーズ情報を蓄積することで，動作の分類だけではなく，動作の解析を行い，人が倒れる前の異常な動作を未然に検知するなどの用途で再利用可能であるという点で有用であると考える．また，OpenPose の処理にかかる時間は画像の枚数に比例していないことがわかったため，ボトルネックになっている部分の改善により，性能改善の余地があることが確認できた．

5. 関 連 研 究

深層学習を用いた動画像の動作識別は，近年数多く研究されており，CNN や LSTM など様々な手法を用いることでより複雑な動作を高精度で識別することが可能になっている．Hara ら [10] は，動画像を入力として行動ラベルを識別するという課題に対し，2 次元の空間に 1 次元の時間空間を加えた 3 次元空間で畳み込みを行う，3D CNN ベースの様々な手法を用いた行動識別について調査した．また，Residual Network(ResNet) [12] ベースの 3D CNN を用いた行動識別による性能改善を示している．Pigou ら [13] は，動画像の一時的な情報を考慮するために特徴量を pooling して使用し，再帰と Temporal Convolution の組み合わせによる性能改善を示した．Li [14] らは，RFID で取得したデータを用いた CNN による human action のマルチクラス分類を行った．Fragkiadaki ら [15] は，LSTM の前後にエンコーダとデコーダを組み込んだモデルを使用して human pose のモーションキャプチャを生成し動作の識別，予測を行った．Ordóñez ら [16] は，CNN と LSTM の組み合わせにより加速度計，ジャイロスコープ，磁力計のデータを前処理せず融合することで識別を行った．しかし，このような処理は計算量が多く，各家庭で深層学習を用いた解析を行うのは非常に困難である．

IoT デバイスから取得したデータを用いて解析を行うためのエッジコンピューティングや，フォグコンピューティングによる様々な分散処理手法が研究されており，Li ら [17] は，IoT デバイスから取得したセンサデータを用いてディープラーニング

によって解析を行う手法を提案した．エッジノードにおける処理能力は限られているため，エッジコンピューティングを使用して IoT の深層学習 アプリケーションを構築し，パフォーマンスを最適化するためのオフロード戦略の設計や評価を行った．Tang ら [18] は，将来のスマートシティに向けて，膨大なインフラの統合をサポートする階層的な分散フォグコンピューティングアーキテクチャを提案した．Yang [19] は，4 つの典型的なフォグコンピューティングシステムのモデルとアーキテクチャの調査を行い，システム，データ，人間，最適化の 4 次元についてのデザインスペースを分析した．本研究はエッジとクラウドで処理を分散させる事によって深層学習を用いた解析をリアルタイムに行うという点で異なる．また，エッジでの前処理により抽出した動画像に含まれる人間のキーポイントの座標値のみをクラウドでの解析に使用することで，生の動画像データをクラウドに送信する通信料やプライバシの問題に対処可能である．

6. まとめと今後の予定

STAIR Actions データセットの動画像から取得した画像を OpenPose を用いて関節点の座標値に変換し，それを特徴量として複数の機械学習手法で動作の識別精度を測定した．既発表研究では，3 カテゴリのみの識別を行っていたが，より多様な動作識別を活用した実アプリケーションへ応用を目指し，STAIR Actions データセットの全 100 カテゴリを用いた識別を行った．実験から，Dropout の導入による性能改善が見られ，時間ステップ数 30 の LSTM モデルに Dropout のみを導入したときが一番良い性能を示すことがわかった．また，提案システムの識別精度や処理時間について評価するために，3D ResNet を用いた識別実験と比較した．結果から，3D ResNet を用いた方が，短時間でより高精度な識別ができることがわかったが，ポーズ情報は再利用可能であり，提案手法は有用であると考える．

本研究ではエッジとして高性能な計算機を用いて実験を行っているため，今後は家庭で利用されるエッジデバイスを用いてより実環境に近い環境での評価を行い，リアルタイム識別の実現を目指す．

謝　　辞

この成果の一部は，JSPS 科研費 JP19H04089，国立研究開発 法人新エネルギー・産業技術総合開発機構（NEDO），JST CREST JPMJCR1503 の委託業務及び，2019 年度国立情報学研究所公募型共同研究（19S0501）の助成を受けたものです．

文　　献

[1] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 4, pp. 2923–2960, 2018.

[2] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. Edge-centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review*, Vol. 45, No. 5, pp. 37–42, 2015.

[3] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16. ACM, 2012.

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[5] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pp. 1145–1153, 2017.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017.

[7] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4732, 2016.

[8] C. Takasaki, et al. A study of action recognition using pose data toward distributed processing over edge and cloud. In *Proc. the 11th IEEE International Conference on Cloud Computing Technology and Science (CloudCom2019)*, pp. 111–118, 2019.

[9] Yuya Yoshikawa, Jiaqing Lin, and Akikazu Takeuchi. Stair actions: A video dataset of everyday home actions. *arXiv preprint arXiv:1804.04326*, 2018.

[10] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proc. the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6546–6555, 2018.

[11] Keras: The Python Deep Learning library . `https://keras.io/`.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[13] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, Vol. 126, No. 2-4, pp. 430–439, 2018.

[14] Xinyu Li, Yanyi Zhang, Ivan Marsic, Aleksandra Sarcevic, and Randall S Burd. Deep learning for rfid-based activity recognition. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pp. 164–175. ACM, 2016.

[15] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4346–4354, 2015.

[16] Francisco Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, Vol. 16, No. 1, p. 115, 2016.

[17] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE Network*, Vol. 32, No. 1, pp. 96–101, 2018.

[18] Bo Tang, Zhen Chen, Gerald Hefferman, Tao Wei, Haibo He, and Qing Yang. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In *Proceedings of the ASE BigData & SocialInformatics 2015*, p. 28. ACM, 2015.

[19] Shusen Yang. Iot stream processing and analytics in the fog. *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 21–27, 2017.