「マルチメディア，分散，協調とモバイル
(DICOMO2019)シンポジウム」 令和元年7月

姿勢推定と RNN を用いた動画動作識別手法の調査

高崎 智香子[1]　竹房 あつ子[2]　中田 秀基[3]　小口 正人[1]

概要：防犯カメラなどの動画像データが活用されるようになってきたが，動画像解析に要する通信量や計算量，プライバシーに関する問題が介在している．また，ディープラーニング技術が画像認識や音声認識を始めとする様々な分野に応用されているが，正確な認識処理を行うためには大量のデータの収集，処理が必要となるため，リアルタイムに解析するのは非常に困難である．本研究では，動画像をリアルタイムに解析し，動作の識別を行うことを目標として，姿勢推定ライブラリ OpenPose とディープラーニングフレームワーク Keras を用いた機械学習手法について考察した．画像 1 枚から抽出した特徴量のみを使用した学習では，約 80%の精度で動作を識別することが可能であることがわかった．次に，同じ動画から取得した 10 枚の画像の時系列を考慮した特徴量データを使用して動作の識別精度を測定したところ，画像 1 枚の識別と比較して識別精度は低下した．また，より長い時系列を考慮した学習を行うために LSTM による学習を行い，時間ステップ数や LSTM のノード数，dropout の導入有無を変化させて動作識別精度を比較したところ，精度が最も良く約 83%に改善することができた．

A Study on Action Recognition Method with Estimated Pose by using RNN

Chikako TAKASAKI[1]　Atsuko TAKEFUSA[2]　Hidemoto NAKADA[3]　Masato OGUCHI[1]

1. はじめに

カメラやセンサ等の発達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，活用されるようになってきた．しかし，取得した動画像はデータサイズと解析計算量が大きく，サーバやストレージを一般家庭に設置して処理するのは難しい．リアルタイムに機械学習を用いて動画像を解析するためには，センサ側での前処理により特徴量を維持したままデータ量を削減した後，クラウド側に集約して処理することが望ましい．

本研究では，深層学習を用いて人の関節情報を抽出する姿勢推定ライブラリ OpenPose[1][2][3][4] を使用し，動画像から取得した関節の特徴量データから，複数の機械学習手法を用いて動作識別を行った際の認識精度を比較した．また，ディープラーニングフレームワーク Keras[5] で構築した NN モデルを用いた動作識別の性能改善を図った．画像 1 枚から抽出した特徴量のみを使用した動作の識別と，同じ動画から取得した 10 枚の画像の時系列を考慮した特徴量データを使用した動作の識別を行い，各手法において 80%以上の精度で動作を識別することが可能であることがわかった．次に，より長い時間の依存関係を学習させるために LSTM を用いた実験を行った．LSTM のノード数や時間ステップ数，dropout の導入有無について変化させ識別精度を比較したところ，時系列を考慮したデータを使用した他の手法による識別よりも高い精度を得ることができた．しかしながら，過学習抑制手法の導入による改善が十分ではないため，実験結果をもとに精度を向上させる手法について考察する．

[1] お茶の水女子大学
[2] 国立情報学研究所
[3] 産業技術総合研究所

2. 背景

2.1 提案する動画像解析システムの概要と目的

本研究では，図 1 のようなシステムを想定している．各一般家庭に設置されたカメラやセンサで取得した動画像から特徴量抽出を行い，その特徴量をクラウドに収集し機械学習処理を行うことで動画に含まれる動作を識別する．クラウド側では動画や静止画を用いず，センサ側で抽出した

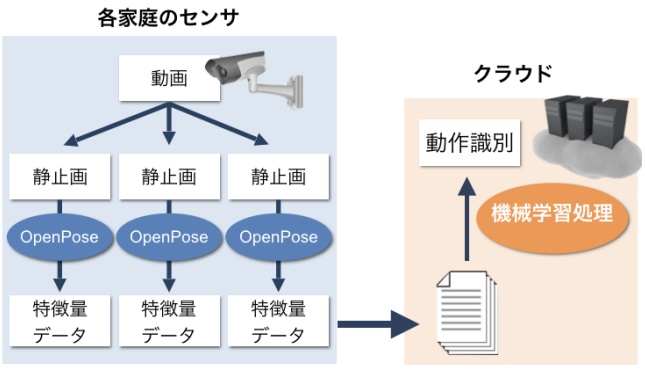

図 1　提案する動画像解析システム

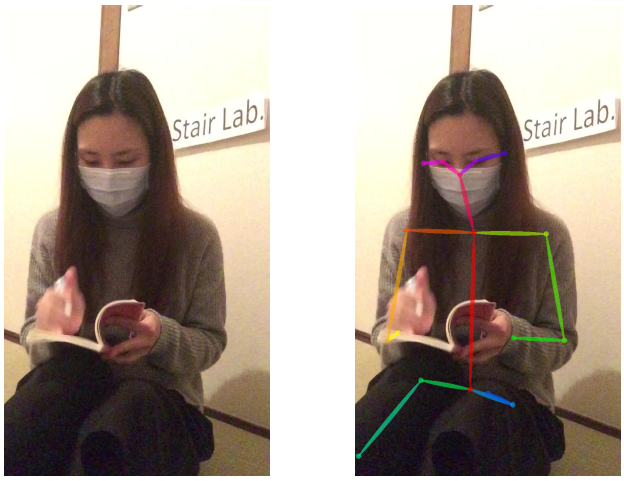

図 2　OpenPose によって取得したキーポイント


```
"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 3　OpenPose によって取得した座標値の一部

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のが本研究の目的である．

2.2　関連技術

2.2.1　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．動画や画像に含まれる人物の身体だけでなく，顔や手の 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．

2.2.2　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，ニューラルネットワークを実装するためのライブラリである．バックエンドとして TensorFlow や Theano，Microsoft Cognitive Toolkit をサポートしており，迅速な実験を可能にすることに重点を置いている．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，畳み込みやリカレント及びその組み合わせによるニューラルネットワークにも対応可能なことが特徴である．Keras の最大の利点として，非常に簡単にネットワークモデルを記述でき，迅速な実験が可能であるという点が挙げられる．

3.　本研究で使用する機械学習手法

本稿では，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て，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，(1)1 枚の静止画による動作識別と (2) 時系列を考慮した複数静止画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 カテゴリの動画を約 1000 ずつ集めた STAIR Actions[6] から取得した画像を利用する．

3.1　使用データ

STAIR Actions データセットのうち，writing，reading newspaper，bowing カテゴリの各動画から 1 秒間分の動画を切り出し，0.1 秒間隔で 1 動画につき 10 枚の静止画を抽出した．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 のデータを作成した．OpenPose によって抽出したキーポイントの例を図 2 に，この画像から取得した座標値データの一部を図 3 に示す．各カテゴリのデータ数は表 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．このとき，同じ動画内の画像が学習データと正解データに分かれないように振り分けた．上記のデータを 2 点間の近さを確率分布で表現し次元圧縮を行う手法である t-SNE[7] を用いて 2 次元に圧縮し，可視化した様子を図 4 に示す．

表 1　STAIR Actions の各カテゴリのデータ数

カテゴリ	(1) データ数	(2) データ数
writing	6470	647
reading newspapaer	8840	884
bowing	11230	1123

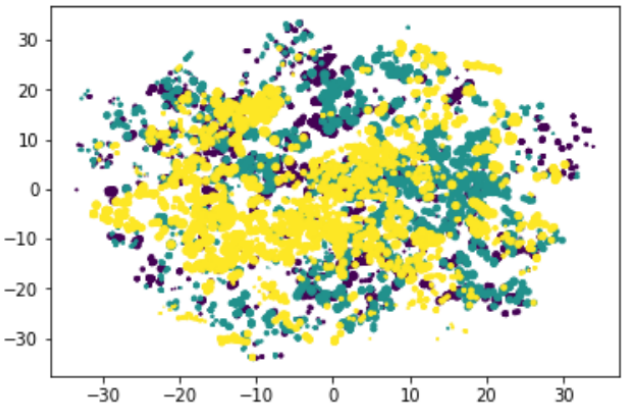

図 4　(1) 画像 1 枚のデータの分散

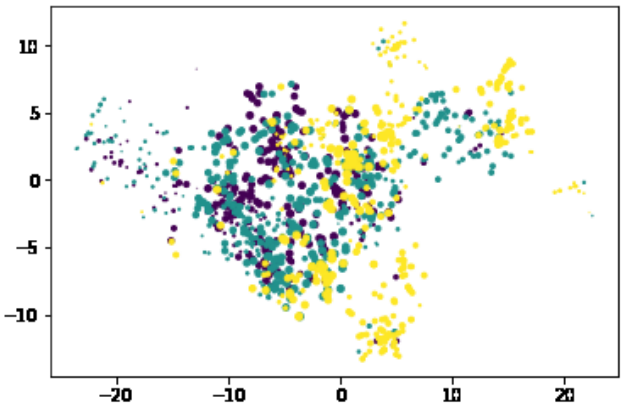

図 5　(2) 時系列を考慮したデータの分散

この図で，紫色は writing，緑色は reading newspaper，黄色は bowing カテゴリのデータを表しており，各カテゴリのデータが分散していることがわかる．

次に，時系列を考慮したストリームデータとして動作の識別を行うために，各動画から取得した画像 10 枚の 50 の特徴量を，1 枚目から時系列順に並べて特徴量 500 のデータを作成した．各カテゴリのデータ数は，表 1 の 10 分の 1 になっており，t-SNE を用いて可視化した様子は図 5 の通りである．この図で，紫色は writing，緑色は reading newspaper，黄色は bowing カテゴリのデータを表している．図 4 と比較して，データ数の違いを考慮してもカテゴリごとにまとまりが見られ，動作の特徴が現れていると考えられる．

3.2　機械学習手法

各実験では，以下の 4 つの手法で動作の認識精度を測定した．

(1) ロジスティック回帰
(2) ランダムフォレスト
(3) SVM (Support Vector Machine)
(4) Keras で構築した NN モデル

ロジスティック回帰はロジスティック関数に回帰させてクラスに属する確率を出力し，ランダムフォレストは複数の決定木の各予測結果の多数決により結果を決定するモデルである．SVM はカーネル関数を用いて射影した高次元空間のマージンを最大化するように最適化するモデルで，本実験ではカーネル関数に RBF を使用した．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パラメータ調節を行った．

次に，NN に対して 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．

(4a) Dropout
(4b) Batch Normalization (BN)
(4c) Dropout と Batch Normalization

Dropout とは，各層のノードの一部を無効化して学習を行い，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．

最後に，特徴量の前後関係をより長い時間考慮して実験を行うために，Recurrent neural network (RNN) の拡張である Long short-term memory (LSTM) を用いて実験した．まず，RNN とは再帰型ニューラルネットワーク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 CEC・入力ゲート・出力ゲート，忘却ゲート，覗き穴結合という 3 ステップの機能を導入することによってこの欠点を解消し，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．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 がある．それぞれについて 2 割のノードを無効化した際と，入力・再帰共に 2 割のノードを無効化した際の精度を測定した．

表 2　各手法による動作の識別精度

	training	validation
1) ロジスティック回帰	0.688	0.640
2) ランダムフォレスト	1.000	0.786
3) SVM	1.000	0.454
4) NN	1.000	0.828
4a) NN w/ Dropout	0.987	0.820
4b) NN w/ BN	1.000	0.842
4c) NN w/ Dropout, BN	0.970	0.813

表 3　ロジスティック回帰，ランダムフォレスト，SVM で最適化したパラメータ

手法	パラメータ	値
1) ロジスティック回帰	C	0.001
2) ランダムフォレスト	bootstrap	false
	criterion	entropy
	max_depth	none
	max features	10
	min_samples_leaf	1
	min_samples_split	3
	n_estimators	100
3) SVM	C	10
	gamma	0.0001
4) NN	中間層の総数	3
	中間層のノード数	50
	epoch 数	1600
	活性化関数	ReLU

4. 実験

4.1 画像 1 枚のデータによる動作識別

まず，データセット (1) 画像 1 枚のデータを使用した際の結果について説明する．1 枚の画像から抽出した特徴量データを使用した際の各手法による動作識別精度の測定結果を表 2 に示す．この表で，ロジスティック回帰，ランダムフォレスト，SVM は，交差検証を用いた GridSearch を行い，最も精度がよかった場合の精度を表しており，表 3 のようにパラメータを設定した．ロジスティック回帰における C は正則化の強さを表し，C が大きくなるほど正則化が弱まることを示す．ランダムフォレストのパラメータには決定木構築時に bootstrap サンプリングを行うかどうかを表す bootstrap，決定木のデータ分割基準となる criterion，決定木の最大の深さと葉の数を設定する max_depth と max_features，葉の構成とノードの分割に必要な最小のサンプル数を表す min_samples_leaf と min_samples_split，複数決定木の精度を測定し多数決を行うバギングに使用する決定木の数を示す n_estimator を設定した．SVM では誤分類を許容する程度を示す C，境界の複雑さを表す gamma を設定した．NN は，ノード数 50 の中間層を 3 層，epoch 数を 1600 に設定し，分割数 3 で交差検証を行った際の平均の精度を示している．

表 2 の 1) から 4) を比較すると，最も高い精度を示しているのは NN で，8 割以上の動作を識別できていた．また，NN の学習における損失を図 6 に，識別精度を図 7 に示す．青はトレーニング，オレンジ色はバリデーションの結果である．グラフから，epoch 数が増えるにつれてトレーニングの損失が 0 に収束しているのに対し，バリデーションの損失は増加しており，過学習が生じていることが分かる．

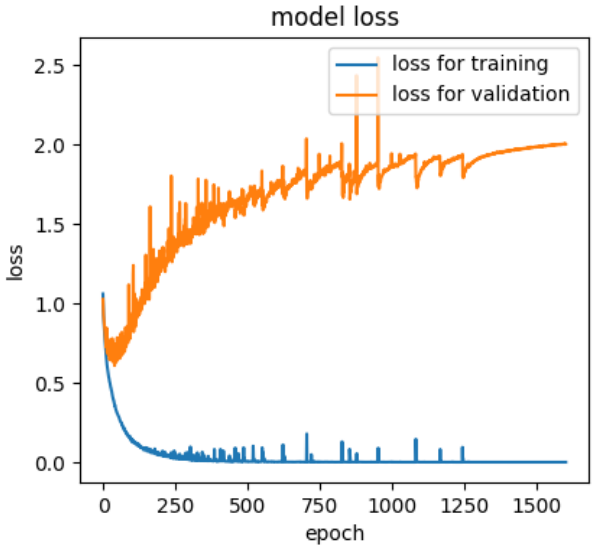

図 6　(4) NN による学習時の損失

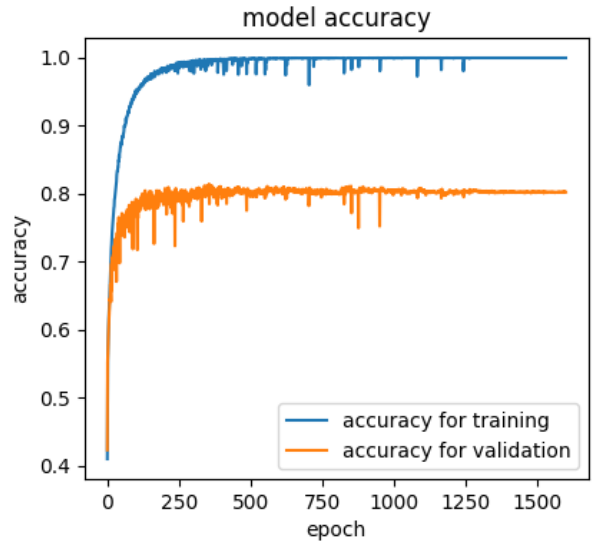

図 7　(4) NN による学習時の識別精度

次に，NN モデルのパラメータを最適化するために，中間層の層数とノード数の変化させて識別精度を測定した．図 8 は，中間層の層数を 3〜6，ノード数を 50, 75, 100, 125 と変化させた際に交差検証を用いて GridSearch を行い，認識精度の測定を行った結果をヒートマップで示している．結果から，最も精度が高いのは中間層の層数を 5，ノード数を 75 に設定した場合で，精度は 0.834 となった．また，層数が 3〜4，ノード数が 100 以上の場合に精度が良くなる傾向が見られたため，ノード数の検証範囲を絞って更に細かいパラメータを用いた実験を行うことで，性能改善が見

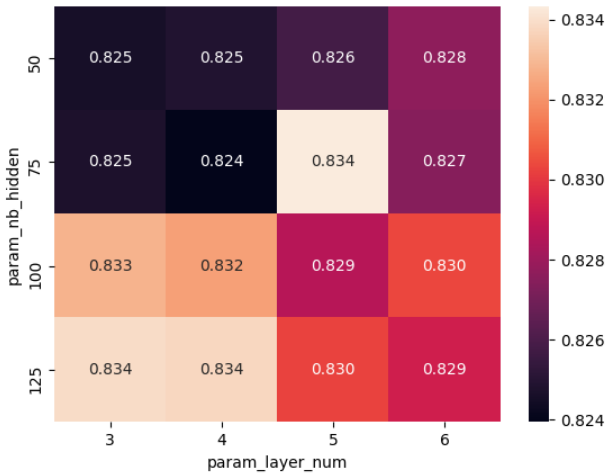

図 8　中間層の層数とノード数による動作識別精度の比較

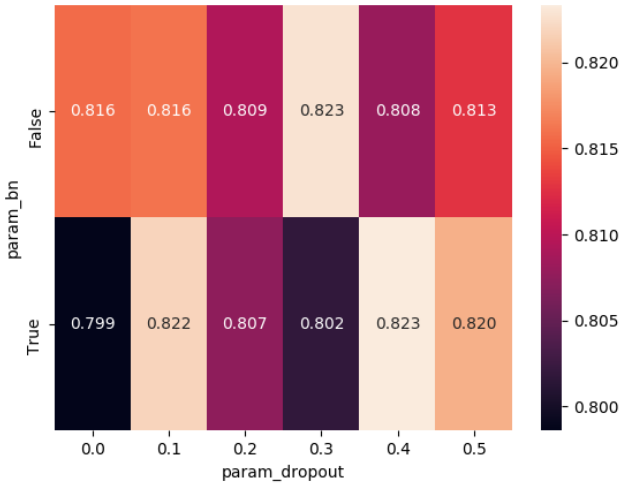

図 9　Dropout の無効化ノードの割合に関する比較

表 4　時系列を考慮したデータを使用した際の各手法による動作の識別精度

	training	validation
1) ロジスティック回帰	0.869	0.580
2) ランダムフォレスト	1.000	0.828
3) SVM	1.000	0.440
4) NN	0.976	0.748
4a) NN w/ Dropout	0.999	0.800
4b) NN w/ BN	0.999	0.813
4c) NN w/ Dropout, BN	0.987	0.765

込めると考えられる．

上記の結果を踏まえ，最も精度が高くなった中間層の層数 5，ノード数 75 に設定した NN モデルに，Dropout と Batch Normalization のそれぞれを導入する場合としない場合について交差検証を用いた GridSearch を行った．測定した認識精度を比較したヒートマップを図 9 に示す．この図において，縦軸が True の場合は Batch Normalization を導入したことを示し，False は導入していないことを示す．横軸が 0.0 の場合は Dropout を導入していないことを示し，それ以外の場合は 0.1〜0.5 の割合でノードを無効化して学習を行っていることを示す．結果から，無効化率 4 割の Dropout と Batch Normalization を導入した場合と，無効化率 3 割の Dropout を導入し，Batch Normalization を導入しない場合に識別精度が 0.823 となっているが，小数点以下 4 桁目以降で差が出ており，無効化率 4 割の Dropout と Batch Normalization を導入した場合は 0.8233，無効化率 3 割の Dropout のみを導入した場合では 0.8228 であった．よって，中間層を 5 層，ノード数を 75 に設定した場合に最も精度が高くなるのは，無効化率 4 割の Dropout と Batch Normalization を導入した場合であることがわかった．

以上の結果から，実験で得られた識別精度は十分でなく，より細かくノード数や Dropout の無効化率を設定して精度を比較することで，識別精度の向上が見込めることがわかった．

表 5　時系列を考慮したデータを使用した際のロジスティック回帰，ランダムフォレスト，SVM で最適化したパラメータ

手法	パラメータ	値
1) ロジスティック回帰	C	0.001
2) ランダムフォレスト	bootstrap	false
	criterion	entropy
	max_depth	none
	max_features	10
	min_samples_leaf	1
	min_samples_split	2
	estimators	300
3) SVM	C	1
	gamma	0.0001
4) NN	中間層の総数	3
	中間層のノード数	500
	epoch 数	1600
	活性化関数	ReLU

4.2　(2) 画像 10 枚の時系列を考慮したデータによる動作識別

次に，データセット (2) 画像 10 枚の時系列を考慮したデータを使用した際の結果について説明する．同じ動画から取得した 10 枚の画像から抽出した特徴量を時系列順に並べたデータを使用した際の，各手法による動作識別精度の測定結果を表 4 に示す．また，ロジスティック回帰，ランダムフォレスト，SVM で交差検証を用いた GridSearch を行い，最も精度のよかったパラメータは表 5 のようになった．NN は，ノード数 500 の中間層を 3 層，epoch 数を 1600 に設定した際の精度を示しており，過学習防止のために無効化率 2 割の Dropout，Batch Normalization とその両方を導入した際の精度も示している．ランダムフォレストによる識別精度が最も高く，0.828 となり，NN では Dropout，Batch Normalization，その両方を導入したいずれの場合においても識別精度を向上させることができた．

次に，中間層の層数とノード数を最適化するために，

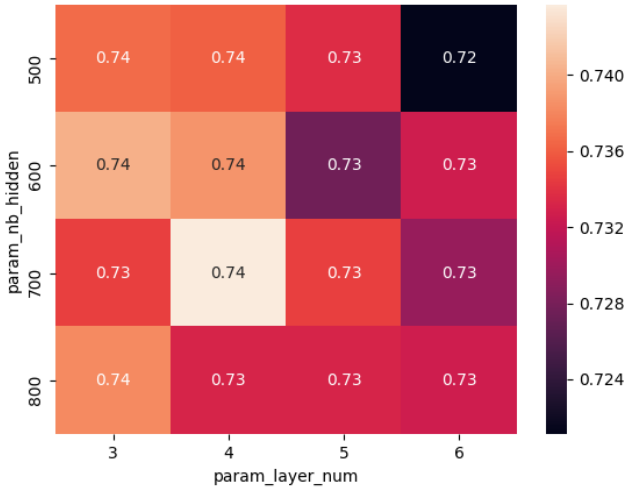

図 10　時系列を考慮したデータを使用した際の中間層の層数とノード数による動作識別精度の比較

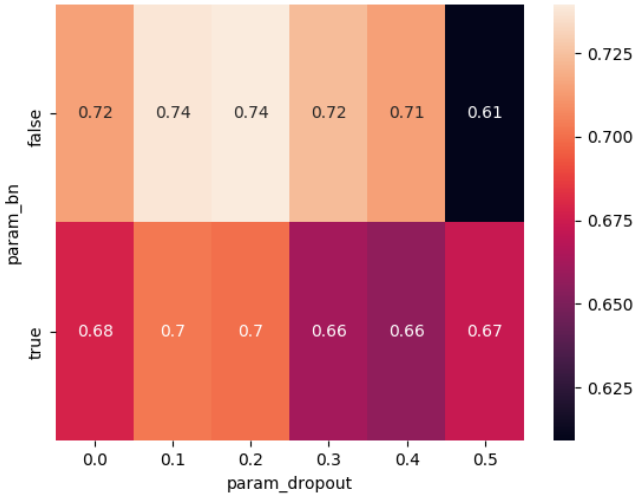

図 11　時系列を考慮したデータを使用した際の Dropout の無効化ノードの割合に関する比較

層数を 3〜6, ノード数を 500，600，700，800 と変化させて GridSearch を行ったところ，図 10 が得られた．中間層 4 層，ノード数 700 に設定したときに 0.744 と最も精度が良くなったため，この NN モデルを用いて Dropout と Batch Normalization の導入有無について GridSearch を行った結果を図 11 に示す．結果から，無効化率 0.1 と 0.2 の Dropout のみを導入した場合に識別精度が 0.74 となっているが，小数点以下 4 桁目までで比較すると，中間層の層数を 4 層，ノード数を 700 に設定した NN で最も識別制度が高くなるのは Dropout のみを無効化率 0.2 で導入した場合であった．また，Batch Normalization を導入せず，Dropout のノード無効化率を 0.0〜0.2 の範囲で変化させた場合に精度が高くなりやすいという傾向が見られたため，この範囲でより細かいパラメータ調整することで性能改善が見込めると考える．

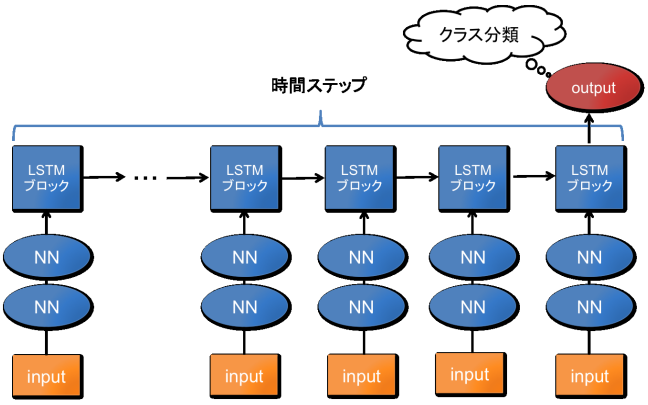

図 12　LSTM モデルの構成

表 6　LSTM による動作識別精度

時間ステップ数	ノード数	training	validation
10	50	1.0	0.802
	100	1.0	0.780
20	50	1.0	0.773
	100	1.0	0.765
30	50	1.0	0.819
	100	1.0	0.829

4.3　LSTM を用いた動作識別

LSTM モデルの構成を図 12 に示す．各画像から取得した特徴量を 2 層の全結合の NN 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．時間ステップ数を 10, 20, 30 と変化させ，ノード数を 50, 100，epoch 数を 1600 に設定した際の識別精度を表 6 に示す．NN による学習と同様に過学習の傾向が見られたため，Dropout を導入した．LSTM のノード数が 50 のとき，100 のときのそれぞれについて dropout のみ，recurrent_dropout のみ，dropout と recurrent_dropout の両方を無効化率 2 割に設定した際の，各時間ステップ数ごとの識別精度を表したヒートマップを図 13，図 14 に示す．このとき，epoch 数 1600 では収束が十分ではなかったため 3000 に設定した．図からノード数 50，100 ともに入力のみ dropout を設定した時が精度が良くなる傾向が見られた．しかしながら，dropout の導入による過学習の抑制が十分ではないため，使用データの正規化やオーギュメンテーションを行うことで改善できると考える．

4.4　考察

1 枚の静止画による動作識別実験と時系列を考慮した複数静止画を用いた動作識別実験を行い，いずれの実験でも 8 割以上の識別制度を得ることができた．しかしながら，本実験では中間層の層数とノード数，Dropout と Batch Normalization の導入についての 2 つのパラメータでのみ交差検証を行っているため，層数とノード数，Dropout の無効化率，BN の有無の全通りについて実験を行い精度を

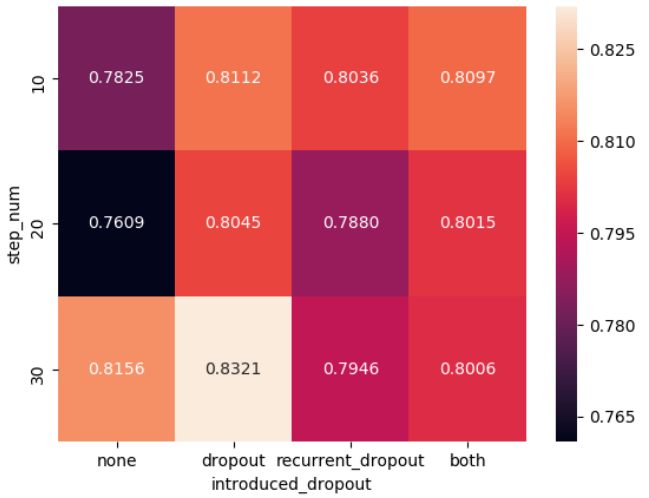

図 13　LSTM(ノード数 50) の時間ステップ数と dropout の導入有無による識別精度の比較

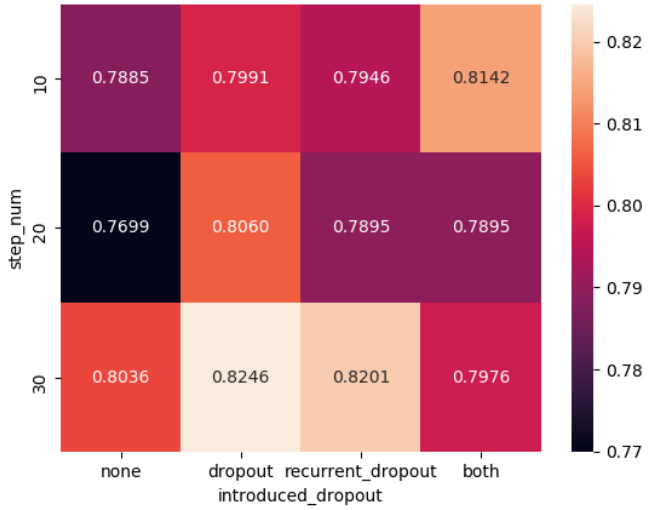

図 14　LSTM(ノード数 100) の時間ステップ数と dropout の導入有無による識別精度の比較

測定する必要があると考える．また，画像 1 枚による識別と画像 10 枚による識別を比較すると，画像 10 枚の方が動作の特徴を捉えやすいがデータ数が少ないため，トレーニングデータを増やして学習を行うことで識別精度を向上できる可能性がある．

LSTM による動作識別では，時間ステップ数が 30 の際に最も精度が良くなった．本研究で機械学習フレームワークとして使用した Keras では，LSTM の覗き穴結合に対応していないため，長期の依存関係を記憶する機能が不足している．Keras のバックエンドとして使用している TensroFlow を使用することで対応することが可能なので，今後，使用する機械学習フレームワークについても考慮していく．

NN や LSTM を用いた学習の様子から過学習の抑制が十分ではないため，データの正規化やオーギュメンテーションを行うことで改善を図る．また，使用データについて，現在はキーポイントの x，y 座標の 2 次元の特徴量を使用しているが，z 座標も合わせた 3 次元の特徴量を使用することでどの程度の精度が得られるのかについても調査する必要がある．

5.　関連研究

Hara ら [8] は，動画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．データセットとして UCF-101[9]，HMDB-51[10]，ActivityNet[11]，Kinetics[12] を用いており，Residual Network(ResNet)[13] 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．

本研究では，動画を使用せずに動画に含まれる人間のキーポイントの座標値のみを用いて十分に動作識別を行えるのかを調査し，リアルタイム解析を行うためにデータ量を削減した上で十分な識別精度を維持することを目指す．

6.　まとめと今後の予定

STAIR Actions データセットの動画から取得した画像を OpenPose を用いてキーポイントの座標値に変換した後，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．1 枚の静止画から識別する実験から，NN の精度が最も高くなることが示された．また，NN では過学習が生じていることがり，NN のパラメータ調整，Dropout と Batch Normalization の導入や，中間層の層数とノード数，Dropout と Batch Normalization の導入有無に関して交差検証を用いた GridSearch を行うことで，精度の向上が期待できることがわかった．時系列を考慮して複数静止画を用いた動作識別実験では，ランダムフォレストの精度が最もよくなった．動作の識別精度は未だ十分とは言えないが，得られた傾向をもとにパラメータの範囲を絞って識別精度を比較したり，トレーニングデータを増やすことで性能改善が見込めると考えられる．LSTM による動作識別では，時間ステップ数が 30 の時に精度が最も良くなることがわかった．また，NN と同様に過学習の傾向が見受けられたため，Dropout を導入することで改善を行った．本研究では機械学習フレームワークとして Keras を採用したが，LSTM の長期記憶を可能にする 3 つの機能のうち，覗き穴結合に対応していないことがわかったため，TensorFlow などの対応可能なフレームワークを使用することでどの程度精度に差が生じるのか調査する必要がある．

今後の課題として，過学習の改善を行っても loss が増加し続けてしまうことから，正規化手法を再考しオーギュメンテーションによりデータ量を増強することで学習の質を高めて実験を行い，3 次元の特徴量を用いて学習すると，どの程度識別精度が向上するのか調査する．また，本研究

ではリアルタイムに動作の識別処理を行うことを目標としているため，動作識別モデルをセンサ側とクラウド側のクラウド側の分散環境に実装し，動画の特徴量取得から動作識別までにかかる時間についての評価や，解析時間と認識精度のバランスを考慮した改善を行う．

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．

参考文献

[1] Z. Cao, G. Hidalgo, T. Simon, S. Wei, Y.Sheikh: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, arXiv preprint arXiv:1812.08008 (2018).

[2] Z. Cao and T. Simon and S. Wei and Y. Sheikh: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, CVPR (2017).

[3] T. Simon and H. Joo and I. Matthews and Y. Sheikh: Hand Keypoint Detection in Single Images using Multiview Bootstrapping, CVPR (2017).

[4] S. Wei and V. Ramakrishna and T Kanade and Y Sheikh: Convolutional pose machines, CVPR (2016).

[5] Chollet, François and others: Keras: The Python Deep Learning library, https://keras.io/ (2015).

[6] Y. Yoshikawa, J. Lin, A. Takeuchi: STAIR Actions: A Video Dataset of Everyday Home Actions, arXiv preprint arXiv:1804.04326 (2018).

[7] L. V. Maaten, G. E. Hinton: Visualizing Data using t-SNE, Journal of Machine Learning Research 9, 2579-2605 (2008). C. Feichtenhofer, A. Pinz, and A. Zisserman: Convolutional two-stream network fusion for video action recognition, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1933-1941 (2016). L. Wang, Y. Qiao, and X. Tang: Action recognition with trajectory-pooled deep-convolutional descriptors, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4305-4314 (2015). L. Wang, Y. Xiong, Z. Wang, and Y. Qiao: Towards good practices for very deep two-stream convnets, arXiv preprint, arXiv:1507.02159 (2015). L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool: Temporal segment networks: Towards good practices for deep action recognition, In Proceedings of the European Conference on Computer Vision (ECCV), pages 20-36 (2016).

[8] Kensho Hara, Hirokatsu Kataoka and Yutaka Satoh: Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?, arXiv preprint, arXiv:1711.09577 (2017)

[9] K. Soomro, A. Roshan Zamir, and M. Shah: UCF101: A dataset of 101 human action classes from videos in the wild, CRCV-TR-12-01 (2012).

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre: HMDB: a large video database for human motion recognition, In Proceedings of the International Conference on Computer Vision (ICCV), pages 2556-2563 (2011).

[11] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles: ActivityNet: A large-scale video benchmark for human activity understanding, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 961-970(2015).

[12] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman: The Kinetics human action video dataset, arXiv preprint, arXiv:1705.06950 (2017).

[13] K. He, X. Zhang, S. Ren, and J. Sun: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770-778 (2016).