7X-06

姿勢推定ライブラリ OpenPose を用いた機械学習による動作識別手法の比較

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 ‡国立情報学研究所 §産業技術総合研究所

1. はじめに

近年，カメラやセンサ等の発達によって一般家庭でライフログを取得することが可能になり，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で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する．

2. 実験

本稿では，OpenPose を用いて画像から抽出した関節点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．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[5] から取得した画像を利用する．

OpenPose は，深層学習を用いて人物のポーズをリアルタイムに抽出する手法であり，身体と顔と手 の 135 の関節点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，GPU などの高性能プロセッサを使用することで，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．

2.1 実験概要

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 の関節点の画像上の x，y 座標を取得して特徴量 50 のデータを作成した．各カテゴリのデータ数は表 1 の通りで，そのうち 7 割を学習データ，3 割を正解データとして使用した．

表 1: データ数

writing	5761
reading newspaper	8160
bowing	9607

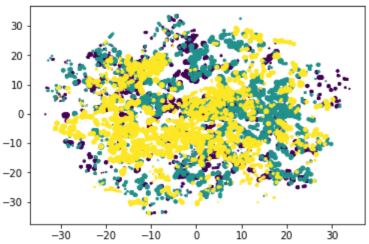

図 1: 使用データの分散

た．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[6] を用いて上記データの特徴量を 2 次元に圧縮し，可視化した様子を図 1 に示す．

本実験では，1. ロジスティック回帰，2. ランダムフォレスト，3. SVM，4. Keras[7] で作成した CNN モデルの 4 手法で動作の認識精度を比較した．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 はカーネル関数で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．CNN は人の神経細胞を模したニューラルネットワークに畳み込み処理を導入したモデルである．また，CNN 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．

Keras は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，バックエンドとして TensorFlow や Theano，Microsoft Cognitive Toolkit をサポートしている．畳み込みやリカレントなどの様々なニューラルネットにも対応可能で非常に簡単にモデルを記述できることが特徴である．このニューラルネットを多層にし たものはディープラーニングと呼ばれ，画像認識・自然言語処理・音声認識など様々な分野に応用されている．

2.2 実験結果

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，GridSearch を用いた交差検証を行い，ハイパーパラメータを最適化した精度を示しており，CNN はノード数 50 の中間層を 3 層，epoch 数を 1600 に設定した際の精度を示している．実験の結果，ランダムフォレストの精度が最も

Comparison of Motion Recognition Methods by Machine Learning Using OpenPose, a Human Pose Estimation Library

Chikako Takasaki†
Atsuko Takefusa‡
Nakada Hidemoto§
Masato Oguchi†
†Ochanomizu University
‡National Institute of Informatics
§National Institute of Advanced Industrial Science and Technology (AIST)

表 2: 各手法による動作の識別精度

	training	test
Logistic Regression	0.668	0.663
Random Forest	1.000	0.986
SVM	1.000	0.857
CNN	0.999	0.957
CNN w/ Dropout	0.970	0.942
CNN w/ BN	0.999	0.969
CNN w/ Dropout, BN	0.948	0.922

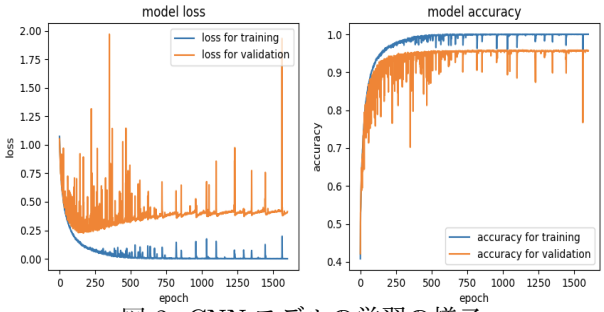

図 2: CNN モデルの学習の様子

高いことがわかった．また，上記の CNN の学習の様子 (図 2) から，過学習が生じていることが分かったため，学習時にノードの一部を無効化する Dropout と入力バッチの正規化を行う Batch Normalization (BN)，その両方を導入した際の識別精度を図 3，図 4，図 5 に示す．結果から 3 つの場合全てで過学習の抑制が確認できた．また，導入前と比較して，Batch Normalization のみを導入した場合の認識精度は約 0.969 と性能が改善されたことがわかった．

次に，CNN モデルの認識精度を改善するために中間層の層数とノード数を変化させて精度を測定した．図 6 は，中間層の層数を 3 〜6，ノード数を 50, 75, 100, 125 と変化させた際の認識精度を GridSearch を用いて交差検証を行い，測定し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 4，ノード数を 125 に設定した場合で，精度は 0.941 となった．

3. まとめと今後の予定

STAIR Actions データセット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，複数の機械学習手法で動作の識別精度を比較した．また，CNN への Dropout と Batch Normalization を導入や，中間層の層数とノード数に関して GridSearch を用いた交差検証による認識精度を比較した．

今後の課題として，各動画の画像の時系列を考慮してデータを作成し，動作識別精度の比較や CNN のパラメータ最適化を行う．

謝辞

この成果の一部は，JSPS 科研費 JP16K00177，平成 30 年度国立情報学研究所公募型共同研究，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) および JST CREST JPMJCR1503 の委託業務の結果得られたものです．

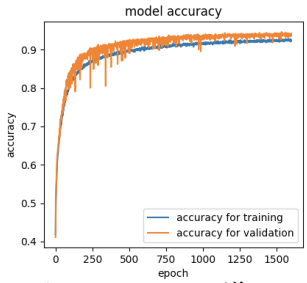

図 3: Dropout を導入

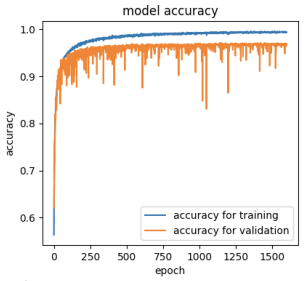

図 4: Batch Normalization を導入

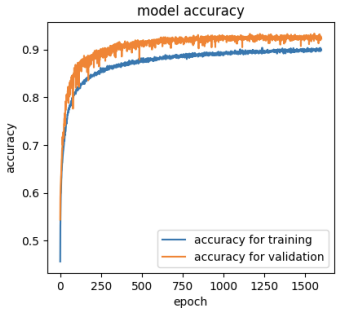

図 5: Dropout と Batch Normalization を導入

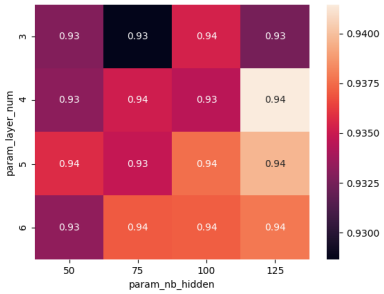

図 6: 中間層の層数とノード数による識別精度の比較

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

[6] L. V. Maaten, G. E. Hinton: Visualizing Data using t-SNE (2008).

[7] Keras: The Python Deep Learning library, https://keras.io/ .