エッジ，クラウド間分散処理を用いたリアルタイム動画像解析

理学専攻・情報科学コース　1940650　高崎 智香子

1　はじめに

子供やお年寄りの見守りサービスや防犯を目的として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で解析するには，データ量と解析計算量が課題となる．加えて，複数家庭のセンサから大量のデータがクラウドに継続して送られることが想定されるため，急激なデータの増加によるシステム負荷上昇への対応も必要である．

我々は，センサ側で姿勢推定ライブラリ OpenPose を用いて動画像データから抽出した特徴量のみを使用し，クラウド側で機械学習によって動画に含まれる動作を識別することで，プライバシや処理遅延の問題に対処する分散処理手法を提案している．クラウドにおいて，分散メッセージングシステム Apache Kafka(以降，Kafka と呼ぶ) と分散ストリーム処理フレームワーク Apache Flink(以降，Flink と呼ぶ) を用いて機械学習処理を行うシステムを構築し，リアルタイム動画像解析システムの実現を目指す．

2　分散動画像解析システム

本研究では，図 1 のようなセンサ，クラウド間分散動画像解析システムを想定している．各一般家庭に設置されたセンサのカメラで動画像を取得し，センサ端末内で前処理を行った後，メッセージングシステムを用いてクラウドにデータを収集して分散ストリーム処理基盤上で分散機械学習を行う．センサ端末で OpenPose を用いてキーポイントの座標データを抽出し，Kafka Producer から Kafka Broker に転送する．クラウドにおいて Kafka Consumer を用いて Kafka Broker からデータを受け取り，Flink の分散ストリーム処理機能を用いて機械学習の推論を行うことで動画像に含まれる動作を識別する．その後，Kafka Producer を用いて Kafka Broker に解析結果を転送する．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キーポイントデータのみを使用して解析を行う．

3　本研究で使用する機械学習手法とデータセット

3.1　機械学習手法

(1) NN モデル，(2) LSTM モデルの 2 手法で動作識別モデルを作成した．(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(MLP) を用いた．(2) LSTM は，RNN の長期記憶ができないという欠点を解消し，データの長期依存を学習可能にした．実験では，時間ステップ数を 10，20，30 と設定したモデルを使用し，過学習を抑制するため，無効化率 2 割の dropout と batch normalization(BN) を導入した．NN では，dropout のみ，BN のみ，dropout と BN の両方を導入した場合とどちらも導入しない場合の 4 パターンで識別した．LSTM の dropout には，入力の時点でノードを無効化する dropout と，再帰の時点でノードを無効化する recurrent dropout の 2 種類があり，dropout のみ，recurrent dropout のみ，dropout と recurrent dropout の両方を導入した場合とどちらも導入しない場合の 4 パターンで実験を行った．

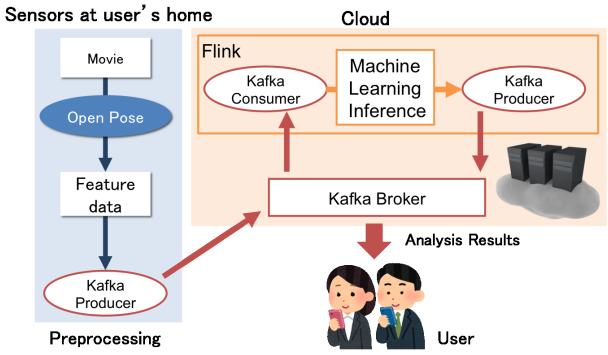

図 1: 提案する動画像解析システム

表 1: データ数

モデル	画像数	間隔	データ数
(1a) NN	10	0.1 sec	87923
(1b) NN	10	0.3 sec	96807
(2a) LSTM w/10 steps	10	0.1 sec	87923
(2b) LSTM w/10 steps	10	0.2 sec	96807
(2c) LSTM w/20 steps	20	0.1 sec	128039
(2d) LSTM w/30 steps	30	0.1 sec	85553

3.2　使用データ

OpenPose を用いて動画像の各フレームから抽出したキーポイントの座標データを使用し，機械学習による動作識別モデルを作成する．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[1] の動画像を利用する．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．各モデルで使用したデータの詳細は表 1 の通りである．(1) NN モデルでは，各画像の特徴量を時系列順に並べて使用し，(2) LSTM モデルでは，各画像の特徴量を 1 step ごとの入力として使用する．識別精度の比較実験では，各データセットを用いて動作識別を行い，予測上位 5 カテゴリに正解カテゴリが含まれる精度を測定した．スループットの測定実験では，データセット (1b) と (2b) のそれぞれを用いてあらかじめ学習した動作識別モデルを Flink プログラムで読み込んで推論で用いた．

4　実験

4.1　識別精度の比較

図 2，図 3 に NN および LSTM による識別精度を示す．図 2 から，NN では BN の導入では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 3 の結果においても dropout の導入による精度の改善が見られた．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るが，(2d) の時間ステップ数 30 のモデルに dropout のみを導入することで最も良い精度を得ることができた．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

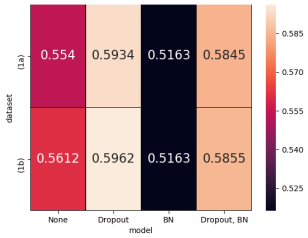

図 2: NN の識別精度

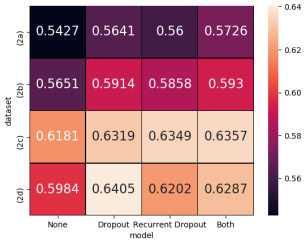

図 3: LSTM の識別精度

表 2: 推論処理時間の測定で使用した計算機の性能

OS	Ubuntu 16.04LTS
CPU	Intel(R) Xeon(R) CPU W5590 @3.33GHz
GPU	NVIDIA GeForce GTX 980
Memory	49Gbyte

ていることがわかる．

4.2 推論処理時間の比較

実験で使用した計算機の性能を表 3 に示す．OpenPose を用いたキーポイント抽出にかかる時間の 5 回平均は画像 10 枚あたり 2.14 秒で，1000 データあたりの機械学習の推論にかかる平均時間は NN モデルでは 0.082 秒，LSTM モデルでは 0.451 秒であった．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おり，リアルタイム処理を行うための適切な前処理時間を今後調査する必要がある．

4.3 提案システムのスループットの調査

実験には，表 3 に示す性能の同質の 5 つのノードを用いる．Master ノードで Kafka Producer と Kafka Broker を動作させデータ転送を行い，Flink の JobManager を動作させ，並列度に応じて 4 台の Worker ノードで動作している Flink TaskManager のスロットにタスクを分配する．各スロットでは，Kafka Consumer を動作させて Kafka Broker からデータを受け取り，キーポイントデータを用いた推論を行って動作識別を行う．その後，Kafka Producer を各スロットで動作させ，動作識別結果を Kafka Broker に転送する．

図 4，図 5 に NN および LSTM の推論を行う場合のスループットを示す．また，1 データの NN と LSTM の推論時間は，NN は平均 0.508ms，LSTM は平均 1.835ms であった．図 4 から，NN の推論には時間がかからないため，Kafka Producer からのデータ転送がボトルネックになり Flink の並列度に応じてスループットが向上しないと考えられる．(2)LSTM の推論を行う場合は，図 5 から Flink の並列度を 1 から 16 に増やすことで 2 倍程度スループットが向上していることが分かる．一方，並列度を 16 より大きくしてもスループットがあまり向上していない．これは (1) の場合同様，機械学習以外の部分がボトルネックになっていることが示唆された．

5 関連研究

近年，深層学習を利用した人間の動作識別について様々な手法が研究されている．Convolutional Neural Network (CNN) や LSTM などの様々な深層学習を使用することで，より複雑な動作を高い精度で識別することができるようになってきた．Hara ら [2] は，動画から行動を識別するため，3D Residual Network(ResNet)[3] による性能改善を示した．しかし，動作識別処理は計算量が膨大であるため，一般家庭において深層学習を使用した解析を行うことは難しい．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．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．

表 3: スループットの測定で使用した計算ノードの性能とソフトウェアバージョン

CPU	Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz 14Cores 28Threads × 2
OS	Alpine Linux 3.10.1
Memory	125Gbyte
Kafka Version	2.2.0
Flink Version	1.7.2
ZooKeeper Version (Master のみ)	3.4.14

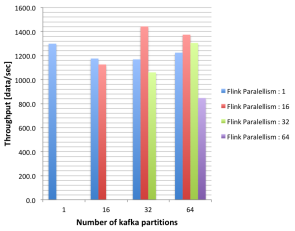

図 4: (1)NN の推論を行う場合のスループット

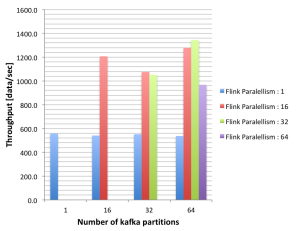

図 5: (2)LSTM の推論を行う場合のスループット

6 まとめと今後の課題

STAIR Actions データセットの動画像から取得した画像から OpenPose を用いて抽出したキーポイントデータを Kafka で収集し，クラウドにおいて Flink を用いて機械学習による動作識別処理を行うシステムを構築し，動作識別精度と解析処理性能を調査した．

今後は，家庭に配備可能なエッジデバイスとクラウド環境でのスループットや処理遅延時間について調査し，リアルタイム解析の実現を目指す．

参考文献

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

[2] 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.

[3] 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.