分散ストリーム処理フレームワークを用いた動作識別手法の検討

高崎 智香子[1] 竹房 あつ子[2] 中田 秀基[3] 小口 正人[1]

概要：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．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センサとクラウドでの分散処理手法を提案している．しかし，複数家庭のセンサから連続的に送られる大量のデータをクラウドで処理するには，急激なデータの増加によるシステム負荷上昇に耐えうる処理基盤が必要である．本研究では，大量のデータを効率よく処理可能な分散ストリーム処理基盤の構築を目指して，エッジで抽出した関節の特徴量データを Apache Kafka を用いて収集し，クラウドにおいて Apache Flink の分散ストリーム処理機能を用いて機械学習処理を行うシステムを構築し，解析スループットを調査した．実験から，解析スループットは Flink の並列度に比例して増加することが確認できた．

A Study on an Action Recognition Method using a Distributed Stream Processing Framework

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

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規模であるため，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

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

© 2020 Information Processing Society of Japan

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]．評価から，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，特徴量のみを用いた学習でも十分な精度が得られることを確認した．しかし，実際の利用環境では複数家庭のセンサから連続的に送られる大量のデータをクラウドで処理するため，急激なデータの増加によるシステム負荷上昇に耐えうる処理基盤が必要である．

本研究では，大量のデータを効率良く処理可能な処理基盤の構築を目指し，センサで抽出した関節の特徴量データを分散メッセージングシステム Apache Kafka(以降，Kafka と呼ぶ)[9] で収集し，クラウドにおいて分散ストリーム処理フレームワーク Apache Flink(以降，Flink と呼ぶ)[10] を用いて機械学習による動作識別処理を行う分散ストリーム処理基盤を構築する．構築した処理基盤を用いた実験から，解析スループットは Flink の並列度に比例して増加することが確認できた．

2. 背景

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．メッセージングシステムには，既発表研究 [11] で高スループットで画像データの収集ができることを確認している Kafka を用いる．また，分散ストリーム処理基盤には，低遅延で処理可能であることを確認している Flink を用いる [12]．図 1 では，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処理を行うことで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キーポイントデータのみを使用して解析を行う．OpenPose，Kafka，Flink について以降で説明する．

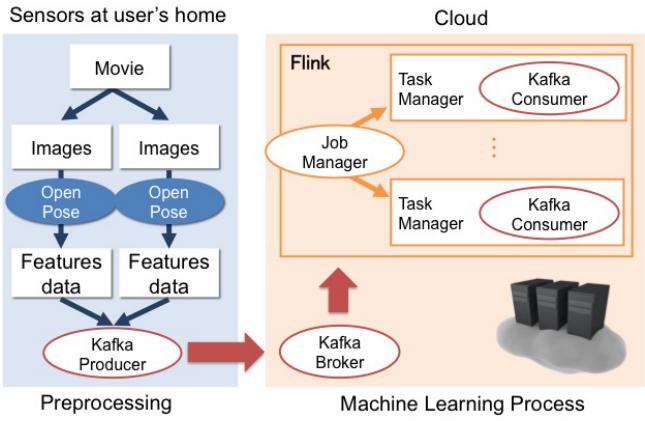

図 1　提案する動画像解析システム

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．

2.2 Apache Kafka

Kafka は高スループットでリアルタイムに大容量データを収集，配信することを目的として開発された分散メッセージングシステムである．データを保存する Broker，Broker にデータを配信する Producer，Broker からデータを参照する Consumer という 3 つのコンポーネントで構成される．Publish-Subscribe モデルを採用し，スループットを調節することが可能である．大量のメッセージを高速に処理することが可能であり，レプリケーションを行うことで耐障害性や高可用性を実現し，リアルタイムデータパイプラインやストリーミングアプリケーションの構築に広く使われている．

2.3 Apache Flink

Flink は，高スループット・低レイテンシを実現する分散ストリームとバッチデータ処理のためのフレームワークである．各処理をステートフルに扱うことで，障害発生時に処理を自動で復旧させる機能を持つため，耐障害性に優れている．無限ストリームを扱うデータストリーム API，静的データを扱うデータセット API，SQL のようなデータベース表現を扱うテーブル API を持つ．また，イベントストリームからパターンを検出するライブラリである FlinkCEP，機械学習ライブラリである FlinkML，グラフ API である Gelly を持ち，様々なユースケースに対応で

© 2020 Information Processing Society of Japan

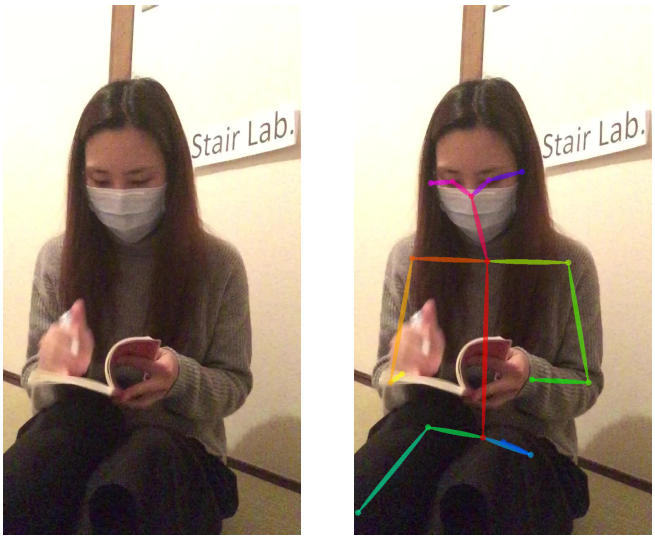

図 2　OpenPose によって取得したキーポイント


```
"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,
0,0,0,0,0,0]}]
```


図 3　OpenPose によって取得した座標値の一部

きる．

3.　本研究で使用する機械学習手法

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．動作識別モデルは予め TensorFlow を用いて作成，学習し，Flink プログラムで読み込んで推論で用いる．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 の動画像を利用する．

3.1　機械学習手法

実験では，NN(Neural Network) で動作識別モデルを作成し，動画像に含まれる動作を識別した．NN は人の神経細胞を模したモデルであり，完全結合の NN(MLP) を用いた．各動画から 10 枚の静止画を取得し，各静止画から抽出したキーポイントデータを中間層 3 層の MLP で学習させ，動画像に含まれる動作のカテゴリ分類を行う．

3.2　使用データ

STAIR Actions データセットの各動画像から，0.3 秒ごとの等間隔に 10 枚の静止画を取得した．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．各静止画の 50 のキーポイント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．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．データ数は 96807 で，このうちの 7 割を学習に，3 割をバリデーションに使用して動作識別モデルを作成した．

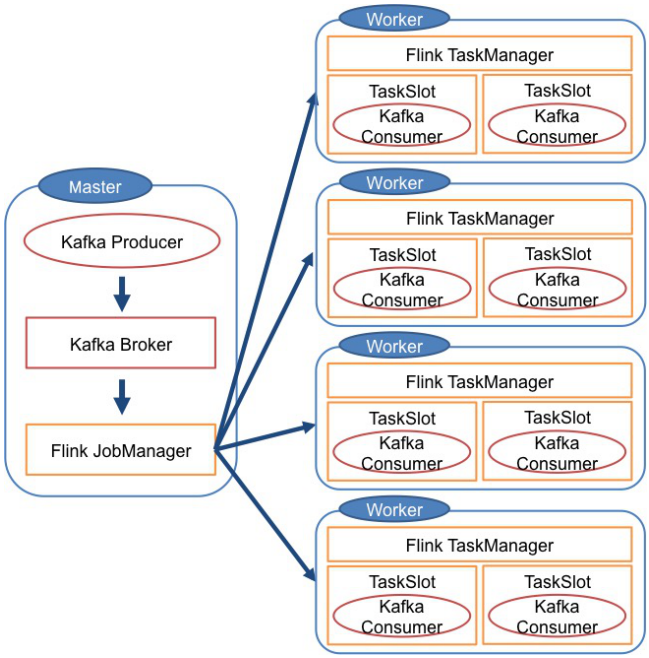

図 4　実験構成

4.　実験

提案システムの解析処理性能を調査するため，(1)Kafka Producer から Kafka Broker へのデータ転送スループット，(2)Kafka Broker から Kafka Consumer がデータを受信し，機械学習の推論を行うまでの解析スループット，(2)Kafka Broker から Kafka Consumer がデータを受信し，解析を行わない場合のスループットを測定した．

4.1　実験環境

図 4 に示す構成で，5 つのノードを用いて実験を行う．Master ノードで Kafka Producer と Kafka Broker を動作

© 2020 Information Processing Society of Japan

表 1　実験で使用した計算ノード (VM) の性能とソフトウェアバージョン

VM CPU	Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz 8Cores 16Treads
VM OS	Ubuntu 16.04.5 LTS
VM Memory	48Gbyte
Container OS	Alpine Linux 3.10.1
Kafka Version	2.2.0
Flink Version	1.7.2
ZooKeeper Version (Master のみ)	3.4.14

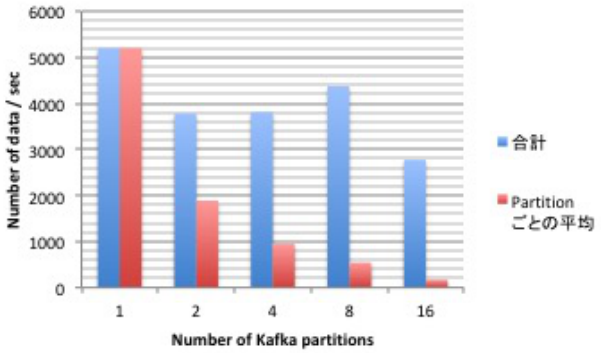

図 5　(1)Kafka Producer から Kafka Broker への転送スループット

させデータ転送を行う．加えて 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．データのカテゴリを生成する Kafka Topic においてデータの Partition 数を設定することで，複数のスロットにデータを分散し，並列処理を行うことができる．Partition 数は 1，2，4，8，16 に， 各 TaskManager のスロット数を 2 として Flink の並列度は 1，2，4，8 に設定した．実験に用いた計算機の性能とソフトウェアバージョンを表 1 に示す．Master および全ての Worker には同質のノードを用いている．各ノードは，北海道大学インタークラウドシステムの仮想サーバ (VM) を用い，学認クラウドオンデマンド構築サービス [13][14] を用いてコンテナベースで環境を構築した．Kafka はバージョン 2.2.0，Flink はバージョン 1.7.2 を使用した．本実験では，Kafka Broker は 1 ノード構成とし，レプリケーションなしとした．また，今回はスループット値の調査を目的とするため，デフォルトの at_least_once とした．

4.2　実験結果

(1)Kafka Producer から Kafka Broker へのデータ転送スループットの Partition ごとの平均を図 5，(2)Kafka Broker から Kafka Consumer がデータを受け取り機械学習の推論を行うまでの合計解析スループットを図 7，スロットごとの平均解析スループットを図 6，(3)Kafka Broker から Kafka Consumer がデータを受け取り解析を行わない場合の合計スループットを図 9，スロットごとの平均スループットを図 8 に示す．横軸が Kafka Topic の Partition 数を示し，縦軸は 1 秒間に処理したデータ数を示す．各特徴量ベクトルのデータサイズは 2KB で，処理データの総数は 10000 とした．

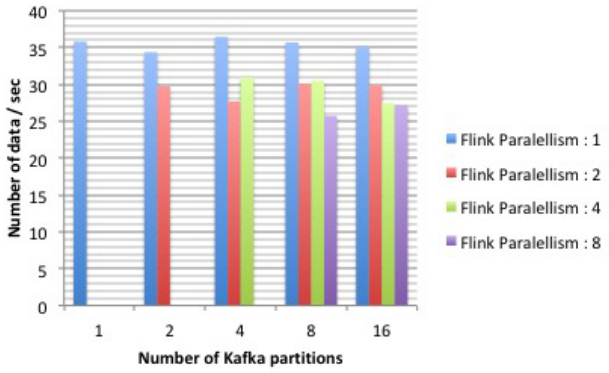

図 6　(2)Kafka Broker から Kafka Consumer がデータを受け取り機械学習の推論を行うまでのスロットごとの平均解析スループット

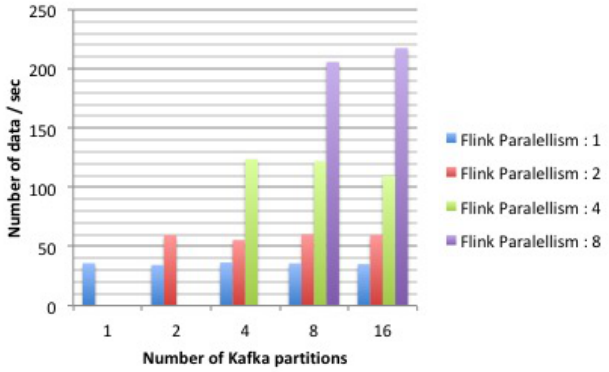

図 7　(2)Kafka Broker から Kafka Consumer がデータを受け取り機械学習の推論を行うまでの合計解析スループット

図 5 では，Producer から Broker への転送スループットは Partition 数によって大きく変化し，1 に設定した場合がもっとも効率よく転送できていることがわかる．図 6，図 7 では，Partition 数に関わらず，Flink の並列度が 1 の時が最もスロットごとの平均スループットが高くなる傾向が見られるが，合計スループットが並列度に比例して向上していることがわかる．並列度の増加によって，TaskManager 内での解析処理の負荷が上昇し，各スロットの解析スループットが落ちているが，Flink の並列処理機能を用いることでより高効率にデータを処理することが可能であると考えられる．Partition 数が 16 のときの結果が 8 のときと同程度となっていたが，今回の実験では並列度を 8 としたためであり，TaskManager のスロット数を増やすことでさら

© 2020 Information Processing Society of Japan

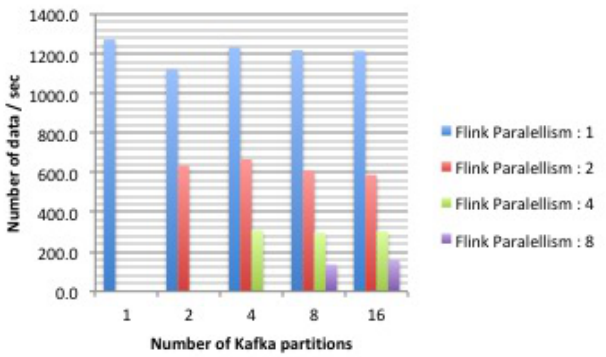

図 8 (3)Kafka Broker から Kafka Consumer がデータを受け取り解析を行わない場合のスロットごとの平均スループット

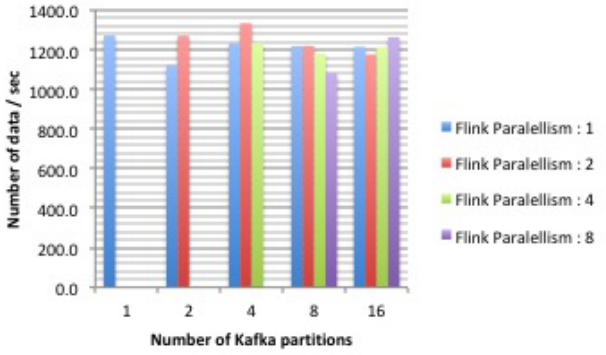

図 9 (3)Kafka Broker から Kafka Consumer がデータを受け取り解析を行わない場合の合計スループット

なる性能向上が期待できる．

図 5 の Partition ごとの平均スループット，図 6 の比較では，Kafka Producer から Kafka Broker へのデータ転送に比べて，Kafka Broker からデータを受け取り解析を行うまでに時間がかかっている．また，スロットごとの平均スループットを機械学習による解析の有無で比較すると，図 6 と図 8 から，Kafka Broker からのデータの受け取りではなく，機械学習の推論に時間がかかっていることが確認できた．図 9 では，解析をしない場合のスループットは，Flink の並列度に比例していないことがわかる．解析をしない場合は，Consumer における処理に時間がかからないため，Kafka での処理の方がボトルネックになっていると考えられる．

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 [15] 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)[16] 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 [17] 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[18] 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 [19] 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 [20] 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 [21] 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 [22] 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[23] 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

本稿では，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，解析処理性能を調査した．実験から，解析スループットは Flink の並列度に比例して増加すること，機械学習による解析がボトルネックであることが確認できた．

今後は複数の機械学習手法を用いて解析性能の評価を行い，より効率良く高精度で処理可能な基盤の構築とリアルタイム動画像解析の実現を目指す．

© 2020 Information Processing Society of Japan

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，2020 年度国立情報学研究所公募型共同研究（20S0501）の助成を受けたものです．

参考文献

[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] Apache Kafka : A Distributed Streaming Platform. . `https://kafka.apache.org/`.

[10] Apache Flink : Stateful Computations over Data Streams. `https://flink.apache.org/`.

[11] A. Ichinose and A. Takefusa and H. Nakada and M. Oguchi. A Study of a Video Analysis Framework Using Kafka and Spark Streaming. In *Proc. Second Workshop on Real-time & Stream Analytics in IEEE Big Data*, pp. 2396–2401, 12 2017.

[12] 孫静涛, 藤原一毅, 竹房あつ子, 長久勝, 吉田浩, 合田憲人. Sinet 広域データ収集基盤のための基盤ソフトウェアの検討. 情報処理学会研究報告 2019-OS-147, pp. 1–9, 7 2019.

[13] 学認クラウドオンデマンド構築サービス. `https://cloud.gakunin.jp/ocs/`.

[14] A. Takefusa, S. Yokoyama, Y. Masatani, T. Tanjo, K. Saga, M. Nagaku, and K. Aida. Virtual Cloud Service System for Building Effective Inter-Cloud Applications. In *Proc. IEEE CloudCom2017*, pp. 296–303, 12 2017.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

© 2020 Information Processing Society of Japan