Vol.2020-HPC-177 No.1
2020/12/21

映像からの行動検出手法SlowFastの分散深層学習の評価

西川 由理[1,2,a)]　佐藤 仁[1]　小澤 順[1,2]

概要：近年，機械学習分野で，映像から人物の動きや仕草を検出する行動検出に関する研究が進みつつある．しかし行動検出は，映像データ量の大きさに起因する数日オーダーの学習時間や，ハイパパラメータ設定の困難さが課題であり，開発効率化のために高速化が期待されている．そこで本研究では，行動検出手法の一つである SlowFast を用いた行動検出データセット AVA の学習に対して分散深層学習を適用し，産業技術総合研究所の大規模 AI クラウド計算システムである ABCI を最大 512 GPU 用いたときのスケーラビリティとその性能要因，検出性能を評価した．性能要因に対する対応策の適用により，64 ノードでのスケーラビリティを約 19.7 倍から約 34.5 倍に改善でき，かつ検出性能も概ね維持できることを確認した．それ以上のノード数では NCCL 経路異常による Ring-AllReduce 処理時間の増加によりスケーラビリティが低下するとともに，検出性能も低下することが分かった．

キーワード：行動検出，SlowFast, 分散深層学習，ABCI

1. はじめに

映像から人物の動きや仕草を検出する行動検出技術 (spatio-temporal action localization) は，防犯や車載，高精度マーケティングをはじめ多くの応用先があることから，近年は深層学習に基づく手法を中心に研究が進んでいる．映像からの行動検出において，中でも 2 次元の空間情報と 1 次元の時間情報を畳み込む 3-dimensional Convolution Neural Network（以下，3D-CNN）を用いる手法は検出性能が高いことで知られ，その一種である SlowFast[1] は大規模な行動検出データセット AVA[2] の検出性能（mAP）を競うランキングにおいて，2020 年 11 月現在 1 位の手法として注目されている．しかし，SlowFast を始めとする 3D-CNN を用いた手法は一般に従来の 2 次元 CNN に比べて計算量が多いほか，I/O 負荷も高いことが知られる．そのため学習時間は，データセットの規模にもよるが，1 日から数日を要することが報告されている．したがって，各応用先に向けた性能チューニングの効率化のため，学習時の高速化が期待されている．

また，深層学習における高速化のために，大規模な計算資源を用いた分散深層学習に関する研究が行われている．特に ImageNet データセットと ResNet-50 モデルを用いた画像認識タスクでは，大規模並列分散環境下での分散深層学習の取り組みが進み，大幅な性能向上に関する事例が報告されている．分散深層学習は一般に，並列数の増加に伴いバッチサイズが増加することによる汎化性能の低下が課題と言われてきたが，Goyal らの linear scaling rule[3] がその解決手法としてデファクトスタンダードになりつつある．SlowFast においても，8 台の GPU を搭載した計算機を 16 ノード用いた計 128 GPU による分散深層学習に linear scaling rule を適用することで，検出性能が 1 ノードでの学習時に匹敵することが報告されている．

ただ，技術の実用化検討においては，SlowFast を用いた AVA データセットの検出性能の再現性に加え，linear scaling rule による高速化の限界や，公開コードの可搬性（他の計算機環境でも台数効果を確認できるか）等を評価する必要がある．それに対し現状は，SlowFast の分散深層学習に用いられた計算機環境や台数効果の詳細が明らかでないことに加え，128 台を超える GPU を用いたときの検出性能やスケーラビリティ，性能低下要因に関する知見も得られていない．すなわち

- 公開コードを用いて並列度を上げたときのスケーラビリティ
- スケーラビリティの低下要因とそれに対する対策の効果
- Linear scaling rule の適用限界

を明らかにする必要がある．

そこで我々は，AVA データセットを用いた行動検出手法 SlowFast の分散深層学習において，産業技術総合研究所の

[1] 国立研究開発法人 産業技術総合研究所
[2] パナソニック株式会社
[a)] nishikawa.yuri@aist.go.jp

ⓒ 2020 Information Processing Society of Japan

大規模 AI クラウド計算システム ABCI[4] を最大 128 ノード (GPU 512 台) 用い，GPU の台数を変化させたときのスケーラビリティと検出性能を評価した．その結果
(1) ミニバッチを生成する並列実行数 (worker 数)
(2) ミニバッチに使用する学習データのインデキシング処理
(3) NCCL[5] の経路情報
の 3 点がスケーラビリティの低下要因になることが分かった．さらに，上記 (1) の worker 数を適切な値に設定するとともに，上記 (2) の処理を行うプロセスを epoch 間で再利用する工夫により，64 ノード 256 GPU を用いた分散深層学習でスケーラビリティを約 19.7 倍から約 34.5 倍に改善でき，かつ検出性能も概ね維持できることを確認した．それ以上のノード数では，(3) NCCL の経路異常による輻輳発生と linear scaling rule の適用限界により，スケーラビリティ，検出性能ともに低下することが分かった．

2. 関連研究

2.1 分散深層学習の関連研究

一般に深層学習では，学習データを「ミニバッチ」と呼ばれる単位に分割し，この単位で学習モデルの更新を行う．分散深層学習ではプロセスごとにミニバッチ単位の学習データを割り当てるため，一度の並列計算における総バッチサイズ (以後，単にバッチサイズと表記する) は並列プロセス数 × ミニバッチで求まる．並列数を増やせば，バッチサイズも増え，学習に必要な計算時間が短縮される一方で，誤差関数の局所解に陥りやすく汎化性能が低下することが報告されており [6]，分散深層学習の課題とされてきた．

2017 年に Goyal らが提示した linear scaling rule[3] は，バッチサイズと学習率を比例させることで汎化性能を保てるという経験則であり，同手法を基本とした改良により ImageNet による画像認識モデルの学習時間の記録更新が続いている [7][8][9][10]．しかし ImageNet/ResNet-50 以外の分散深層学習の事例に関する研究報告は多くなく，映像データを扱う事例も少ない．

2.2 映像を用いた行動検出の関連研究

「行動検出」とは，トリミングされていない映像に含まれる人物（1 名ないし複数名）の動作区間とその位置（座標）を特定し，その動作の分類を行う Spatio-Temporal Action Localization タスクを指す．**表 1** に示すように，関連タスクには，動作区間のみを検出する Temporal Action Proposals，予め一つの動作区間がトリミングされた動画の動作分類を行う Trimmed Action Recognition，動作区間検出と分類を共に行う Temporal Action Localization とがあり，区別されている．なお Trimmed Action Recognition と Temporal Action Localization タスクは，まとめて「行動認識」と呼ばれることが多い．

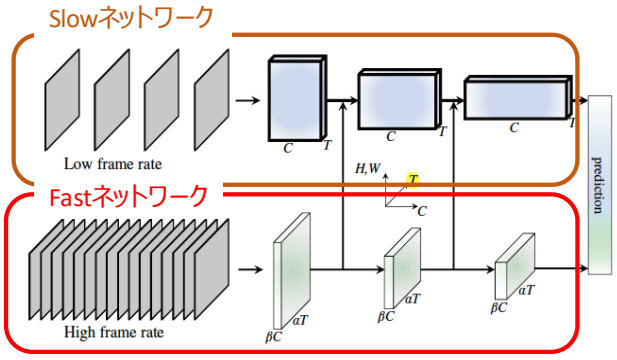

図 1 SlowFast のアーキテクチャ [1]

動画における行動認識や行動検出技術では，画像の空間情報と時間情報の扱い方を検討することになるが，一般に空間情報は，従来の画像認識・検出でよく用いられる 2 次元 CNN（2D-CNN）に基づく手法の性能が高いため，時間情報の特徴量抽出手法が鍵となる．深層学習ベースの行動認識手法として初期に提案されたのは，2D-CNN によって抽出された空間情報と Optical flow を入力とする CNN の出力を統合する Two-stream CNN[11] であり，現在もこの改良アルゴリズムが盛んに開発されている．

一方，2 次元の空間情報と 1 次元の時間情報を 3 次元情報として畳み込む 3D-CNN の代表例として C3D[12], I3D[13] が挙げられる．しかし C3D はパラメータ数の多さから畳込み層の数を増やせない課題や学習データセットの不足のため Two-Stream CNN の性能を下回っていた．I3D は Kinetics-400 という大規模データセットと共に提案された手法で，空間情報を扱う 2D-CNN を ImageNet の学習済モデルで初期化することと Kinetics の使用により高い認識性能を示し，3D-CNN ベースの手法の有効性を示した．しかし，I3D も認識したい行動の種別に関わらず動画を固定長のフレームに分割して学習を行う計算効率の悪さが指摘されていた．本稿がターゲットとする SlowFast は，この課題に鑑みて提案されたものであり，従来の 3D-CNN ベースの手法に比べて計算コストが改善されている．それでもなお，1 日から数日程度の学習時間を要することから，本研究では分散深層学習による高速化について調査する．

3. 行動検出手法 SlowFast と AVA データセット

3.1 SlowFast の概要

SlowFast は，Facebook Research が 2018 年末に発表した，高精度に行動検出を行うモデルの一つである．**図 1** のアーキテクチャに示されるように，空間的解像度が高いが動きに弱い Slow ネットワークと，その逆で空間的解像度が低いが動きに強い Fast ネットワークの 2 つを用いることで，空間的な文脈情報に強く，かつ動きにも強い動作検出を行う．具体的には，Slow ネットワークのサンプリング間

表 1　行動検出の関連タスク

分類	動作区間検出	動作分類	人物領域抽出	SlowFast が対象とするタスク
Temporal Action Proposals	○	-	-	
Trimmed Action Recognition	-	○	-	
Temporal Action Localization	○	○	-	✓
Spatio-Temporal Action Localization	○	○	○	✓

隔が τ であるとき，Fast ネットワークのサンプリングレートを Slow ネットワークの α 倍，チャネル数を $1/\beta$ 倍に設定する．典型的には，$\tau = 8$（i.e. 動画のフレームレートが 32fps なら，Slow ネットワークのフレームレートは 4fps になる），$\alpha = 4, \beta = 8$ などに設定される．また，バックボーンネットワークには，ImageNet の物体認識等にも採用される，CNN ベースの ResNet-50 または ResNet-101 が用いられる．SlowFast は 2020 年 11 月現在，行動認識タスクのデータセットである Kinetics[13] において 2 位，後述する行動検出タスク用データセットである AVA[2] の version 2.1 において 1 位である．

3.2　AVA の概要とデータ量

AVA は，437 本の 15 分間の動画に対し，1 秒間に 1 回の間隔で正解ラベルが付与される．また，データセット全体の約 6 割を学習データとして用いるため，学習データ数における正解ラベル数は 437 本 ×60%×15 分 × 60 秒 = 約 23.6 万件となる[*1]．さらに SlowFast では，上述の Fast ネットワークのフレームレートに合わせて学習データを読み込む必要がある．32fps の動画において，パラメータを標準的な $\tau = 8$, $\alpha = 4$ に設定したとき，Fast ネットワークのフレームレートは 16fps となるため，1 ラベルに対して 16 枚の画像を読み込む必要がある．すなわち，ImageNet 等と同様に 224 x 224 ピクセルにトリミングした RGB 画像を用いる場合，学習データの総データ量は 23.6 万件 $\times 224^2$ ピクセル ×3 (RGB channel)×16 枚 = 約 56.5GB となる．AVA データセットや SlowFast による検出性能に関する詳細は付録 A.1 で紹介する．

3.3　分散深層学習の手順

SlowFast を用いた AVA データセットの学習手順，および分散深層学習の手順について述べる．なお，本稿ではバックボーンに ResNet-50 を適用するものとする．

SlowFast のパラメータは $\tau = 8$, $\alpha = 4$, $\beta = 8$ とする．ネットワークは Kinetics-600 データセットにより ResNet-50 を学習したモデルを用いて初期値する．学習は 20 epochs 行うが，そのうち最初の 5 epochs は線形に学習率を増加させる linear warmup[3] を適用し，以後 10, 15 epoch 目に学習率を 1/10 とする stepwise decay を適用する．その他のハイパパラメータは [1] と同様に設定する．また，バッチサイズは，計算機 1 ノードあたり 32 に設定する．ここで SlowFast における「バッチサイズ」とは画像の枚数ではなく，正解ラベルの数を指す．前述のとおり，Fast 側のネットワークには 1 ラベルあたり 16 枚の画像が読み込まれる．したがって，バッチサイズが 32 のとき，1 ノードでの 1 iteration あたり 512 枚の画像（224×224 ピクセルの RGB 画像の場合，約 77MB）の読み込みが行われる．

次に，分散深層学習の実現方法を述べる．学習率は前述のとおり Goyal らの linear scaling rule を採用する．具体的には，計算機 1 ノードあたりの学習率を 0.05 に設定し，n ノードを用いるときの学習率を $0.05n$ とする．また，著者らのコードは PyTorch の DistributedDataParallel (DDP) により実現されており，本稿でもその実装を踏襲する．ノード間通信は OpenMPI を用い，TCP initialization を用いて通信の初期化を行う．そして各ノード内で，全 GPU プロセスを起動する SlowFast の main 関数 (`run_net.py`) を実行する．

表 2　ABCI の計算ノードのスペック

CPU	Intel Xeon Gold 6148 (27.5M Cache, 2.40 GHz, 20 core) × 2
GPU	NVIDIA Tesla V100 SXM2 × 4
Mem	384 GiB
SSD	1.6TB NVMe SSD × 1

4.　評価

4.1　評価環境

分散深層学習の評価は，産業技術総合研究所の大規模 AI クラウドシステム ABCI 上で行った．**表 2** に計算ノード 1 台のスペックを示す．ABCI は 1 ノードあたり 4 台の GPU を搭載する．また計算ノード間は EDR Infiniband により，Full-bisection Fat Tree 構成で接続される．ただしラックを跨ぐ計算ノード間は Full-bisection の帯域が 1/3 となるように接続されている．

計算ノードの OS は CentOS 7.5, Linux のカーネルは v3.10.0 である．ソフトウェアについて，GPU に対しては，CUDA Toolkit v10.0.130.1，CuDNN v7.6.5 を使用し，GPU 間の集団通信を行うため NCCL v2.5.6-1 および OpenMPI v2.1.6 を使用した．SlowFast の公開コード [14] は，PyTorch フレームワークを用いて python で記述されている．このため Python およびライブラリは，Python

[*1] 原稿執筆時点でアクセスできない動画データが存在し，本研究で用いた学習ラベル数は約 19.5 万件である．

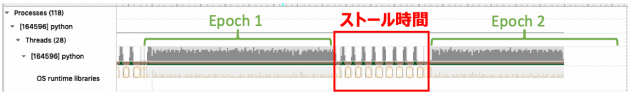

図 2 Epoch 間のストール現象

表 3 計算ノード数 4 における Worker 数とストール時間長の関係（単位：秒）

worker 数	ストール時間	10 iteration の学習処理時間
0	0	90
4	50	19
6	75	15
8	110	11

v.3.6.5, PyTorch v1.5.0 をそれぞれ使用した．また SlowFast では，人物検出のため，Detectron [15] を内部的に利用しており，2020 年 5 月 21 日時点での最新コードを取得して適用した．学習は，ABCI の計算ノード 1 台，16 台，32 台，64 台，128 台 (GPU 4 台，64 台，128 台，256 台，512 台）でそれぞれ行った．

また後述のように，PyTorch ではミニバッチを生成するデータローダの並列実行数である”worker 数”を設定することができる．基本的には多くの worker を用いるほど CPU 負荷が上がるが，ミニバッチあたりの学習時間 (以後，1 iteration と述べる）が短くなることが知られている．本評価では，予め worker 数 1, 2, 4, 6, 8 のときの速度を比較し，8 を既定値とした．

4.2 性能プロファイリングとボトルネック改善

まず予備評価として，NVIDIA Nsight Systems を用い，各計算ノード数での分散深層学習における，2 epoch 分の学習実行時間の内訳を求めた．32 ノード, worker 数 8 でのプロファイリング画面の例を**図 2** に示す．分析の結果，図に示すとおり，学習の epoch 間に一切の CUDA イベントがない 2 分弱のストール現象が発生しており，その時間長が並列数によらず一定であることが分かった．

本現象の原因調査のため，学習時の各種パラメータを変更して評価したところ，ストール時間長は worker 数と関連しており，**図 3** に示すように，実際に GPU を用いた学習処理の時間が worker 数の増加に応じて減少する一方，ストール時間が増加することが明らかになった．この影響は，iteration 数が少なくなる多ノードでの学習において顕著となり，例えば 64 ノード, worker 数 8 での実行時には，実行時間の約半分をストール時間が占める．

本現象は PyTorch 特有の課題として GitHub の issue をはじめとするユーザコミュニティでも度々議論されており，各 worker がミニバッチに含むデータサンプルをインデキシングする sampler プロセス生成処理のオーバヘッドが原因であることが指摘されている [16]．そこで，オーバヘッドを軽減するため，初回 epoch で生成した sampler プロセスを，次回以降の epoch でも再利用する改善策を適用した．以後，本改善策を RepeatSampler と呼ぶ．この結果，いずれの計算ノード数および worker 数でも，表 3 に示すストール時間を 0 秒台と，無視できる程度に短縮できることを確認した．さらに，worker 数が 6 以下かつ計算ノード数が 32 以下における iteration 処理時間は表 3 の値を保つが，それを超える場合は iteration 処理の時間が増加することが分かった．したがって，次節以降では worker 数を 6 に設定し，改善策である RepeatSampler 適用前と適用後のスケーラビリティと検出性能を評価する．

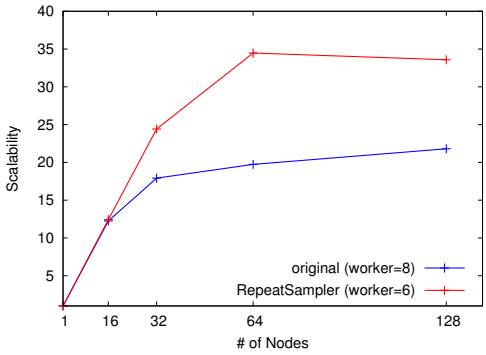

図 3 SlowFast のスケーラビリティ

表 4 各ノード数における計算処理と通信処理の比率

ノード数	計算処理	通信処理
32	56.2%	43.8%
64	40.3%	59.7%
128	21.9%	78.1%

4.3 スケーラビリティ評価と性能要因の考察

SlowFast の分散深層学習を 5 epoch 実施したときの実行時間から求めた台数効果を評価した．結果を**図 3** に示す．横軸はノード数，縦軸は 1 ノード（GPU 4 台）の実行時間を基準としたときのスケーラビリティであり，青線は改善前のコード（worker 数 8），赤線は RepeatSampler 適用後（worker 数 6）を表す．

改善前は，1 ノードでの実行時間が約 9 時間 4 分であったのに対し，32 ノードで約 30 分，64 ノードでは約 27 分と，ストール時間により 32 ノード以降の性能が飽和した．一方，RepeatSampler を適用後は，32 ノード以降のスケーラビリティが改善し，32 ノードで約 22 分，64 ノードで約 15 分半となり，1 ノードと比べ約 34.5 倍高速となった．

一方，128 ノードのスケーラビリティは，64 ノードよりも悪化した．この要因を調べるため，32 ノード，64 ノード，128 ノードにおける実行時プロファイルを取得し，その内訳を**表 4** に示す．ノード数の増加に伴い通信処理の割合が増加しているが，さらに時間換算すると，128 ノードの通信時間が 64 ノードの約 1.8 倍に増加したことが分かった．

ⓒ 2020 Information Processing Society of Japan

表 5 異なるノード数による検出性能 (mAP)
ベースライン: 24.2%

ノード数	RepeatSampler なし	RepeatSampler あり
16	24.6%	24.6%
32	24.4%	24.4%
64	24.0%	24.0%
128	21.1%	20.9%

また，Nsight Systems によるプロファイリングの結果，通信時間の 95%以上が，ncclAllReduce の実行時間であることも明らかになった．一方，学習時の実行ログに記載された NCCL[5] の経路情報を確認すると，ABCI の GPU 間接続網におけるデータの send 時と receive 時の経路が非対称であり，Infiniband に対し冗長な経路が形成されていることを確認した．すなわち，今回用いた NCCL モジュールによる経路情報が ABCI アーキテクチャに最適化されておらず，GPU 間接続網の nvLink に比べて帯域の小さい Infiniband で輻輳が発生し，iteration 毎に各計算リソースの勾配情報を集約する Ring-AllReduce 処理に時間を要していることが考えられる．一方，ABCI アーキテクチャ向けにソースビルドした NCCL ライブラリの適用により，ABCI を用いた ImageNet-1k の学習において，512 GPU 程度まで Ring-AllReduce の処理時間が概ね一定であることが報告されている [17]．本研究で用いた AVA データセットの総学習データ量は約 56.5GB であり，ImageNet に比べて小規模である．したがって NCCL ライブラリの改良により通信性能を改善できる可能性があり，今後の課題とする．

4.4 検出性能

最後に，SlowFast の分散深層学習を行った時の，AVA データセットの検出性能 (mAP) を評価した．結果を**表 5** に示す．これにより，RepeatSampler の適用による検出性能への影響はほぼ見られないことが確認された．一方，著者らが報告した検出性能 24.2%に対し，64 ノードでは 0.2%，128 ノードでは 3.1%低い結果となった．このことから，SlowFast のソースコードの大きな改変を伴わない AVA データセットの大規模分散深層学習においては，スケーラビリティと検出性能の双方の観点から，64 ノード（256 GPU）が限界であることが示唆された．

したがって，今後の課題としては，前節で述べた NCCL ライブラリの改善に加え，Forward/Backward 処理と通信のオーバラップや FP16 の適用，SlowFast のバッチサイズと画像サイズ，τ の値を学習中に動的に変更するマルチグリッド学習 [18] などにより，実行時間の改善を図る予定である．

5. まとめ

本稿では，行動検出用の映像データセット AVA を用いた SlowFast を用いた分散深層学習における，GPU の台数を変化させたときのスケーラビリティと検出性能を評価した．その結果，(1) ミニバッチを生成する並列実行数 (worker 数)，(2) ミニバッチに使用する学習データのインデキシング処理，(3) NCCL の経路情報の 3 点が性能要因であることが分かった．このうち (1) の worker 数を適切な値に設定するとともに，(2) の処理プロセスの再利用を行う RepeatSampler の適用により，64 ノード (256 GPU) におけるスケーラビリティを約 19.7 倍から約 34.5 倍に改善することができ，かつ検出性能も概ね維持できることを確認した．一方 128 ノード (512GPU) 以上では，Ring-AllReduce 処理時間の増加によりスケーラビリティが低下するとともに，linear scaling rule の適用限界により検出性能も 3%以上低下することが示された．したがって，ソースコードの大きな改変を伴わない AVA データセットのを用いた分散深層学習においては，スケーラビリティ，検出性能の観点から 64 ノードが限界であることが示唆された．今後は，NCCL 経路情報の改善や計算と通信のオーバラップ，FP16 の適用やマルチグリッド学習 [18] 等により，実行時間の改善を図る予定である．

参考文献

[1] Feichtenhofer, C., Fan, H., Malik, J. and He, K.: SlowFast Networks for Video Recognition, *ICCV* (2019).
[2] Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C. and Malik, J.: AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions, *CVPR* (2018).
[3] Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y. and He, K.: Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour, *CoRR*, Vol. abs/1706.02677 (2017).
[4] 小川宏高，松岡 聡，佐藤 仁，高野了成，滝澤真一郎，谷村勇輔，三浦信一，関口智嗣：AI 橋渡しクラウド–AI Bridging Cloud Infrastructure (ABCI)–の構想，情報処理学会研究報告，Vol. 2017-HPC-160, No. 28, pp. 1–7 (2017).
[5] NVIDIA: NVIDIA Collective Communication Library (NCCL), `https://developer.nvidia.com/nccl`.
[6] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. and Tang, P. T.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=H1oyRlYgg⟩ (2017).
[7] Akiba, T., Suzuki, S. and Fukuda, K.: Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes, *CoRR*, Vol. abs/1711.04325 (2017).
[8] Mikami, H., Suganuma, H., U.-Chupala, P., Tanaka, Y. and Kageyama, Y.: ImageNet/ResNet-50 Training in 224 Seconds, *CoRR*, Vol. abs/1811.05233 (2018).
[9] Ying, C., Kumar, S., Chen, D., Wang, T. and Cheng, Y.: Image Classification at Supercomputer Scale, *CoRR*, Vol. abs/1811.06992 (2018).
[10] Yamazaki, M., Kasagi, A., Tabuchi, A., Honda, T., Miwa, M., Fukumoto, N., Tabaru, T., Ike, A. and

ⓒ 2020 Information Processing Society of Japan

Nakashima, K.: Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds, *CoRR*, Vol. abs/1903.12650 (2019).
[11] Simonyan, K. and Zisserman, A.: Two-Stream Convolutional Networks for Action Recognition in Videos, *Advances in Neural Information Processing Systems (NIPS)*, Vol. 27, pp. 568–576 (2014).
[12] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M.: Learning Spatiotemporal Features With 3D Convolutional Networks, *ICCV* (2015).
[13] Carreira, J. and Zisserman, A.: Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, *CVPR* (2017).
[14] Fan, H., Li, Y., Xiong, B., Lo, W.-Y. and Feichtenhofer, C.: PySlowFast, `https://github.com/facebookresearch/slowfast` (2020).
[15] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y. and Girshick, R.: Detectron2, `https://github.com/facebookresearch/detectron2` (2019).
[16] : DataLoader with option to re-use worker processes, `https://github.com/pytorch/pytorch/issues/15849`.
[17] Ueno, Y. and Yokota, R.: Exhaustive Study of Hierarchical AllReduce Patterns for Large Messages Between GPUs, *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 430–439 (online), DOI: 10.1109/CCGRID.2019.00057 (2019).
[18] Wu, C.-Y., Girshick, R., He, K., Feichtenhofer, C. and Krahenbuhl, P.: A Multigrid Method for Efficiently Training Video Models, *CVPR* (2020).

付　　録

A.1 行動検出データセット AVA

A.1.1 AVA の概要

AVA は前述のとおり，人物の行動検出向け公開データセットである．437 本の 15 分間の動画に対し，1 秒間に 1 フレームの spatio-temporal ラベル (人物のバウンディングボックスおよび複数の行動ラベル）が付与されている．行動ラベルの一覧とそのデータ数を**図 A·1** に示す．横軸は行動ラベルの種類を表し，縦軸はデータ数を表す．図に示すように，AVA の行動ラベルには 80 種類が存在し，それぞれ，Person-Person（対人行為）, Person-Object（対物行為）, Pose（姿勢）の 3 種類に大別されている．したがって，1 名の人物に対して，複数のラベルが付与される場合もある（例：荷物を持ちながら (hold/carry) 歩く (walk)）．また，各行動ラベルに対する学習データ数は様々であり，10 万件以上のデータがある行動（立つ，(相手) を見る，座る等）が 6 種類あるのに対し，学習データ数が 100 件未満の行動ラベルも 15 件ほど存在する．

AVA には 211k の学習 (train) データおよび 57k の検証 (validation) データが用意されており，上述の 80 クラスで学習する．評価は，そのうち 60 クラスに対する検証データを用い，人物検出枠と正解枠との Intersection over Union (IoU) が 0.5 以上の人物に対する行動ラベルの mAP (mean Average Precision) を評価指標として行われる．なお，現在はアノテーションデータの精度が改善された version 2.2 が公開されており，本稿の評価でもそれを用いる．

A.1.2 SlowFast による AVA の検出性能

SlowFast による AVA（バックボーンネットワーク: ResNet-50) の検出性能を**図 A·2** に示す．なお，図中の横軸は行動ラベルの種類，縦軸が個々の行動の検出性能 (AP: Average Precision) を表す．また青棒 (”Slow-only”) は，”Fast” ネットワークを用いなかった時のベースライン性能 (mAP=19.0%) であり，緑線が Fast ネットワークと併用したときの性能改善を表す．

図 A·2 によると，時間分解能の高い Fast ネットワークの併用により，動作が早い行動（例：走る，ダンスをする，手を叩く）などの検出性能が改善していることがわかる．一方，行動ラベルにより mAP のばらつきは大きく，AP=80%に近い行動（歩く，座る）がある一方，AP が 10%に満たない行動が半数以上 (31 種類) を占め，AP の平均である mean AP (mAP) は 24.2%に留まる．

また，図 A·2 の横軸に示す行動ラベルの順序は図 A·1 と同じく，データ数に対して昇順にソートされている．このことから，データ数の量と検出性能は必ずしも一致しない．

ⓒ 2020 Information Processing Society of Japan

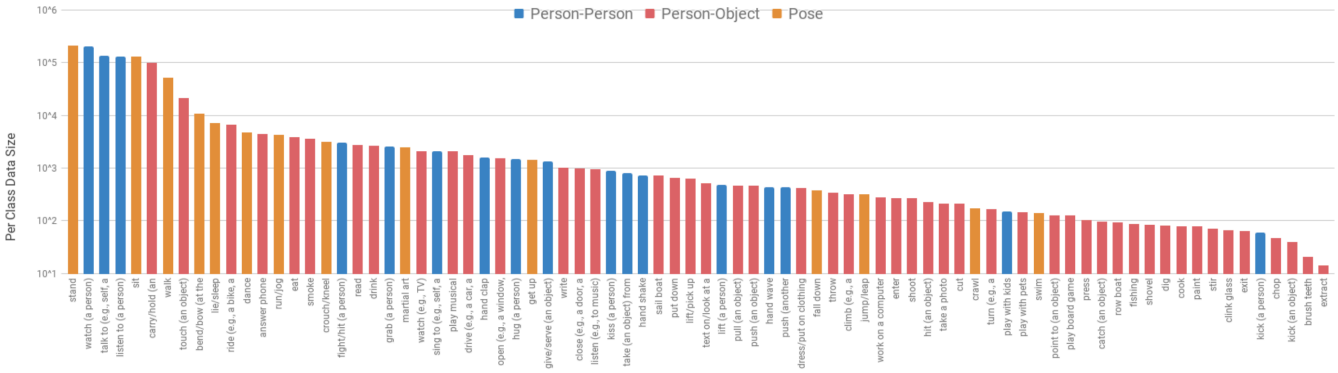

図 A·1 AVA の行動ラベル別のデータ数 [2]

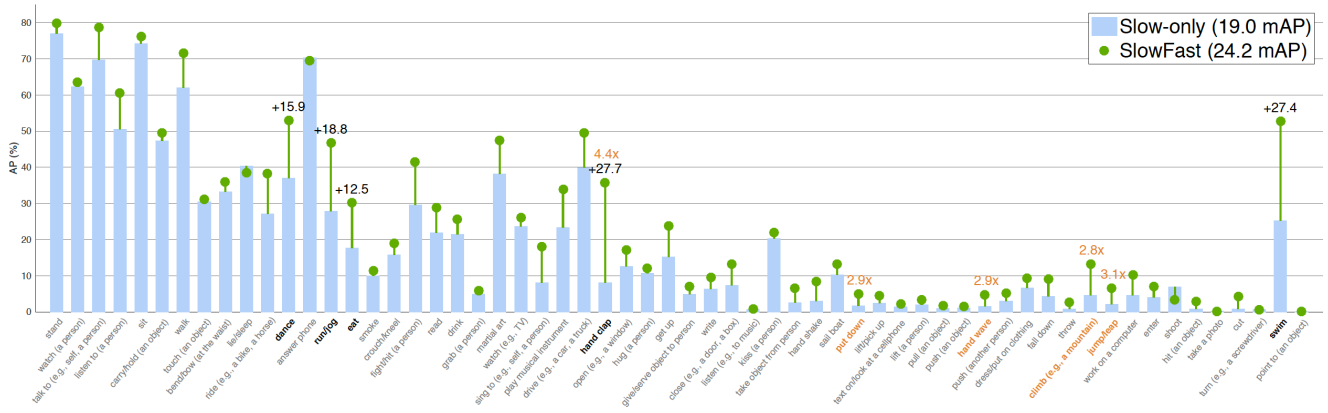

図 A·2 SlowFast の AVA に対する検出性能 [1]