動作認識のための合成データ活用に向けたドメイン適応手法の比較

礒井 葉那[1]　竹房 あつ子[2]　中田 秀基[3]　小口 正人[1]

概要：ディープニューラルネットワークの進歩に伴う学習データ不足の問題について様々な議論が行われており, その解決策の 1 つに合成データを利用した学習がある. 合成データには生成が比較的容易であるという利点があるが, 合成データを用いて学習したモデルには, 実データ解析時にドメインシフトによって解析精度が低下するという課題がある. 本研究では, 合成動画像データを活用した高精度な実動画像データ識別の実現を目的とし, 写実的な合成動画像データを用いて 3D ResNet と TSN をベースとするモデルでそれぞれ学習し, その動作識別精度を比較した. 実験の結果, 合成データと実データの特徴の違いはモーションよりも色や形状, 質感にあること, オプティカルフローを用いる TSN ベースのモデルの方が高精度に実データの動作識別が可能であることがわかった.

A Study on Domain Adaptation Methods for Utilizing Synthetic Data in Action Recognition

HANA ISOI[1]　ATSUKO TAKEFUSA[2]　HIDEMOTO NAKADA[3]　MASATO OGUCHI[1]

1. はじめに

ディープニューラルネットワーク (DNN) の発展により, コンピュータービジョン分野において様々な技術が進歩している. 最近の研究では動画像から人間の行動を解析することができるようになってきており [1][2][3], 家庭内の子供や高齢者の見守りなどへの応用が期待されている.

DNN による画像解析の学習精度は，ラベル付き学習データセットのサイズとバリエーションに大きく依存していることが知られている [4] が, 十分なデータの収集とラベル付けには大変な時間と費用がかかる. そうした学習データ不足の問題に対する解決策として, 合成データを活用することが注目されている [5][6][7]. 合成データとはコンピュータを用いて生成したデータのことであり, 実データと比較して大量かつ多様なデータを容易に生成することができるという利点がある. 特に, 静止画像よりも作成が困難である動画像データでは, 合成データの活用が期待されている [8]. しかし, 学習時に用いたデータと利用時に用いるデータの性質が異なる場合 (ドメインシフト) には精度が低下してしまうため, 多くの場合にドメイン適応による対応が必要となる [9][10][11][12]. 特に, 動画像データのドメイン適応については十分に研究されておらず, 合成動画像データによるドメイン適応では, 高精度な実データ解析は実現されていない [13].

そこで我々は, 合成データを用いた高精度な実動画像データ解析を実現することを目的として, 写実的な合成データを作成して 3D ResNet でドメイン適応による学習を行い, その動作識別精度を調査した [14]. 実験に用いるデータセットとして, 実動画像 Ochahouse-Real と合成動画像 Ochahouse-Syn により構成される Ochahouse Dataset を作成した. しかし, その時点では作成した 3D ResNet とドメイン適応手法である DANN を組み合わせたモデルはドメインシフトに十分に対応できず, ドメイン適応を行わずにラベル付き Ochahouse-Syn のみで学習したモデルで Ochahouse-Real の解析を行う実験では十分な解析精度が

[1] お茶の水女子大学
Ochanomizu University
[2] 国立情報学研究所
National Institute of Informatics
[3] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology

得られなかった.

本研究では, 合成データと実データ間において人間のモーションの違いは比較的少ないと考え, オプティカルフローを用いて学習する TSN (Temporal Segment Networks)[15] による実験を行い, dd 3D ResNet による学習と比較した. 実験の結果, TSN とドメイン適応を用いた学習のほうが 3D ResNet とドメイン適応を用いた学習よりも高精度で実データの動作識別ができることがわかった. また, Ochahouse Dataset の合成データと実データ間には動作の違いよりも見た目の違いのほうが大きく, 見た目の違いから起こるドメインシフトに対応することで動作識別精度をさらに改善できる可能性があることがわかった.

2. 関連研究

2.1 合成データ

合成データとは, コンピュータにより人工的に生成されるデータであり, 実データと比較して大量かつ多様な生成が容易であるという利点をもつ. 合成データは, 実データに加えて学習データの多様化・増量や, ドメイン適応を含む転移学習に活用される.

合成データのみでの学習は, ロボットのシミュレーション実験などを主な目的として研究されてきた. 文献 [5] では合成画像のみを用いて実画像の学習を行うことを目指し, 前段階として ImageNet で事前学習し, ランダム化されたレンダリングピクセルでファインチューニングしたニューラルネットワークでロボット制御が行えることを示している. Tobin らは 2017 年, シミュレートするテクスチャ, オクルージョンレベル, シーンの照明, カメラの視野, レンダリングエンジン内の均一なノイズに対してドメインランダム化を行うことで, 単純な環境でシミュレートされた画像のみで学習した DNN でドメイン適応を行わずに実画像での高精度な物体検出に初めて成功した [6].

実データに加えて学習データを増強する目的で, 実際の都市での運転シーンにおける物体検出のための合成動画像データセット Virtual KITTI [7] が作成された. 彼らはカメラの視点, 光源, オブジェクトのプロパティをランダム化した写実的な画像をレンダリングによって生成し, 合成データが物体検出, 特にマルチオブジェクトの追跡において実世界の解析に有用であることを示した.

動画像における合成データに関する先行研究には [8][16][13] がある. 文献 [8] では, 多様で写実的な人間行動動画像のデータセット PHAV (Procedural Human Action Videos) を作成し, 実データに加えて学習すると HMDB-51[17], UCF-101[18] における解析に有効であることを示した. 文献 [16] では, 動画像のテキストや背景は物体の動きを表現するオプティカルフローにほとんど影響を与えないことに着目し, 背景を簡略化した人間行動合成動画像データセットを作成した. このデータセットから抽出したオプティカルフローで, RGB 画像とオプティカルフローそれぞれに畳み込みを施し統合する動画像解析ネットワークの 1 つ TSN (Temporal Segment Networks) [15] を追加で学習することにより, UDF-101 および HMDB-51 における精度向上に有効であることを示した.

実データと併用して用いられたこれらのデータセットと異なり, ゲームプレイ動画像から収集した 50 クラスの行動データセット Kinetics-Gameplay は, ドメイン適応による動作分類のために作成された [13].

2.2 ドメイン適応

ドメイン適応とは, ドメインシフトに対応するための手法であり, 合成データで学習された分類器を実データに用いる場合に必要とされることが知られている [9][10][11][12]. ドメイン適応の代表的な手法には, 解析したいデータであるターゲットデータと正解ラベルなどの多くの情報を持つソースデータとを同時にネットワークに入力してデータ間に共通する特徴を学習させる DANN (Domain-Adversarial Neural Networks)[19] の他に, ソースデータで学習させた特徴抽出器をターゲットデータ用特徴抽出の学習に用いる ADDA (Adversarial Discriminative Domain Adaptation)[20], 2 つのクラス分類器を用いて特徴抽出後の分布を近づける MCD (Maximum Classifier Discrepancy)[21] などがある.

動画像におけるドメイン適応では, Chen らが TA^{3}N (Temporal Attentive Adversarial Adaptation Network) というドメイン適応ネットワークを提案した [13]. 文献 [13] では, 学習と同時に時間的ダイナミクスのアラインメントを行い, またドメインの不一致を利用して時間的ダイナミクスを明示的に考慮することで高精度なドメイン適応を実現した. Pan らはクロスドメイン共同アテンション機構を提案し, ドメイン間の時間的なずれの問題に対処する方法を提案した [22]. また Choi らは, より識別性の高いクリップに焦点を当て, ビデオレベルのアラインメントを直接最適化する注意メカニズムを提案した [23]. さらに, 補助タスクとしてクリップ順序予測を使用し, これらにより行動に大きく関与している人物や物体に焦点を当てた表現を学習することに成功した.

文献 [13] では合成データで動画像ドメイン適応を行っている. Kinetics-Gameplay というゲームプレイ動画から作成したデータをソースデータに利用して, ターゲットデータ (Kinetics の 30 のサブクラス) の分類に 17.22% から 27.50% の精度向上を達成した. しかしながら, この精度はラベルを使用してターゲットデータで学習した場合の 64.49%には遠く及ばない.

3. Ochahouse Dataset

我々は, 動画像による動作識別問題における合成動画像の

表 1　Ochahouse Dataset の動作クラスとデータ数

クラス	walking	sitting down	sitting	standing up	lying down	lying	getting up
合成データ Ochahouse-Syn	997	747	1118	780	250	250	250
実データ Ochahouse-Real	96	44	56	51	32	39	32

図 1　実動画像 Ochahouse -Real の 1 フレーム

図 2　合成動画像 Ochahouse -Syn の 1 フレーム

効果を評価するためのデータセット **Ochahouse Dataset** を作成した．これは部屋の中を 1 人の人が自由に動きまわり 7 種類の動作をする様子を 1 台の固定されたカメラで収録した実動画像 **Ochahouse-Real** と，合成動画像 **Ochahouse-Syn** で構成される．Ochahouse-Real は，お茶の水女子大学の実験住宅 *OchaHouse*[24] 内で筆者が各動作を行い収録した．Ochahouse-Syn の作成には Unity®を使用した．Ochahouse-Syn では [8] と同様に，Unity Asset Store から入手した既成人物モデルや行動アニメーションを利用した．いずれもフレームレートは 5fps である．Ochahouse Dataset では，walking, sitting down, sitting, standing up, lying down, lying, getting up の 7 種類の動作クラスを作成した．各動作クラスのデータ数は表 1 の通りであり，各動画像は約 3 秒から 7 秒程度の長さとなっている．作成した動画像データの 1 フレームを図 1, 図 2 に示す．

4.　実験設定

Ochahouse-Real の動作識別をするために Ochahouse-Syn を活用した学習を行う実験の方法について述べる．学習には 3D ResNet をベースとするモデルと，TSN をベースとするモデルを利用し，それらのモデルでの学習時にドメイン適応を行う効果を調査した．4.1 節では 3D ResNet をベースとするモデルでの学習方法について，4.2 節では TSN をベースとするモデルでの学習方法について，4.3 節ではこれらの学習におけるハイパーパラメータの決定方法について述べる．

4.1　3D ResNet ベースモデル

3D ResNet は動画像分類に用いられる主流なネットワークの 1 つで，三次元畳み込みを用いる．三次元畳み込みネットワークでは，時系列順に並んだ複数枚の画像を 1 つの入力とし，1 枚の画像の縦・横の空間方向と複数の画像間の時間軸方向の三次元の次元削減を行うことで動画像から特徴を抽出し，動作分類を行う．

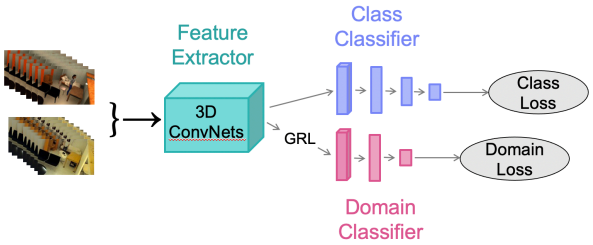


図 3　3D ResNet based DANN

4.1.1　ドメイン適応を行う学習

合成データのみでの学習では Ochahouse-Syn のみで学習を行うが，ドメイン適応を行う学習では Ochshouse-Syn と Ochahouse-Real の両方を用いる．ドメイン適応を用いた手法では，3D ResNet-18 と DANN を組み合わせた図 3 のようなネットワーク 3D ResNet-18 based DANN を作成した．このネットワークでは，まず 3D ResNet に基づいた特徴抽出器で空間方向・時間軸方向の 3 次元の次元削減を同時に行い，動画像から特徴抽出を行う．DANN と同様に，抽出された特徴は，クラス分類器と，勾配反転層 (GRL, Gradient Reverse Layer) を経てドメイン分類器とにそれぞれ提供され，クロスエントロピー誤差関数でクラス分類損失 L_y とドメイン分類損失 L_d が算出される．クラス分類損失 L_y を最小化，ドメイン分類損失 L_d を最大化するようにこれらの加重和 $L_y + \lambda L_d$ を最適化する敵対的学習を行うことで，クラス間の違いは識別できるように，ドメイン間の違いは混合するようになり，ネットワークにドメイン間に共通する動作の特徴を学習させる．

4.2　TSN ベースモデル

TSN (Temporal Segment Networks)[15] は，動画像分類に用いられるもう一つの主なネットワークである．TSN は，動画像は画像フレームから特徴抽出を行う Spatial Stream と，オプティカルフローから特徴抽出を行う Temporal Stream の 2 つの畳み込みネットワークにより構成される．テスト時には，それぞれ学習された Spatial Stream の出力と，Temporal Stream の出力との加重平均から分類を行う．

TSN には，3D ResNet と比べ，オプティカルフロー作成の必要があり End-to-End 学習ができないというデメリットがあるが，モデルのパラメータが少なく学習しやすいというメリットがある．

4.2.1　TSN+DANN

Spatial Stream と Temporal Stream のそれぞれで，3D ResNet の時と同様に DANN によるドメイン適応を行い，

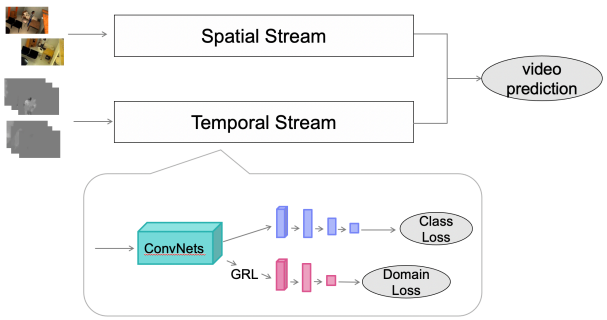


図 4　TSN based DANN

クラス分類損失 L_y を最小化, ドメイン分類損失 L_d を最大化するようにこれらの加重和 $L_y + \lambda L_d$ を最適化する学習を行う. 図 4 のようなモデルを作成した.

4.3　ハイパーパラメータの決定

実装には PyTorch[25] を用い, 損失関数にはクロスエントロピー誤差を, 最適化手法には 勾配降下法 (SGD) を用いて, バッチサイズ 16 または 32 で 60 エポックの学習を行った. バッチサイズはドメイン適応時に同量のソースデータとターゲットデータを用いるため, 2 倍の 32 となる. 計算には産業技術総合研究所の ABCI を用いた.

また, ハイパーパラメータの決定には Optuna[26] によるベイズ最適化を用いた. learning rate と weight decay はそれぞれ 1e-5 から 1e-2 までの小数値 (ログスケール), ドメイン適応のパラメータである θ は 0.5 から 10 までの小数値, γ は 4 から 14 までの整数値, TSN の dropout は 0 から 1 までの小数値から探索した. 各学習手法に対し 30 回の試行を行い決定したパラメータを表 2 に示す.

ドメイン適応を行うときの最適化の目的関数 $L_y + \lambda L_d$ のハイパーパラメータ λ について, 学習の初期は小さい値から始まり, 中盤から増加して 1 になるよう, 本実験では以下のように設定した.

$$\lambda = \theta \left(0.5 \cdot \left(\frac{1}{1 - 2\gamma(p - 0.5)} - 0.5 \right) \cdot \frac{1}{\frac{1}{1-2\gamma \cdot 0.5} - 0.5} + 0.5 \right)$$

ここで p は学習の進行率で, 0 から 1 まで線形に増加する. λ が 0 から 1 まで次第に増加することで, 学習の初期はドメイン分類損失よりクラス分類損失の方をより優先してモデルを更新させる.

5.　実験結果

4 章の設定で学習し実データの動作識別を行った実験の結果を本章で述べる. 5.1 節では実データ動作識別精度について, 5.2 節では学習したモデルから抽出される Ochahouse-Dataset の各動画像の特徴量の可視化について, 5.3 節では, 学習曲線について述べる.

5.1　動作識別精度

3D ResNet ベースモデル, TSN ベースモデルそれぞれの学習手法で学習した時の実データ Ochahouse-Real の動作識別精度を表 3 に示す. また, TSN ベースモデルにおける Spatial Stream, Temporal Stream それぞれの精度を表 4 に, TSN ベースモデルで Spatial Stream と Temporal Stream の出力からそれぞれ 1:1, 1:3, 1:9, 1:25 の加重平均をとった両 Stream 合わせた出力から動作識別をした精度を表 5 に示す.

表 3(i)(iv) では, Ochahouse-Real の正解ラベルを用いて教師あり学習した場合は 3D ResNet では 81.14% , TSN では 78.13% の精度で動作認識ができたが, (i)(iv) 以外では Ochahouse-Real のラベルを使わない学習のため, これらの精度より低くなることに注意が必要である. 表 3(ii)(iii)(v)(vi) から, 3D ResNet ベースモデルでも TSN ベースモデルでもドメイン適応を行うことで実データ Ochahouse-Real の動作識別精度が低下していた. また, 表 3(ii)(v), (iii)(vi) から, ドメイン適応を行う場合でも行わない場合でも, 3D ResNet ベースモデルでの解析精度よりも TSN ベースモデルでの解析精度の方が高いことが読み取れる.

表 4(ii)(iii),(v)(vi) から TSN ベースモデルでは Spatial Stream より Temporal Stream の方が動作識別精度が高いこと, Spatial Stream ではドメイン適応により Ochahouse-Real の動作識別のテスト精度が向上したが, Temporal Stream では低下していた. また, 表 5 よりこれらの出力を合わせて得られる TSN の動作識別精度は, Temporal Stream の出力の重みを大きくするほど高くなるが, Temporal Stream だけを用いて予測する場合の動作識別精度を上回らなかったことがわかる. さらに, 表 3, 4, 5 から, Ochahouse-Syn を用いて Ochahouse-Real の正解ラベルを用いずに学習した場合はいずれの学習手法でも, Ochahouse-Real で教師あり学習した精度には及ばず, ドメインシフトに十分に対応できていないことがわかる.

5.2　特徴量プロット

3D ResNet と TSN の Spatail Stream および Temporal Stream の pooling 層の前までを特徴抽出器とみなし, 学習された特徴抽出器で抽出した Ochahouse-Syn, Ochahouse-Real の特徴量を UMAP [27] でプロットしたものを図 5, 6, 7 に示す. 各色は赤が walking, 青が sitting down, 緑が sitting, シアンが standing up, マゼンタが lying down, 黄が lying, 黒が waking up の動作クラスを表している. 各図の source data は Ochahouse-Syn の特徴量を, target data は Ochahouse-Real の特徴量を表す.

図 5(a) では, 3D ResNet ベースモデルにおいて, ドメイン適応を行わない場合に source data と target data の特徴の分布の形状が一致しておらず, また target data の特

表 2　optuna による探索で決定したハイパーパラメータ

	learning rate	weight decay	θ (ドメイン適応)	γ (ドメイン適応)	dropout(TSN)
3D ResNet	1e-3	1e-5	-	-	-
3D ResNet + DANN	1e-3	3e-5	1.1	7	-
TSN (Spatial Stream)	7e-3	5e-5	-	-	0.46
TSN (Temporal Stream)	2e-4	5e-5	-	-	0.96
TSN+DANN (Spatial Stream)	1e-4	5e-5	6	9	0.28
TSN+DANN (Temporal Stream)	5e-3	5e-5	6	7	0.15

表 3　実データの動作識別精度

	学習手法	精度 (%)
(i)	3D ResNet(実データで学習)	81.14
(ii)	3D ResNet	40.00
(iii)	3D ResNet + DANN	27.57
(iv)	TSN(実データで学習)	78.13
(v)	TSN	71.88
(vi)	TSN + DANN	59.38

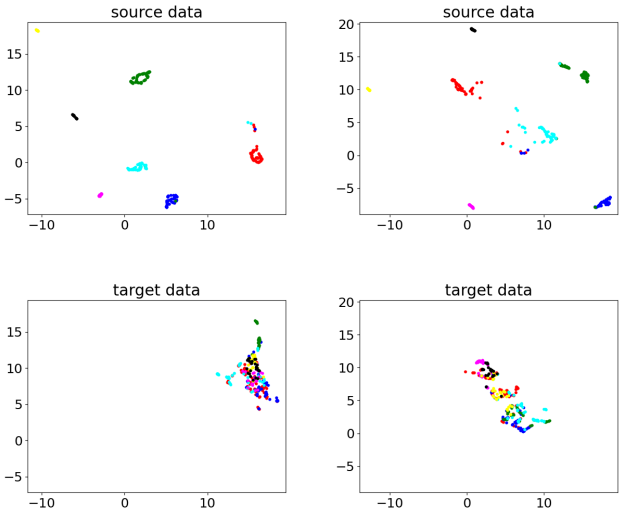
表 4　TSN ベースモデルでの動作識別精度

		精度 (%)
(i)	Spatial Stream (実データで学習)	87.50
(ii)	Spatial Stream	31.25
(iii)	Spatial Stream + DANN	37.50
(iv)	Temporal Stream (実データで学習)	79.69
(v)	Temporal Stream	71.88
(vi)	Temporal Stream + DANN	59.38

表 5　TSN ベースモデルでの動作識別精度

		精度 (%)
(i)	TSN (実データで学習, 1:1)	78.13
(ii)	TSN (実データで学習, 1:3)	78.13
(iii)	TSN (実データで学習, 1:9)	78.13
(iv)	TSN (実データで学習, 1:25)	78.13
(v)	TSN (1:1)	31.25
(vi)	TSN (1:3)	37.50
(vii)	TSN (1:9)	56.25
(viii)	TSN (1:25)	71.88
(x)	TSN+DANN (1:1)	53.13
(xi)	TSN+DANN (1:3)	56.25
(xii)	TSN+DANN (1:9)	59.38
(xiii)	TSN+DANN (1:25)	59.38

徴の点は各色が混ざり合って配置されているため target data の動作分類が十分にできないことがわかる．図 5(b) ではドメイン適応によって target data の特徴の点が色ごとに比較的まとまって配置されるよう改善したが, source data と target data の特徴の分布の形状は赤や水色の一部の点を除き一致していないため, ドメイン適応が十分でないことがわかる．図 6(a)(b)(c)(d) では, TSN ベースモデルの Spatial Stream, Temporal Stream ともに, ドメイン適応を行わない場合は source data と target data の特徴の分布は異なっているが, ドメイン適応によって改善していることが読み取れる．3D ResNet ベースモデルよりも



(a) 3D ResNet　(b) 3D ResNet + DANN

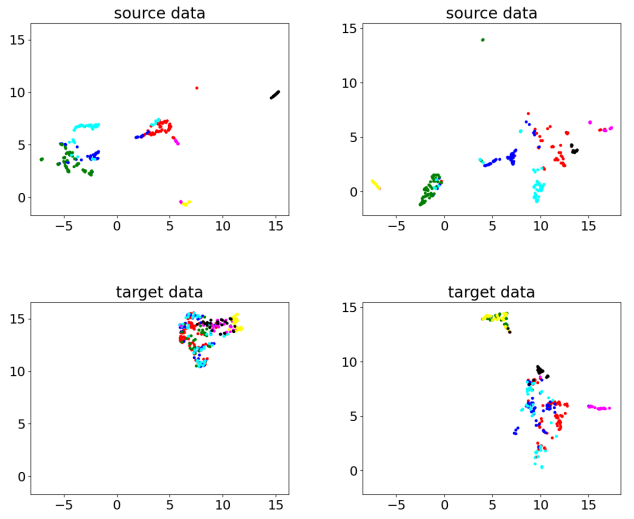
図 5　UMAP による特徴量の可視化 (3D ResNet)

source data で各点の色ごとに分かれていないのは, TSN の方が学習に使うフレーム数が少ないため, 動作識別が困難になっているためと考えられる．

5.3　学習の経過

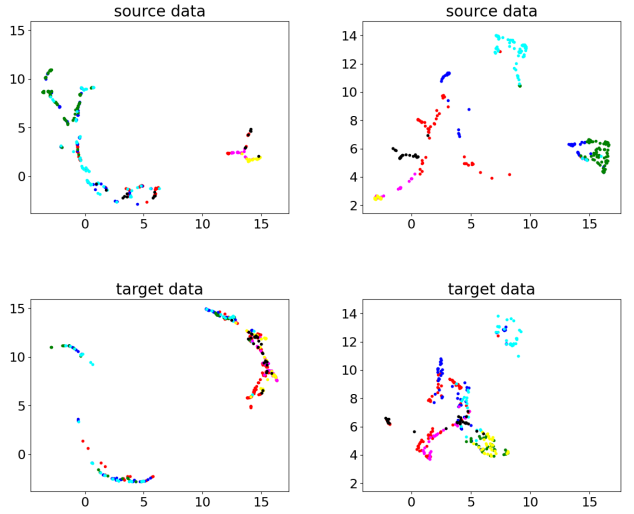
3D ResNet ベースモデル, TSN ベースモデルでの学習時の各エポックでのクラス分類精度を図 8,10 に, クラス分類ロスを図 9,11 に示す．横軸は学習エポック数を, 縦軸は精度またはロスを表している．緑線は source data での学習時, オレンジ色の線は source data でのテスト時, 青線は target data でのテスト時の精度またはロスを示す．

図 8(a)(b) から, 3D ResNet ベースモデルでの学習において, ドメイン適応によって検証精度が向上したことが読み取れる．図 10(a)(b) から, TSN ベースモデルの Spatial Stream ではドメイン適応を行わない場合は Ochahouse-Syn の学習が進んでも Ochahouse Real の検証精度が上がらないが, ドメイン適応を行う場合は学習の終盤で上がっている．しかし, Ochahouse-Syn の学習はうまくいかなくなってしまっていることが読み取れる．一方, 図 10(c)(d) では, TSN ベースモデルの Temporal Stream ではドメイン適応の有無にかかわらず, Ochahouse-Syn の学習が進むにつれ Ochahouse-Real の精度が向上しており, またドメイン適応を行う場合の方が Ochahouse-Real の動作識別の



(a) Spatial Stream　(b) Spatial Stream + DANN

図 6　UMAP による特徴量の可視化 (TSN Spatial Stream)



(c) Temporal Stream　(d) Temporal Stream + DANN

図 7　UMAP による特徴量の可視化 (TSN Temporal Stream)

検証精度が向上している．また，図 11(b)(c) では，損失が十分に下がりきっていないため，まだ学習途中である可能性がある．

5.4　考察

3D ResNet ベースモデル，TSN ベースモデルの Spatial Stream，Temporal Stream いずれの場合でも，ドメイン適応によって Ochahouse-Real の動作識別の検証精度が向上したことから，ドメイン適応が有効であることがわかった．しかし，テスト精度は下がってしまっていることから，検証データに過学習している可能性がある．また，Ochahouse-Real の動作識別精度が TSN ベースモデル (Temporal Stream)，TSN ベースモデル，3D ResNet ベースモデル，TSN ベースモデル (Spatial Stream) の順に高いことから，Ochahouse-Real と Ochahouse-Syn の動画像間では，オプティカルフローの違いよりも RGB 画像フレームの違いが大きいこと，つまりモーションの違いよりも色や形状・質感といった見た目の違いの方が大きな影響を与えていると考えられる．よって，これらの違いから起こるドメインシフトに対応することで，動作識別精度をさらに改善できる可能性がある．また，現時点では TSN ベースモデルにドメイン適応を行う学習手法よりもドメイン適応を行わずに TSN の Temporal Stream のみで学習を行う方が動作識別精度が高い点からも，ドメイン適応を活用する学習手法については改善の余地があると考えられる．本研究においてドメインの混合に用いている敵対的学習は安定的に高性能に学習することが難しいことが知られているため，学習を安定化・高性能化させることにより，改善できる可能性がある．

6.　まとめと今後の課題

本研究では，ラベルなし実動画像データの解析に向けた合成動画像データ活用方法について検討した．人間の動作を収録した実動画像データ Ochahouse-Real と合成動画像データ Ochahouse-Syn を用いて実験を行った結果，人間が見ると合成動画像データは写実的であり動作識別できるにもかかわらず，これらのデータ間の違いは 3D ResNet ベース，あるいは TSN ベースのニューラルネットワークにおいて大きく，合成データのみでの学習では十分な精度で実データの解析ができないことがわかった．また，データ間の違いは動画像の人の動きによりも，画像の形状や色・質感といった見た目にあること，ドメイン適応を用いる学習手法はさらに改善できる可能性があることがわかった．

今後は敵対的学習を安定化・高性能化させる手法を取り入れ，モデルの精度向上を目指す．

謝辞　この成果の一部は，JSPS 科研費 JP19H04089，JP19K11994，JP18K11488 及び，2021 年度国立情報学研究所公募型共同研究 (21S0602) の助成を受けたものです．

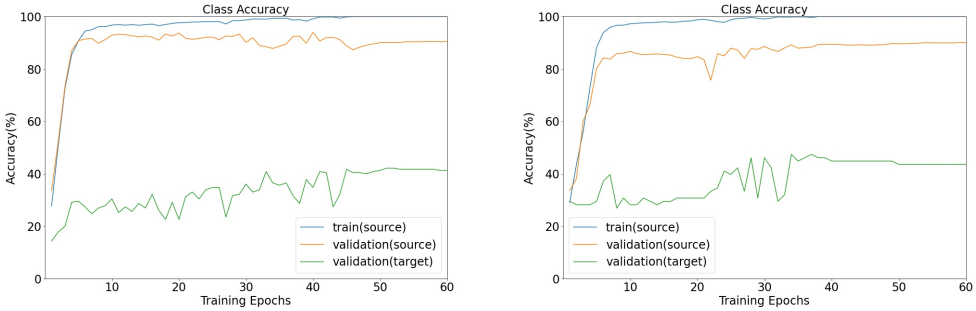
参考文献

[1] Guangchun Cheng, Yiwen Wan, Abdullah N. Saudagar, Kamesh Namuduri, and Bill P. Buckles. Advances in human action recognition: A survey. *ArXiv*, abs/1501.05964, 2015.

[2] D. Wu, N. Sharma, and M. Blumenstein. Recent advances in video-based human action recognition using deep learning: A review. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2865–2872, May 2017.

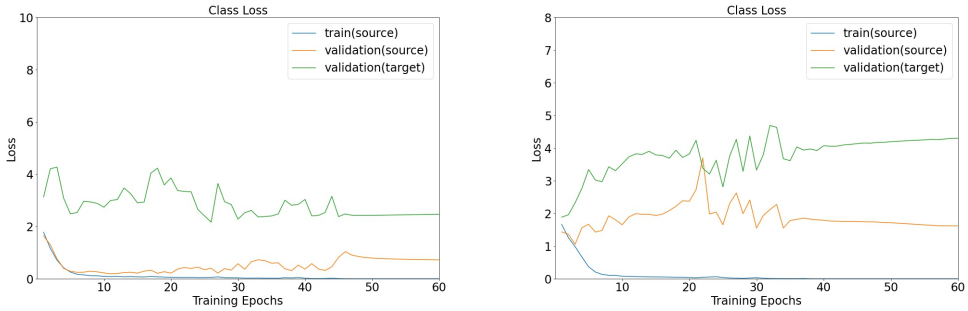
[3] C. Takasaki, A. Takefusa, H. Nakada, and M. Oguchi. A study of action recognition using pose data toward distributed processing over edge and cloud. In *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 111–118, 2019.

[4] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE in-*



(a) 3D ResNet　　(b) 3D ResNet + DANN

図 8　3D ResNet ベースモデルでの学習の様子 (クラス分類精度)



(a) 3D ResNet　　(b) 3D ResNet + DANN

図 9　3D ResNet ベースモデルでの学習の様子 (クラス分類損失)

ternational conference on computer vision, pages 843–852, 2017.

[5] Fereshteh Sadeghi and Sergey Levine. Cad^{2}rl: Real single-image flight without a single real image. *ArXiv*, abs/1611.04201, 2016.

[6] Joshua Tobin, Rachel H Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[7] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtualworlds as proxy for multi-object tracking analysis. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, 2016.

[8] César Roberto de Souza, Adrien Gaidon, Y. Cabon, and A. Peña. Procedural generation of videos to train deep action recognition networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2594–2604, 2017.

[9] D. Vázquez, A. M. López, J. Marín, D. Ponsa, and D. Gerónimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):797–809, 2014.

[10] J. Xu, S. Ramos, D. Vázquez, and A. M. López. Domain adaptation of deformable part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2367–2380, 2014.

[11] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[12] Pau Panareda Busto, Joerg Liebelt, and Juergen Gall. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 14.1–14.12. BMVA Press, September 2015.

[13] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[14] 儀井葉那, 竹房あつ子, 中田秀基, and 小口正人. 室内動作認識のためのドメイン適応による合成データ活用の検討. 第 *13* 回データ工学と情報マネジメントに関するフォーラム *(DEIM2021), E24-5*, pages 1–8, 2021.

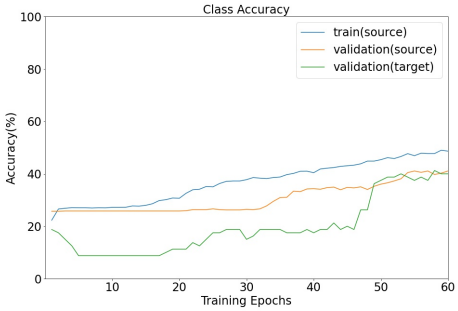
[15] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 20–36, Cham, 2016. Springer International Publishing.

[16] M. Ballout, M. Tuqan, D. Asmar, E. Shammas, and G. Sakr. The benefits of synthetic data for action categorization. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

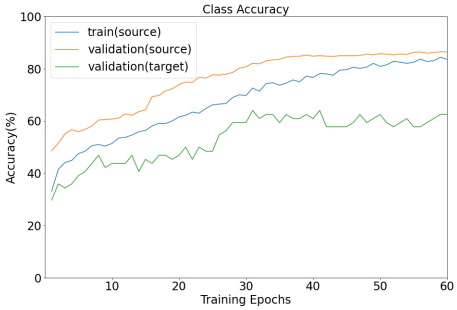
[17] Hilde Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb51: A large video database for human motion recognition. pages 2556–2563, 11 2011.

[18] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

(a) TSN (Spatial Stream)



(b) TSN+DANN (Spatial Stream)

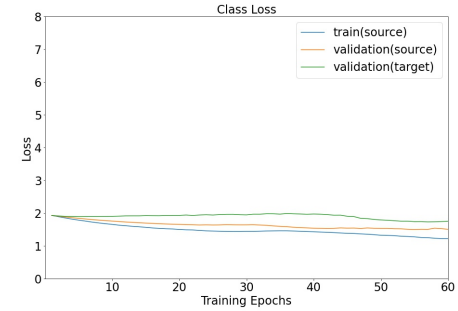


(c) TSN (Temporal Stream)

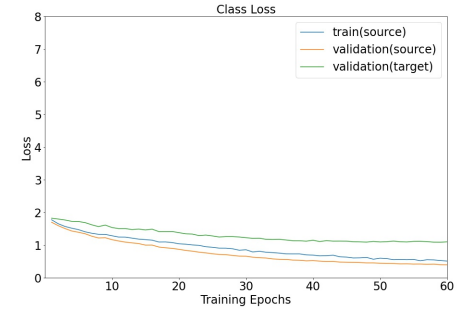
(d) TSN+DANN (Temporal Stream)

図 10 TSN ベースモデルでの学習の様子 (クラス分類精度)

(a) TSN (Spatial Stream)



(b) TSN+DANN (Spatial Stream)



(c) TSN (Temporal Stream)

(d) TSN+DANN (Temporal Stream)

図 11 TSN ベースモデルでの学習の様子 (クラス分類損失)

[19] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[20] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recogni-*

tion (CVPR), pages 2962–2971, 2017.

[21] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[22] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11815–11822, Apr. 2020.

[23] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 678–695, Cham, 2020. Springer International Publishing.

[24] OchaHouse Project Page. http://siio.jp/?OchaHouse.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[26] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, T. Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2019.

[27] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.