室内における動作識別のための合成動画像データセット構築の検討

礒井 葉那[1] 竹房 あつ子[2] 中田 秀基[3] 小口 正人[1]

概要：近年ディープニューラルネットワーク (DNN) により動画像から人間の行動を分析することが可能になり, 一般家庭で老人や子供の見守りなどに応用することが期待されている. しかし, 室内における人間の行動解析のためのデータセットは現状不十分であり, またそのようなデータセットを現実の動画像で作成するには多大な手間やコスト・プライバシといった課題がある. 我々は，既発表研究で人間の室内行動解析のためのデータセットの構築, および現実の動作解析のための合成動画像の生成方法を確立することを目指し, Unity を用いて合成動画像データセットを試作・評価した. 本研究では, 試作したデータに追跡カメラでの撮影・学習時のクラス数の均一化という改良を行い, 実写動画像 STAIR Actions を用いて評価した. その結果, 作成した合成動画像の改良により, 本合成動画像における動作分類の精度が改善したこと, さらに動作などのバリエーションを増やすことが必要であることがわかった.

Consideration of Synthetic Video Dataset for Action Classification in Living Space

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) により動画像から人間の行動を分析することが可能になり [1][2][3], 一般家庭における高齢者や子供の見守りなどへの応用が期待されている. DNN を用いた学習では, 大量かつ多様な学習データが必要となるが [4], データセットを現実の画像で作成するには多大なコストを要する上に, 実際の人が写る動画像を公開・利用することにはプライバシ上の問題がある. そのため, 現状実際の行動解析システムの研究に用いることができるベンチマークデータセットは存在しておらず, このことが屋内行動解析に関する研究の発展の妨げとなっている. 一方, 物体検出などの一部のコンピュータビジョンに関する研究分野では, 現実のデータ収集の問題に対応するために CG で合成画像を生成して学習データとする試みが行われており [5][6][7], 合成データが現実のデータの解析に有効であることが示されている. そこで我々は, 人間の室内行動解析のための合成データセット構築を目指し, 仮想空間内の部屋で人が動く様子を部屋の四隅上から撮影した合成動画像データの作成を提案し試作した. しかし, その時点では作成した合成動画像は現実の動画像データ STAIR Actions[8] の解析に用いるには不十分であり, ランダムと同等な解析精度であった [9]. 各動作ごとの学習データ数が不均一であったこと, また動画像中の人が映る大きさが小さいことが解析精度が不十分な原因の一つであると考えられた.

そこで本研究では, 合成動画像の作成方法を改良し, 各動作の動画像数の均一化, および仮想空間内のカメラに人の追跡・ズーム機能を追加した. 実験により, 動画像数の均一化による解析精度の改善は確認できなかったが, 追跡・ズーム機能によって合成動画像における動作識別精度を改善できることがわかった. さらに, 実写動画像データセット STAIR Actions での評価から, 作成した合成動画像は多様性が不十分であること, 実写データの比較対象として STAIR Actions だけでは不十分であり, 実際に室内にカメラを設置して撮影したような実写データとの比較が必要であるということがわかった.

本論文では以降, 2 章では合成動画像に関する先行研究について, 3 章では作成した合成動画像についての概要と今回実装した追跡・ズーム機能について, 4 章では実験に使用する動作判別システムの概要とデータローダーの改良について述べる. さらに 5 章ではこれらの改良点が及ぼす合成動画像データにおける動作識別精度への影響を実験で

[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

調査し, 6 章では作成した合成動画像データを実写動画像データ STAIR Actions を用いて評価し, 7 章ではまとめと今後の課題について述べる.

2. 合成動画像に関する関連研究

2.1 ドメインランダム化

合成データによる学習では, 時刻, 天候などのドメインをランダム化してデータセットを多様化することで現実のデータの解析に対応できるようになることが知られている.

Fereshteh Sadeghi らはシミュレーションで作成した画像のみを用いて画像から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 DNN でロボット制御を行うことに成功した [10].

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った. これにより, 単純な環境において合成画像のみで学習した DNN でドメイン適応を行わずに現実世界での高精度な物体検出に初めて成功した [6]. [6] に基づき, Gaidon らは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" を生成した [11]. 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.

さらに, Cesar らは天候や照明などがランダム化された動作分類のための合成動画像を生成し, 静止画像だけでなく動画像においても合成データが有用であることを示した.

これらの研究は分類や検出といった基本的な機械学習タスクに対応するものであるが, 本研究ではこれらを応用し, 実際の室内行動解析システムに適用可能な合成データの生成を目指す.

2.2 センサでの劣化を模した加工

センサでのバイアスがディープニューラルネットワークの精度の低下をもたらすことがわかっている [6]. Alexandra Carlson らは, センサでの劣化を模した加工を合成データに施すことで, 現実のデータの解析精度が向上することを実験により明らかにした. 本研究でも同様に, センサでの劣化を模して, ノイズ・ぼかしを施した動画像を生成している.

3. 作成した合成動画像

3.1 概要

行動解析のためのデータセットの作成に Unity [12] を用いる. Unity とは, Unity Technology 社が提供するゲームエンジンであり, 無料で様々なプラットフォーム上で動かすことができるという特徴がある. Unity には独自のコミュニティがあり, Asset Store でユーザが提供する多くのアセットが公開されているため, それを素材として新たな動画像の作成が容易であるという利点がある.

図 1: 座る様子（上）と立ち上がる様子（下）

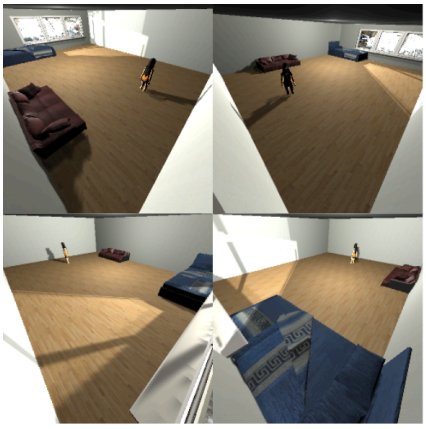

図 2: 作成した合成動画像の 1 フレームの例

作成した動画像では, 部屋の中を人型モデルがランダムに歩き回る・立ち止まるの動作をし, ソファ前に来ると座り, 数秒後に立ち上がる動作を繰り返す. データセットでは, 部屋の四隅上から 256 * 256 ピクセル, 5fps で撮影したものを用意した. 座って立ち上がる様子を図 1 に, 四隅から撮影した動画像の 1 フレームを図 2 に示す.

3.2 現実の動画像に似せるための前処理

空間内の 2 つの照明がランダムに明るさを変えながら, ランダムに移動することで, 照明条件の変更を表現した. 照明の明るさを, 部屋全体が明るく見える程度から, 薄暗く見えにくい状態まで変化させ, 照明自体も部屋の内部をランダムに移動させた.

また, 式 (1) のようにモデル化したガウスノイズを適用した.

$$I_{noise}(x,y) = max(min(I(x,y) + \eta_{gauss}, 0), 255) \quad (1)$$

図 3: ノイズ・ぼかしを付与した 1 フレームの例

ここで $I_{noise}(x, y)$ は処理後の位置 (x, y) における画像の値, $I(x, y)$ は元の画像の位置 (x, y) の値, η_{gauss} はガウス分布に基づく値である.

さらに, 式 (2) で表現するガウスフィルタを適用し, ぼかしを施した. $I_{blur}(x, y)$ は処理後の位置 (x, y) の画像の値, $K(m, n)$ は二次元ガウス分布に基づくカーネルである.

$$I_{blur}(x, y) = \sum_m \sum_n I(x + m, y + n)K(m, n) \qquad (2)$$

ノイズ・ぼかし処理を施した 1 フレームを図 3 に示す.

3.3 追跡カメラ

本稿では, カメラに人物追跡・ズーム機能を追加した. これは, 動画像のフレームの中心に人がより大きく写ることにより, 高精度に動作分類ができるようになるかを調査するためのものである.

追跡は, アニメーション中に動き回る人の座標を取得し, カメラの方向を動的に設定することで実現している. さらに, 取得した座標からカメラとの距離を計算し, カメラが人を一定の大きさ捉え続けるように画角を設定しズームする. 画角とはカメラで撮影される写真に写される光景の範囲を角度で表したものであり, 次の式のように計算した.

$$FoV = 180 * 2 * \arctan\left(\frac{k}{d(camera, human)}\right)$$

ここで FoV は画角, $d(camera, human)$ はカメラと人モデルとの距離, k は定数である. 追跡カメラによる動画像の 1 フレームの例を図 4 に示す.

4. 実験に使用する動作判別システム

動画像は 5fps で撮影し, 各フレームを 112*112 ピクセルで保存する. それらの連続した 16 枚をまとめて 1 つの入力データとし, r3d-resnet [13] で動作を学習させ, 歩く・立ち止まる・座る・座っている・立ち上がるという 5 つの動作に分類する. 学習用データ・検証用データ・テスト用データ数

図 4: 追跡カメラで作成した合成動画像の 1 フレームの例

表 1: フレームにつけられたラベルの内訳

	フレーム数	動画数 (修正前)	動画数 (修正後)
walking	8928	536	6591
standing	2992	171	1547
sitting down		39	1492
standing up	4208	30	1461
sitting		216	4103

はそれぞれ 1000, 300, 300 である. 実装には PyTorch[14] を用い, 損失関数にはクロスエントロピー誤差の実装である torch.nn.CrossEntropyLoss を, 最適化手法は確率的勾配降下法の実装 torch.optim.SGD を採用した. 学習には産業技術総合研究所の AI 向けクラウドシステム ABCI を利用した.

4.1 データローダーの変更

既発表研究 [9] では, 連続したフレームを 16 枚ずつに区切り, 各々を 1 動画像としてそれらに動作ラベルをつけていた. 動作ラベルは, 各フレームにおける Unity アニメーションでのアニメーション State の変化から作成していた. 例えば, フレーム間で「walking」から「sitting down」に遷移していればこの動作は「座る」であるというラベル付けをしていた. しかし, アニメーション State および動作ラベルの内訳を調査すると, 取り出される動作の量に偏りがあることがわかった (表 1). また, 座る動作に入った直後から動画像が始まっている場合は, 実際には立っている状態から座るように見えても「座っている」のラベルがつけられていた. そのため, アニメーション State「sitting down」のうち座る時間が約 3 秒, 座っている時間が約 6 秒, 立ち上がる時間が約 3 秒程度であるにもかかわらず動作が 39, 30 216 個に分けられており, このアニメーション State「Sitting Down」から「座る」「座っている」「立ち上がる」の振り分け方が適切でないことがわかった.

そこで, 次の 3 つの変更を加えた. まず, 各クラスでの動画像数が一定になるように, 動画像を読み込む際に余分なデータは切り捨てることにした. 次に, 切り捨てを行ったと

表 2: データローダー変更によるテスト精度の変化 (%)

	テスト精度（変更前）	テスト精度（変更後）
何もなし	94.3	95.7
ノイズ付与	95.6	96.6
ぼかし付与	95.1	92.2
照明変化	89.1	95.2
全て付与	91.3	90.5

表 3: カメラ変更によるテスト精度の変化 (%)

	テスト精度（変更前）	テスト精度（変更後）
何もなし	95.7	99.9
ノイズ付与	96.6	99.9
ぼかし付与	92.2	100.0
照明変化	95.2	99.8
全て付与	90.5	99.7

きにデータが足りなくなることを防ぐため, 動画像データ間でフレームの重複を許すように変更した. つまり, フレーム数が 16000 であれば, 動画像数は $16000 - 15 = 15985$ になる. そして, アニメーション State に加えて人の腰の高さも取得し, その変化量から「座る」「座っている」「立ち上がる」の振り分けを行うことにした. これらの変更により取り出せる動画像数は表 1 のようになり, 全ての動作クラスで 1,400 個以上の動画像データを取り出すことができるようになった.

5. 合成動画像データセットの動作識別実験

5.1 データローダー変更

作成した合成データにおいて, データローダーの変更が動作判別に与える影響を調査する. この時, 照明条件の変化・ノイズ・ぼかしの付与による影響も調査する.

データローダー変更前と変更後の合成データでのテスト精度の比較を表 2 に示す. 表 2 より, データローダーの変更によって, あまり精度は変化しなかったことがわかった. 精度が向上したデータと低下したデータがあることから, この差は初期値によるものだと考える. また, 同様に照明変化・ノイズ・ぼかしの効果は確認できなかった.

5.2 カメラ変更

カメラに追跡・ズーム機能をつけることによって, 動作判別に及ぼす影響を調査する. また, 照明条件の変化・ノイズ・ぼかしの付与による影響調査する.

学習データごとの合成データでのテスト精度を表 3 に示す. どのデータにおいても精度が向上していることから, 動画像中に人が大きく写ることによってこの合成データにおいてより動作の特徴を学習しやすくなったと言える.

6. STAIR Actions を用いた評価

実写動画像データセット STAIR Actions[8] を用いて作成した合成動画像を評価する. 本研究で作成した動画像は特定の動作を学習するために切り取ったものではなく, 部屋の中で様々な動きをする人を撮影するものであるが, 現在そのような実写動画像を用意することができないため, 日常動作を集めたという点から STAIR Actions を選んだ. 本合成動画像とは異なり, STAIR Actions は人の動作を学習させるためにクラウドソーシングで収集された動画像であり, 動作を行う人間は動画像の中心部に大きく写っている. STAIR Actions に含まれている動作クラスと本合成動画像データに含まれている動作クラスで共通するものは「歩く」「座る」「立ち上がる」の 3 クラスであるため, 実験はこれらの 3 クラスの動作分類を行う. STAIR Actions データセットのこれらの 3 クラスの動画像の 1 フレームの例を図 5 に示す.

図 5: STAIR Actions ビデオデータの 1 フレームの例（上から walking, sitting down, standing up）

6.1 STAIR Actions でのテスト

追跡カメラで照明条件の変化・ノイズ・ぼかしを施した合成動画像での学習の様子を図 6 に示す. 追跡により合成動画像でも人が中心に大きく写るようになったが, STAIR Actions 判別ができていないことは同様である. よって, 作成した合成動画像のもつ特徴と STAIR Actions のもつ特徴は異なっており, 人が写る大きさのの補正のみでは不十分であると考えられる.

6.2 STAIR Actions での学習

次に, STAIR Actions で学習した DNN で作成した固定カメラの合成動画像データのテストを行った結果を図 7 に, 追跡カメラでテストを行った結果を図 8 に示す. 図 7 より,

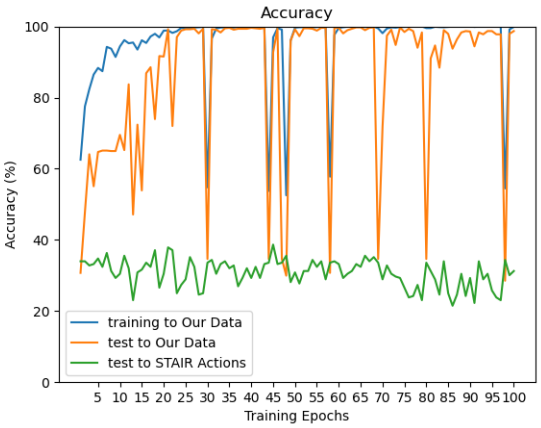

図 6: 追跡カメラデータで学習して STAIR Actions でテストをした精度

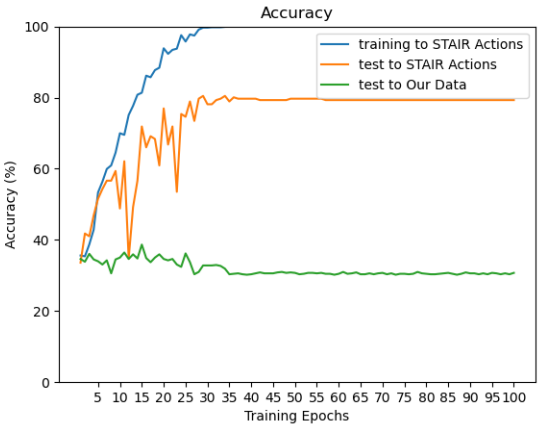

図 7: STAIR Actions で学習して固定カメラの合成データでテストをした精度

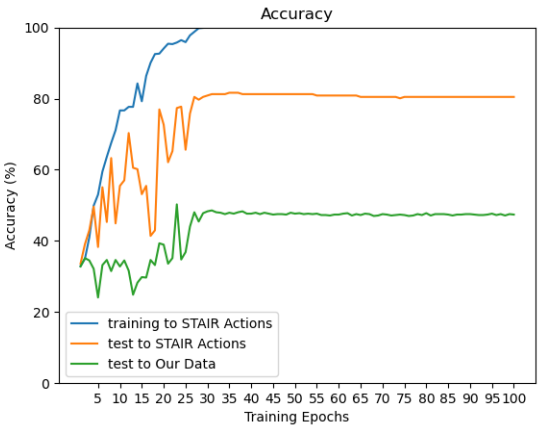

図 8: STAIR Actions で学習して追跡カメラの合成データでテストをした精度

STAIR Actions での動作を学習した DNN では作成した固定カメラの合成動画像の動作判別精度は上がらないことがわかった．図 8 では，追跡カメラの合成動画像の動作判別精度が向上したがそれでも約 50%程度と低くとどまっていることが示された．

6.3 考察

これらの実験から，作成した合成動画像と STAIR Actions とでは性質が異なっており，同じ動作を行う動画像であっても同じ特徴を学習することができないということが示唆された．カメラに追跡機能を実装して人が大きく映るようにすることで多少は特徴が近くなるが，それだけでは不十分であると考えられる．異なる要素の一因として，STAIR Actions の方が本合成動画像よりも多様性があることが挙げられる．STAIR Actions では様々な性別・国籍の人が各々の用意した場所で動作を撮影しており，その動作にも人ごとに多様性があるのに対し，本合成動画像では 1 人の人型モデルが 1 つの部屋の中で決められた動作を行っている．よって，STAIR Actions の解析精度を向上させるためには，本合成動画像おいても人の各動作や背景・動作を行う人物におけるバリエーションを増やすなどの改良が必要だと考えられる．また，本研究で想定している屋内の見守り等では定点カメラによる監視が主流であることなどから，実写データの比較対象として STAIR Actions のみでは不十分であると考えられる．

7. まとめと今後の展望

我々は，室内における行動解析のため合成動画像データセットを改良し，また実写動画像データ STAIR Actions を用いて評価した．学習データの偏りを改善した効果は見られなかったが，カメラに追跡・ズーム機能をつけて人が中心部に大きく映るようにすると，合成動画像において動作識別がより高精度でできるようになった．しかし，これらの変更によって STAIR Actions の解析精度が向上することはなく，合成動画像と STAIR Actions のもつ特徴が大きく異なっているということがわかった．また，本研究で想定している屋内の見守り等との撮影方法の違いから，比較対象として STAIR Actions のみでは不十分であることがわかった．

今後は作成した合成データセットを室内の行動解析に利用できるように改良するとともに，実際に室内にカメラを設置して撮影したような実写データとの比較も行っていく．

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] 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] Chikako Takasaki, Atsuko Takefusa, Hidemoto Nakada, and Masato Oguch. A study of action recognition using pose data toward distributed processing over edge and cloud. *the 11th IEEE International Conference on Cloud Computing Technology and Science (CloudCom2019)*, pages pp.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 international conference on computer vision*, pages 843–852, 2017.

[5] Alexandra Carlson, Katherine A. Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. Modeling camera effects to improve visual learning from synthetic data. In *ECCV Workshops*, 2018.

[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] César Roberto de Souza, Adrien Gaidon, Yohann Cabon, and Antonio Manuel López 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.

[8] Yuya Yoshikawa, Jiaqing Lin, and Akikazu Takeuchi. Stair actions: A video dataset of everyday home actions. *ArXiv*, abs/1804.04326, 2018.

[9] 中田 秀基 小口 正人 礒井 葉那, 竹房 あつ子. 室内における日常動作解析のための合成動画像データセット構築に向けて. 第 *12* 回データ工学と情報マネジメントに関するフォーラム, 2020.

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

[11] 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.

[12] Unity. https://unity.com.

[13] Du Tran, Hong xiu Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2017.

[14] 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.