

深層学習を用いた画像中の人物の 向き推定に関する研究

Research on Human Orientation Estimation in Images Using Deep Learning

和田大河† 山本雄平‡ 佐野睦夫†

Taiga Wada

Yuhei Yamamoto

Mutsuo Sano

1. はじめに

現在、スポーツ科学の分野では選手のパフォーマンス向上やチームの戦略分析を目的とした、ICT 活用に関する研究が進められている。そこでは、モーションキャプチャーや加速度センサ、GPS センサなどを用いたセンシング技術による定量的なデータの取得や、ビデオカメラで撮影した映像に対する画像処理などが行われている。実際のスポーツシーンで利用されている一例として映像分析ソフト [1] が存在し、これらのソフトを利用することで得られるフィードバックがトレーニングに活かされている。

スポーツシーンにおける選手のデータを獲得する手法として、トラッキングがある。トラッキングとは、映像から選手を識別し、その位置情報を用いて追跡を行う手法である。トラッキングを用いた既存研究の一例として、単視点で撮影した動画からパーティクルフィルタなどを用いて選手をトラッキング [2] することや、トラッキングのための選手の識別や位置特定の精度を向上させることを目的とし、単視点複眼の映像から深層学習による識別手法 [3] も試行されている。また、トラッキングにより選手の位置や軌跡などの情報を獲得し、それらの情報を用いてプレー分析を行う研究 [4] もあり、そこでは選手の体の向きを軌跡の方向から推定した結果を用いて分析を行っている。しかし、選手の軌跡の方向と実際に選手が向いている方向が同一とは限らないため、分析データには誤りが含まれる。

一方で、画像中の人物の体の向きを推定する研究 [5] が行われているが、8方向 (カメラに対して正面向きを基準として 45 度毎に回転した向き) を正確に推定する技術は未だ実用レベルに至っておらず、更なる研究の発展が期待されている。

本研究では、単一のカメラで撮影した画像中の人物の向きを深層学習により 8方向に分類する手法を検証する。

2. 既存研究

単眼画像から人物の向きを推定する研究も取り組まれている。Tao ら [6] は、単眼画像から DCTHOG 特徴と Part-based random decision forest を用いて歩行者の体向きを推定している。Flohr ら [7] は、ステレオビデオ画像から歩行者の体と顔の位置と向きを同時に推定する確率的手法を提案している。歩行者の体向き推定は、歩行者の体型、服装、所持品、向き、照明環境などの違いによる見えの変化が大きいことが課題となる。

車載カメラ画像から道路シーン中の歩行者の置かれている状況を認識することを目的とした、歩行者の体の向き推定に関する研究 [5] がある。手法としては、単眼カメラ画像から、Convolutional Neural Network (CNN) [8] を用いて歩行者の体向きを推定することで、(S) こちら向き、(N) むこう向き、(E) 右向き、(W) 左向きの 4方向の平均正答率は 84% と高い精度を出している。また、体全体と顔部分のそれぞれからの推定結果を組み合わせた方が、体全体のみから推定した場合より良い成果が得られるという点が考察されている。

これらの既存研究から着想を得て、我々は、深層学習を用いた方向推定であれば、単視点カメラによる映像に対して 8方向の向き推定を行った場合でも、高い精度が出せるのではないかと仮説を立てた。

3. 事前実験

8方向を推定するための手法を選択するために、YOLOv4 [9] と VGG19 [10] のそれぞれで深層学習を行い、構築した独自モデルによる方向推定の試行し、その精度を検証した。本事前実験では、市販の単一のカメラで動画を撮影し、この動画を FFmpeg というフリーソフトウェアで画像化する。その際、画像は 1 秒あたり 30 枚生成される。また、カメラに対して正面の向きを a とし、その状態から反時計回りに 45 度毎に回転した向きを b, c, d, e, f, g, h の 8方向にクラス分類し、学習を行った。

3.1 YOLOv4 を用いた方向推定の精度検証

YOLOv4 では、GitHub 上で公開されている MS COCO データセットを学習したモデル (yolov4.weights) [11] を用いることができるが、本研究目的は 8方向の推定が可能な手

† 大阪工業大学 大学院 情報科学研究科, Graduate School of Information Science and Technology, Osaka Institute of Technology

‡ 関西大学 環境都市工学部, Faculty of Environmental and Urban Engineering, Kansai University

法の検証であるため、独自の学習データを用意した。学習データは撮影した画像(図1)とその画像対して、LabelImgというグラフィカル画像アノテーションツールを用いてアノテーションを行うことで出力されたテキストファイル(図2)である。テキストファイルの中身は、「クラス番号、矩形の中心の x 座標、矩形の中心の y 座標、矩形の幅、矩形の高さ」の情報であり、クラス番号はアノテーションの際に任意に定めた値、その他は画像の大きさに対する相対値で表す。各方向 400 枚毎の計 3,200 枚を学習データとして用意した実験時の学習グラフを図3に、実験結果を表1に示す。学習時の最終 loss 値は 0.060 (図3)となった。表1に示した適合率は、正と予測した画像のうち、実際に正であるものの割合である。テストデータは、学習データには含まれていない画像であり、且つ画像中の人物の服装が異なる画像 721 枚を用意した。また、各クラスのテストデータ数は同一ではない。適合率は最も高いクラスで a (正面向き) の 0.213、最も低いクラスで d (左斜め後ろ向き) の 0.000 となり、このとき、別の特定クラスへの予測結果の偏りが見られた。全体的に適合率が低く、YOLOv4 を用いた検証結果は良好ではなかった。

3.2 VGG19 を用いた方向推定の精度検証

VGG19 とは、深さが 19 層ある畳み込みニューラルネットワーク(図4)である。本事前実験では、ニューラルネットワークライブラリである Keras を用いて実装を行った。また、学習データは撮影した画像の人物領域をトリミングした後、背景のマスク処理を行った画像(図5)である。3.1節の事前実験とは異なり、同じ人物で 3 種類の服装パターンを学習データに用意した。そして、学習データの画像は VGG ネットワークの標準入力サイズである、幅、高さともに 224px にリサイズをした後、VGG19 による学習を試行した。

各方向 400 枚毎(計 3,200 枚)を学習データとして用意した実験時の学習グラフを図6に、実験結果を表2に示す。学習の epoch 数は 100 であり、最終 loss 値は 0.026 (図6)となった。3.1節の事前実験と同様、テストデータには、学習データには含まれていない画像であり、且つ画像中の人物の服装も異なる画像を 1,062 枚を用意した。また、各クラスのテストデータ数は同一ではなく、学習データと同様に人物領域のトリミング、背景のマスク処理と幅と高さともに 224px にリサイズしたものである。適合率は c (左向き) と e (後ろ向き) が 1.000 であるが、a (正面向き) と g (右向き) は 0.000 であり、a (正面向き) の全てのテストデータに対して e (後ろ向き) と推定し、(右向き) の全てのテストデータに対して c (左向き) と推定していた。さらに、他のクラスの適合率も全体的に低い数値となった。



図1 YOLOv4 の学習データ例 (画像)



図2 YOLOv4 の学習データ例 (テキストファイル)

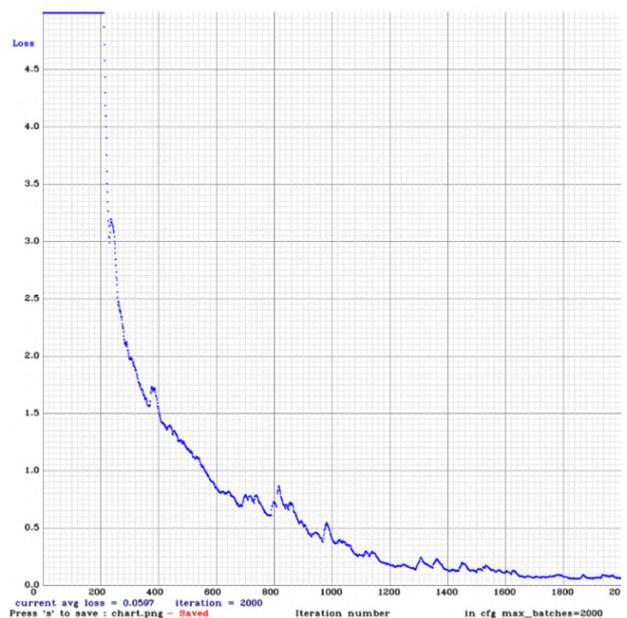


図3 YOLOv4 の学習グラフ

表1 YOLOv4 の事前実験結果

クラス名	適合率
a	0.213
b	0.047
c	0.171
d	0.000
e	0.158
f	0.069
g	0.152
h	0.147
平均	0.120

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_1 (Dense)	(None, 4096)	102764544
dense_2 (Dense)	(None, 4096)	16781312
dense_3 (Dense)	(None, 4)	16388
Total params: 139,586,628 Trainable params: 119,562,244 Non-trainable params: 20,024,384		

図 4 VGG19 の構造



図 5 VGG19 の学習データ例 (事前実験)

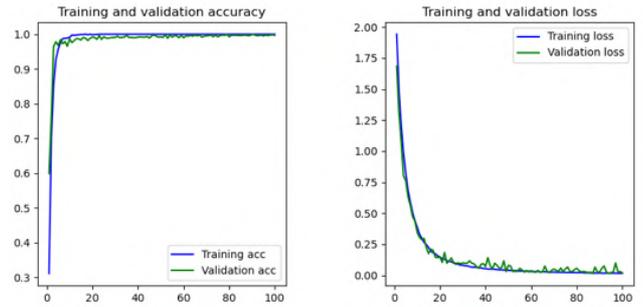


図 6 VGG19 の学習グラフ (事前実験)

表 2 VGG19 の実験結果 (事前実験)

クラス名	適合率
a	0.000
b	0.207
c	1.000
d	0.070
e	1.000
f	0.474
g	0.000
h	0.233
平均	0.373

3.3 考察

YOLOv4 を用いて構築したモデルより、VGG19 を用いて構築したモデルの方が適合率は改善されることが確認できた。しかし、VGG19 を用いた場合でも全体的な適合率はあまり良好とは言えない。また、特定クラスへの偏りが発生しており、a (正面向き) と e (後ろ向き) や g (右向き) と c (左向き) を区別できていないという問題点もある。これらの結果から、単一のカメラで 8 方向を正確に推定するためには、それぞれの方向の特徴をより学習に組み込むための別の手法が必要であると結論づけた。

4 提案手法

VGG19 では学習を行う際に、画像データの入力チャンネル数は 3 がデフォルトとなっているが、4 に設定することができる。これは学習要素として入力される画像データの情報が 3 つのチャンネル (RGB) ではなく 4 つのチャンネル (ARGB) に変更可能という意味である。一般的な画像データは R (Red), G (Green), B (Blue) の 3 色のチャンネルの情報により構成されているが、一方で、画像データには A (アルファ) 値というピクセルごとに設けられた付加情報を表す数値が存在している。特に、透過度として表されることが多く、日常的に利用されている画像データは

RGB 画像であり、A の数値は全てのピクセルが 255 となっているため、画像全体は不透明である。これらより、「4 つ目のチャンネルである A の数値を、各方向を示す特徴量として任意に変更した画像を学習データとし、4 つチャンネルによる学習を行うことで効果的な独自モデルの構築が可能となり、高精度な方向推定が可能になる」と仮説を立てた。

4.1 システムの開発

VGG19 では学習を行う際に、画像データの入力チャンネル数は 3 がデフォルトとなっているが、4 に設定することができる。これは学習要素として入力される画像データの情報が 3 つのチャンネル (RGB) ではなく 4 つのチャンネル (ARGB) に変更可能という意味である。一般的な画像データは R (Red), G (Green), B (Blue) の 3 色のチャンネルの情報により構成されているが、一方で、画像データには A (アルファ) 値というピクセルごとに設けられた付加情報を表す数値が存在している。特に、透過度として表されることが多く、日常的に利用されている画像データは RGB 画像であり、A の数値は全てのピクセルが 255 となっているため、画像全体は不透明である。これらより、「4 つ目のチャンネルである A の数値を、各方向を示す特徴量として任意に変更した画像を学習データとし、4 つチャンネルによる学習を行うことで効果的な独自モデルの構築が可能となり、高精度な方向推定が可能になる」と仮説を立てた。この仮説を立証すべく、開発したシステムを図 7 に示す。本システムは ARGB 画像生成処理、モデル生成処理とデータ推定処理により構成される。

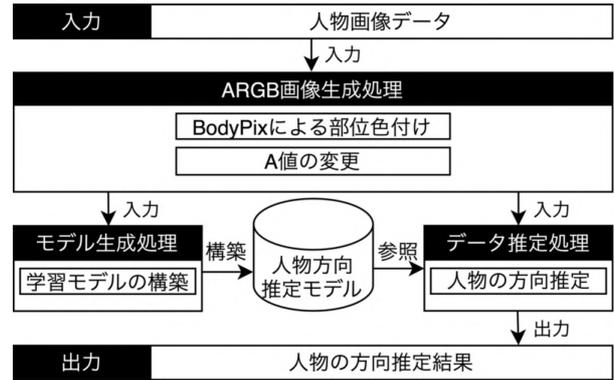


図 7 本システムの概要



図 8 元画像

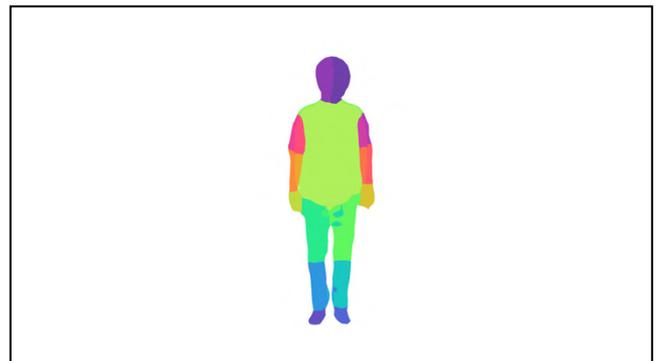


図 9 BodyPix 適用後

- leftFace
- rightFace
- rightUpperLegFront
- rightLowerLegBack
- rightUpperLegBack
- leftLowerLegFront
- leftUpperLegFront
- leftUpperLegBack
- leftLowerLegBack
- rightFeet
- rightLowerLegFront
- leftFeet
- torsoFront
- torsoBack
- rightUpperArmFront
- rightUpperArmBack
- rightLowerArmBack
- leftLowerArmFront
- leftUpperArmFront
- leftUpperArmBack
- leftLowerArmBack
- rightHand
- rightLowerArmFront
- leftHand

図 10 BodyPix の色の判別

4.2 ARGB 画像生成処理

4.2.1 BodyPix による部位色付け

まず、ここで入力するのは単一のカメラでフレーム内に 1 人の人物のみが含まれるように撮影した人物画像データである。次に、この画像に対して BodyPix[12]を適用する。BodyPix とはブラウザと TensorFlow.js により利用できるオープンソースの機械学習モデルであり、人と人の部位を 24 種類にセグメンテーションすることが可能である。セグメンテーションとは画像内のピクセルを意味のある領域ごとにグループ化する技術を指す。つまり、BodyPix では画像内のピクセルを、人と表すピクセルと背景を表すピクセルの 2 種類に分類でき、人を表すピクセルを 24 の部位のいずれかに分類が可能である。

入力した元画像と BodyPix 適用後の画像の一例を図 8 と図 9 に、色付けの判例を図 10[12]に示す。画像に含まれる体の各部位の位置や範囲は各方向ごとに異なるため、図 9 の一例では、A (正面向き) 特有の特徴を RGB 値で表している状態である。

4.2.2 A 値の変更

4.2.1 項での処理により得られた各方向特有の特徴を、元画像の RGB 値は変化させずに保持させるために、4 つ目のチャンネルである A 値を利用する。元画像と BodyPix 適用後の画像はサイズが同一であるため、元画像の各のピクセルの A 値を BodyPix 適用後の画像の同座標のピクセルの RGB 値に従って変更する。

24 種類の各部位の RGB の値と、それにより変更する元画像の A の値についてまとめたものを、表 3 に示す。A の値は学習の際に特徴量として認識されやすくするために、連続する値 (0, 1, 2, ...) ではなく、可能な限り部位間の値の差が大きくなるように任意に決定した。一例として、rightHand (右手) を考える。BodyPix 適用後の画像では rightHand (右手) として検出されたピクセルは R=194, G=219, B=64 となっている。したがって、rightHand (右手) として検出されたピクセルと同座標の元画像のピクセルの A 値は全て 80 に変更する。

同様の処理を全てのピクセルに行った結果を図 11 に示す。部位毎に透過具合が異なっていることが確認できる。これにより、期待していた「元画像の RGB 値は変化させず、各方向特有の特徴を保持している 4 つのチャンネルの画像 (ARGB 画像) の生成が実現できる。

表 3 各部位の色情報

部位	R	G	B	A
rightFace	143	61	178	0
leftFace	110	64	170	10
torsoFront	175	240	91	20
torsoBack	135	245	87	30
rightUpperArmFront	255	78	125	40
rightUpperArmBack	238	67	149	50
rightLowerArmFront	239	167	47	60
rightLowerArmback	255	140	56	70
rightHand	194	219	64	80
leftUpperArmFront	210	62	167	90
leftUpperArmBack	178	60	178	100
leftLowerArmFront	255	115	75	110
leftLowerArmback	255	94	99	120
leftHand	217	194	49	130
rightUpperLegFront	40	234	141	140
rightUpperLegBack	28	219	169	150
rightLowerLegFront	47	150	224	160
rightLowerLegback	65	125	224	170
rightFeet	99	81	195	180
leftUpperLegFront	96	247	96	190
leftUpperLegBack	64	243	115	200
leftLowerLegFront	26	199	194	210
leftLowerLegback	33	176	213	220
leftFeet	84	101	214	230

4.3 モデル生成処理

4 つのチャンネルの学習要素として設定できる VGG19 を用いて、深層学習を行い、人物方向推定モデルを構築する。また、学習データとして入力するのは 4.1 節の処理により生成した 4 つのチャンネルの画像 (ARGB 画像) である。学習の際、3.2 節の VGG19 を用いた事前実験と同様、人物領域をトリミングした後、幅と高さを 224px にリサイズした画像データを利用する。

4.4 データ推定処理

4.2 節の処理により構築した人物方向推定モデルを用いて、テストデータを用意し、画像中の人物の方向推定を行う。テストデータとして入力するのは、4.2 節と同様に、4.1 節の処理により生成した 4 つのチャンネルの画像 (ARGB 画像) の人物領域をトリミングした後、幅と高さを 224px にリサイズした画像データを利用する。これにより、4 つのチャンネルによる方向推定が可能となる。

5. 実験計画

5.1 実験目的

本実験は、3 章で検証した結果に基づき、4 章で示した本システムの処理を行い、構築した独自モデルによる方向推定の精度評価を行うことが目的である。



図 11 元画像の A 値変更後

5.2 実験内容

本実験では、まず、市販の単一のカメラを用いて室内にいる 1 人の人物の 4K サイズ (幅 3,840px, 高さ 1,920px) の動画を撮影した。次に、その動画を FFmpeg というフリーソフトウェアを用いて 1 秒あたり 30 枚の画像切り出しを行う。それらの画像を入力データとし、第 4 章で示した本システムにより処理を行う。また、4.2 節の処理により生成した ARGB 画像は予め学習データとテストデータに分けておく。そして、構築した独自モデルの方向推定の結果を適合率、再現率、F 値で評価する。再現率とは、実際に正である画像のうち、正であると予測されたものの割合を表し、F 値とは、適合率と再現率の調和平均である。また、

3.2 節の VGG19 を用いた事前実験の結果に再現率と F 値の評価を加え、両者の比較を行う。

5.3 実験データ

実験データの一例を図 12, 図 13 に示す。クラス分類の基準は 3 章の事前実験と同様である。データの中には歩いている状態や屈んでいる状態、両腕を大きく動かしている状態などの画像が含まれている。また、4 種類の服装パターンが含まれており、それぞれの服装パターンの画像枚数は各クラス均一となっている。学習データとして各クラス 550 枚 (計 4,400 枚)、テストデータとして各クラス 38 枚 (計 304 枚) の画像を用意した。学習の epoch 数は 400 である。



図 13 実験データ例 (服装パターン)

6. 結果と考察

6.1 実験結果

本システムを用いて構築した独自モデルの学習グラフを図 14, 推定結果の評価を表 4, 3.2 節の事前実験の結果に再現率, F 値の評価を加えたものを表 5 に示す。

事前実験モデルに見られた特定のクラスへの推定結果の偏りは独自モデルでは発生しなかった。それにより、平均適合率が 0.362 ポイント改善し、a (正面向き) と b (左斜め前向き) の適合率は最も高い 0.921 となったが、また、平均再現率は 0.222 ポイント、平均 F 値は 0.201 ポイントの改善が確認できる。

一方で、e (後ろ向き) の適合率は最も低い 0.579 となっており、a (正面向き) と b (左斜め前向き) とは 0.342 ポイントの差が発生していることが確認できる。全体的な評価は改善されているが、クラス間の精度の差は未だ大きいと言える。

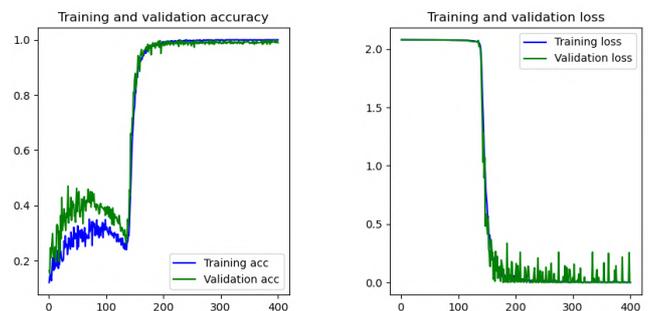


図 14 独自モデルの学習グラフ

表 4 独自モデルによる推定結果の評価

クラス名	適合率	再現率	F 値
a	0.921	0.321	0.476
b	0.921	0.321	0.476
c	0.632	0.276	0.384
d	0.789	0.303	0.438
e	0.579	0.265	0.363
f	0.842	0.311	0.454
g	0.658	0.281	0.394
h	0.631	0.276	0.384
平均	0.745	0.294	0.421

表 5 事前実験モデルによる推定結果の評価

クラス名	適合率	再現率	F 値
a	0.000	0.000	0.000
b	0.207	0.051	0.082
c	1.000	0.189	0.318
d	0.070	0.016	0.027
e	1.000	0.173	0.295
f	0.474	0.087	0.147
g	0.000	0.000	0.000
h	0.233	0.056	0.090
平均	0.373	0.072	0.220

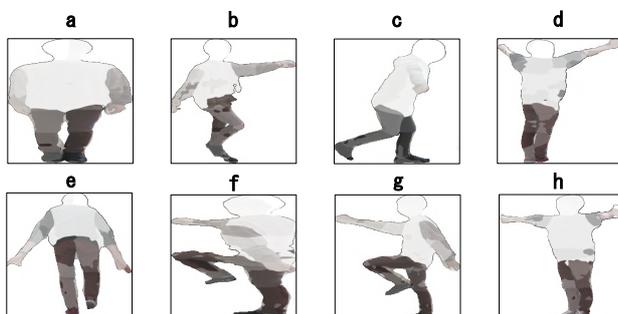


図 12 実験データ例 (各方向)

6.2 考察

事前実験モデルと独自モデルの評価を比較した結果、独自モデルでは大幅な改善が確認できた。これにより、仮説であった「4 つ目のチャンネルである A の数値を、各方向を示す特徴量として任意に変更した画像を学習データとし、4 つチャンネルによる学習を行うことで効果的な独自モデルの構築が可能となり、高精度な方向推定が可能になる」は、本システムにより立証されたと言える。

一方、クラス間の推定精度に差が発生している要因として、4.2.1 項における BodyPix による色付け処理では、部位の誤認識や認識漏れが発生しており、それらの誤った画像データが学習データに含まれていることが考えられる。図 15 はその一例であり、黒枠の部分の色は本来であれば左頭部を示す色であるはずが、背中に色付けされている。また、赤枠の部分は本来は体の一部分であるため色付け対象となるはずが、認識漏れにより、色付けされていない。これらより、可能な限り正確な色付け処理が行われた画像のみを学習データとして用いる配慮が必要である。しかし、BodyPix を適用した全ての画像データに対して誤認識や認識漏れをしていないか確認することは容易ではない。したがって、人物や部位を正確に検出できるような動画を撮影できる環境を用意するという点も重要であると言える。

4. おわりに

本研究では、単一のカメラで撮影した画像中の人物の向きを深層学習により 8 方向に分類する手法を検証した。事前実験では良好な結果は得られなかったが、開発したシステムにより 4 つのチャンネルによる深層学習を行うことで独自の人物方向推定モデルを構築し、8 方向の分類を実現した。

今後の展望としては、まず、BodyPix の誤認識や認識漏れによる誤ったデータを精査することによる推定精度の向上である。次に、本研究では、撮影環境は屋内のみだったため、今後スポーツシーンでの活用を考えていることから、屋外で撮影したデータによる実験は不可欠であると言える。また、同様にスポーツシーンでの活用を想定した場合、1 枚の画像中に複数人が写っていることも考えられるため、そのような画像を本システムに用いることによる結果も検証する必要があるだろう。今後も、関連研究の手法との組み合わせを模索していくことで人物の方向推定精度の更なる向上や、安価なスポーツセンシング製品の開発に繋がると期待する。

謝辞 本研究に際して、山本研究室の皆様には実験データの撮影の協力や有益な助言をいただきました。心より感謝申し上げます。

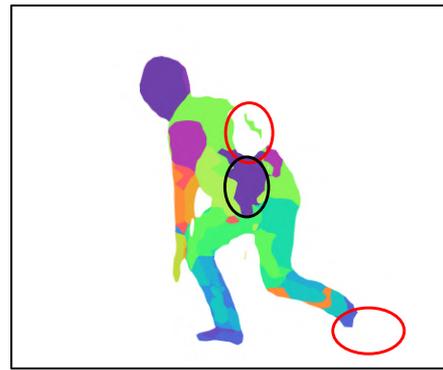


図 15 BodyPix による誤認識の例

参考文献

- [1] FITNESS APOLLO, Hudl 日本公式サイト <<http://hudl.jp/sportscode/>>, (入手 2021.02.17) .
- [2] 西濃拓郎, 滝口哲也. 有木康雄: 複数尤度を用いた 3 次元パースフィルタによる選手の追跡, 画像の認識・理解シンポジウム (MIRU2010), pp.307-312, 2010.
- [3] “権藤聡志. 樽川香澄. 井上智雄. 岡田謙一: スポーツの試合を再現した仮想空間を複数視点で提示する戦略分析支援システムの提案, 情報処理学会研究報告, Vol.2013-DCC-3, No.1, pp.1-5, 2013.
- [4] 田中ちひろ, 山本雄平, 姜文淵, 田中成典, 中村健二, 中島伸介: アメリカンフットボールのパスプレーの移動履歴を考慮したマッチアップ分析に関する研究, 知能と情報 (日本知能情報ファジィ学会誌), Vol.32, No.1, pp.580-89, 2020.
- [5] 熊本浩二, 山田啓一: 道路シーン中の歩行者の置かれている状況の認識～歩行者の体向き推定の基礎検討～, 情報処理学会第 79 回全国大会講演論文集, 2017, 1, pp.429-430, 2017.
- [6] J.Tao et al.:Part-based RDF for Direction Classification of Pedestrians and a Benchmark, ACCV2014 Workshops.
- [7] F.Flohr, et al.:A Probabilistic Framework for Joint Pedestrian Head and Body Orientation Estimation, IEEE Trans.ITS, 2015.
- [8] 内田祐介, 山下隆義: [サーベイ論文]畳み込みニューラルネットワークの研究動向, 信学技報, vol.117, no.362, PRMU2017-105, pp.25-38, 2017.
- [9] Bochkovskiy, A., Wang, Y. and Liao, M.: YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004.10934, 2020.
- [10] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, ICLR 2015, 2015
- [11] YOLOv4, <<https://github.com/AlexeyAB/darknet>>, (accessed 2021-02-23).
- [12] Oved, D. and Zhu, T.: Introducing BodyPix: Real-time Person Segmentation in the Browser with TensorFlow.js, <<https://medium.com/tensorflow/tagged/bodypix>>, (accessed 2021-01-28).