修 士 論 文 概 要 書

Master's Thesis Summary

Date of submission: 01/24/2022

専攻名 (専門分野)	情報理工・ 情報通信専攻	氏 名 Name	藤森 詩織	指 導 教 員 Advisor	渡辺 裕 印 Seal
研究指導名 Research guidance	オーディオビジュアル情報処理研究	学籍番号 Student ID number	CD 5120F072-5		
研究題目 Title	骨格情報を持つ動物画像のデータセット作成手法の研究 Research on methods for creating datasets of animal images with skeletal information				

1 まえがき

近年，深層学習による動物の個体識別や姿勢推定が普及してきている．具体的には，CNN を用いた個体識別や，転移学習による姿勢推定が挙げられる．しかし，データセットの少なさが要因で，これらの成功事例はチンパンジーなどの一部の動物に限られている．そこで本研究では，動物画像のデータセット作成の負担を軽減する事を目的とする．

動物の姿勢推定技術の一つに，DeepLabCut (DLC) がある[1]．DLC を用いると，動画の一部を手動でラベル付けし学習させるだけで，残りのフレームが自動でラベル付けされる．全フレームをラベル付けする必要がなくなりデータセット作成の負担軽減に繋がる．

本研究では，DLC の精度を上げる手法を提案し，データセット作成の負担を軽減させることを目指す．

2 DLC とその課題

DLC は深層学習によって実験動物の動画像から，所望の関節位置を推定し追跡するためのツールとして開発された．DLC は ImageNet によって事前学習されているため，少数の訓練データを学習させるだけで，追跡対象の身体部分を推定し追跡することが可能となった．

しかし，DLC は姿勢推定時の誤検出とブレが課題である．誤検出とは取得したい特徴点ではない点を誤って検出してしまうことである．DLC の精度を確認するための予備実験では，3 割以上のフレームで誤検出が起きるデータがあった．ブレとは特徴点の動きがないフレームで，動きがあるように検出することを意味する．図 1 では GT のグラフに動きがないが DLC グラフには動きがあるフレームが存在する．このような例をブレと定義する

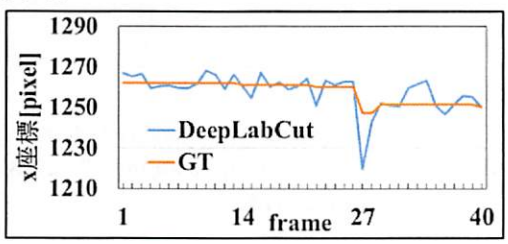


図 1 ブレの例

3 関連技術

提案手法で用いる，IsolationForest (IF)，線形補間および Total Variation (TV) について説明する．

IF は異常検知に用いられる手法である．決定木を拡張させた手法で，作成された決定木のうち最も浅い木に分割された値が異常値であると判定される．

線形補間では，推定点は左右の最近接点を結ぶ線上にあるという仮定をもとに，最近接点の値を直線で結ぶことで推定したい点が推定される．

TV はデータの総変化量を意味する．ノイズ除去の手法として用いられ，入力信号の TV を小さくするような信号を見つけることで，ノイズ除去後の信号を求める．

4 提案手法

本研究では，DLC の課題である誤検出とブレを補正することで，データセット作成の負担を軽減する手法を提案する．提案手法の処理フローを図 2 に示す．本論文における運動時とは歩行のように「動き続けている状態」を指し，静止時とは食事のように「その場で行われる動作」を指す．静止時には，足を一歩前に出す，のような動きも含まれる．

提案手法では，まず DLC により入力動画の姿勢推定を行う．次に，得られた座標値から求めたフレーム間の座標変化率を特徴量として，IF による異常検知を行う．ここで，異常検知の結果から異常フレームのペア (N,M) を作成し，フレーム N からフレーム M-1 を誤検出フレームとする．誤検出フレーム抽出後は，誤検出フレームの前後の正常フレームの座標値を線形補間で結ぶことで，誤検出の補正をする．その後，静止時のみ TV によるブレ補正を行う．

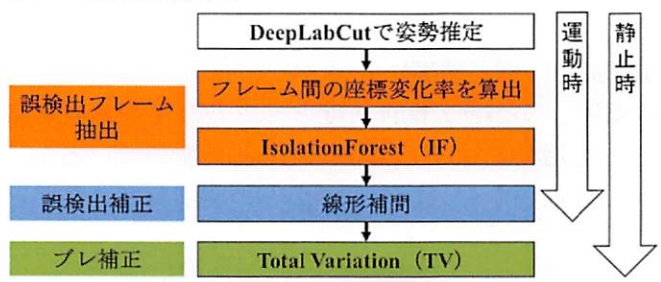


図 2 提案手法の処理フロー

5　提案手法による誤検出・ブレの補正実験

運動時，静止時それぞれ 3 本の動画を使用し，提案手法による誤検出とブレの補正実験を行った．動画は全て異なる馬の動画で，1920×1080[pel]サイズのカラー動画である．本実験では，馬の 4 本の脚のそれぞれつま先，かかと，膝を取得する特徴点として指定した．評価指標は推定した関節位置を正しく推定できた割合を表す PCK スコアを使用した．運動時は DLC による姿勢推定の結果とその結果に対して誤検出補正を行った結果の二つの PCK スコアを比較した．

静止時は，DLC による姿勢推定の結果と誤検出補正後にブレ補正を行った結果の二つの PCK スコアを比較した．また，補正前後で TV を算出し，TV の減少率を求めた．

全 12 ヶ所の特徴点で PCK スコアと，静止時のみ TV の減少率を計測し，その平均を算出した．運動時，静止時それぞれの結果を表 1，表 2 に示す．運動時，静止時ともに補正後は PCK スコアが上がっていることがわかる．また．静止時はブレ補正を行うことで TV は 3 動画で平均 78.94%減少した．

表 1 運動時の PCK スコア比較

	補正前	補正後
運動 1	0.815	0.987
運動 2	0.806	0.905
運動 3	0.896	0.959

表 2 静止時の PCK スコア比較

	DLC	ブレ補正後	TV 減少率
静止 1	0.959	0.979	88.48%
静止 2	0.942	0.978	75.68%
静止 3	0.974	0.994	72.37%

6　有効性の検証実験

提案手法を用いて作成されたデータセットの有効性を確認するための実験を行った．本実験では，正解データ（GT），DLC による推定結果および提案手法によって作成したデータの三つのデータを使用し，RandomForest による馬の行動分類を行い，分類精度を F 値で比較する．本実験では，ラベル付きのデータセットを使用し，このラベルを GT とした．

また，データ作成時に全フレームのうち人手でラベル付けしているフレームの割合をラベル付け負担として評価指標として加える．GT のラベル付け負担は 1.0 となる．

運動時は，脚が地面に付いているか，付いていないかの分類を，かかとの関節角度とつま先の Y 座標を特徴量として用いて行った．静止時は，脚が動いているか，動いていないかの分類を，かかとの関節角度とかかとの Y 座標を用いて行った．運動時と静止時のクラス分類の精度を表 3，表 4 に示し，DLC と提案手法におけるラベル付け負担を表 5 に示す．

表 3，表 4 より運動時，静止時どちらも，提案手法によるデータセットを用いてクラス分類を行った場合のクラス分類精度は，DLC の結果を使用した場合よりも精度が高く，GT を使用した場合よりは精度が低いことがわかる．ラベル付けの負担を比較すると，提案手法によるデータは GT よりも負担を軽減できることがわかる．

表 3 運動時のクラス分類精度

	運動 1	運動 2	運動 3
GT	**0.891**	**0.946**	**0.945**
DLC	0.578	0.897	0.881
提案手法	0.825	0.923	0.935

表 4 静止時のクラス分類精度

	静止 1	静止 2	静止 3
GT	**0.892**	**0.962**	**0.877**
DLC	0.633	0.644	0.647
提案手法	0.785	0.939	0.854

表 5 DLC と提案手法のラベル付け負担

運動 1	運動 2	運動 3	静止 1	静止 2	静止 3
0.170	0.156	0.139	0.085	0.012	0.037

7　結論

本研究では，DLC を使用する際の課題である誤検出とブレを補正することで，動物画像のデータセット作成の負担を軽減することを目的とした．提案手法では，誤検出に対して線形補間を用い，ブレに対して TV を用いることで補正した．実験により，誤検出に対して線形補間を行うと，補正前後で PCK スコアが上昇し，ブレ補正を行うと，PCK スコアを下げずに TV を削減できることがわかった．このことから，提案手法を用いることで，DLC の姿勢推定結果よりも GT に近い特徴点の動きを捉えられることが確認できた．

データのラベル付け負担は，提案手法では GT より負担を軽減でき，その減少率は，運動時は平均 84.5%，静止時は平均 95.5%だった．しかし，提案手法による誤検出補正の実験では，全ての誤検出を補正することはできなかった．より多くの誤検出を補正する手法を検討することで，提案手法により作成されたデータを使用した際のクラス分類精度が向上すると考えられる．

参考文献

[1] A. Mathis, P. Mamidanna1, ” Markerless tracking of user-defined features with deep learning, ” arXiv preprint arXiv: 1804. 03142, 2018.

2021 年度

早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻　修士論文

骨格情報を持つ動物画像のデータセット作成手法の研究

Research on methods for creating datasets of animal images with skeletal information

藤森 詩織

（5120F072-5）

提出日：2022.01.24

指導教員 ： 渡辺 裕

研究指導名 ： オーディオビジュアル情報処理研究

目次

第1章 序論

1.1 研究の背景

近年，深層学習による動物の個体識別や姿勢推定が普及してきている．その代表的な一例として，畳み込みニューラルネットワーク（CNN）の個体識別への適用がある．京都大学と英オックスフォード大学の研究グループは，CNN を用いたチンパンジーの個体識別に成功している[1]．しかし，動物園では，飼育員が膨大な時間と労力をかけて，飼育動物の録画映像を確認し，動物の健康状態や動物同士の関係を分析している．そのため，チンパンジーの個体識別だけでなく，トラッキングや行動分類についても自動で分析可能になれば，飼育員の作業負担の軽減につながると期待されている．

また，転移学習を用いてチンパンジーの姿勢推定が可能になった例もある．これは，人間のデータセットで学習された姿勢推定モデルをチンパンジーに転用することで，チンパンジーでも姿勢推定を可能にした手法である．

しかし，これらの成功事例は一部の動物に限られているのが現状である．その主な要因として，データセットの少なさが挙げられる．何百万種も存在する動物に対して，骨格情報を持つデータセットを作成するには，膨大な時間と手間がかかる．そのため，データセットが存在し，動物の姿勢推定が可能になれば，自然保護や自然科学など多くの分野に活用できると考えられる．したがって本論文ではデータセット作成の負担を軽減することを目指す．

1.2 本研究の目的

動物の姿勢推定技術の一つに，DeepLabCut がある[2]．この技術を用いると，動画の一部を手動でラベル付けし学習させるだけで，残りのフレームが自動でラベル付けされる．全てのフレームをラベル付けする必要がなくなる，という点でデータセット作成の負担軽減に繋がる．

しかし，DeepLabCut のラベル付けには，特徴点を誤検出するなどの課題がある．そこで，本研究では，DeepLabCut の精度を上げる手法を提案することで，動物画像のデータセット作成の負担を軽減させることを目的とする．

1.3 本論文の構成

本論文の構成を以下に示す．

第 1 章は本章であり，本論文の研究の背景および目的について述べる．

第 2 章では，本研究に関連する技術について述べる．

第 3 章では，予備実験（1）から分かる DeepLabCut の課題について述べる．
第 4 章では，提案手法に繋がる予備実験（2）について述べる．
第 5 章では，本研究で提案する手法について述べる．
第 6 章では，提案手法を用いた実験，結果および考察について述べる．
第 7 章では，提案手法で作成したデータの有効性検証のための実験について述べる．
第 8 書では，本論文の結論と今後の課題について述べる．

第2章 関連技術

2.1 まえがき

本章では，本研究で用いる技術である DeepLabCut，IsolationForest，線形補間，Total Variation について述べる．また，提案手法の有効性を検証するための実験に用いた RandomForest について述べる．

2.2 DeepLabCut

DeepLabCut は，ディープラーニングによって実験動物の動画像から，所望の関節位置などを推定し，それらの位置を追跡するためのツールとしてドイツエバーハルト・カール大学テュービンゲンの Alexander Mathis らの研究者グループによって開発された．GitHub からダウンロードして利用することができる[3].

DeepLabCut は特定の実験動物に特化した技術ではなく，追跡対象のデータセットを用意することで，動物の特徴点推定やその追跡を高精度に実行できる．これは，転移学習と呼ばれる手法によって実現されている．DeepLabCut の詳細な構造に関しては 2.2.1 節で述べる．

2.2.1 DeepLabCut の構造

DeepLabCut は Deeper，Stronger，and Faster Multi-Person Pose Estimation Model（DeeperCut）を基に構築されている．

DeeperCut について簡単に説明する．DeeperCut は Elder らによって提案された姿勢推定技術で，ECCV2016 で発表された[4]．同年の CVPR2016 で Elder らが発表した DeepCut を改良したモデルである[5].

DeepCut と DeeperCut はどちらもボトムアップ型の姿勢推定技術である．ボトムアップ型の姿勢推定は，検出する身体部分の候補点となるキーポイントを画像から抽出し，抽出されたキーポイントを，人物ごとにクラスタリングし複数人物の姿勢推定を行う手法である．DeepCut による複数人物の姿勢推定の処理フローを以下に示す．

A. CNN を用いて，入力画像から人物のキーポイントが抽出される．
B. A で抽出されたキーポイント同士の繋がりを，全ての組み合わせ分考える．
C. B で得た全組み合わせから，人物ごとの組み合わせのみ残す．
D. C での残されたキーポイントのうち，主要なキーポイントを出力する．

処理フローの B では，全通りの組み合わせを考えているため，計算量が膨大であり，実行時間が長くなる．この実行時間の長さが，DeepCut の課題である．

この課題にアプローチするために, DeeperCut のキーポイント抽出には Residual Network（ResNet） とよばれる構造が使われている．ResNet は Microsoft Research によって 2015 年に提案されたニューラルネットワークの一種である[6]．画像分類問題において，ネットワークの層を深くすることでより高次元の特徴を獲得することができるとされている．しかし，ただネットワークの層を重ねるだけでは性能の劣化が示された．この劣化は，勾配消失問題が原因とされた．ResNet では shortcut connection を導入することで勾配消失問題を解決し，152 層を重ねることで高い分類性能を達成した．

また, キーポイントを抽出するだけでなく, クラスタリングの補助となる出力を ResNet に追加している．具体的には，肘なら肘，膝なら膝というような身体部分ごとの位置を表すベクトル場と，各キーポイントのペアに関するベクトル場の二つである．ペアに関するベクトル場というのは，あるキーポイントから他の部位への線分を出力し，関節点ペアの距離や角度など，部位ごとの関係性を表すベクトル場である．このペアに関するベクトル場を追加することで，複数人の姿勢推定という困難なケースでも，精度を向上させることに成功している．

さらに，DeeperCut ではステージごとにキーポイントの抽出を行っている．つまり，ステージ 1 では肘と膝, ステージ 2 では顔のように, 段階的に部位抽出を行っている.DeepCut のように一度にキーポイントを抽出すると，キーポイント同士の繋がりに関する全組み合わせを考えるための計算量は，指数的に増える．一方，段階的にキーポイントの抽出を行うことで，キーポイント同士の繋がりの組み合わせを考える計算量を減らすことができる．

以上のように，DeeperCut では，ResNet 構造を採用し，特徴抽出のネットワーク層を DeepCut よりも深くすることで，より高次元の特徴を獲得している．また，ResNet にクラスタリングの補助となる出力を追加することで, DeepCut よりも複数人の姿勢推定に強いモデルとなっている．さらに, キーポイントの抽出を一度ではなく, 段階的に行うことで, DeepCut よりも姿勢推定の際の実行時間を短くすることに成功している．

DeepLabCut では DeeperCut の ResNet 構造に，出力として Deconvolution 層を追加している．Deconvolution 層は各関節位置をそれぞれ異なる画像として表現し，各画像に対する信頼度を学習する．

また，DeepLabCut は ImageNet とよばれるデータセットによって事前学習されている．これは転移学習と呼ばれる手法であり，転移学習によって少数の訓練データを学習させるだけで，追跡対象の身体部分を推定し追跡することが可能となった．

2.2.2 DeepLabCut の検出精度

Alexander Mathis らは，マウスを対象として，DeepLabCut の骨格検出精度を調べる実験を実施した[2].

まずは，人間の手動ラベリングの変動性を定量化するため実験が行われた．全 1080 フレームから成る動画から複数のフレームを抽出し，すべてのフレームに関してマウスの鼻，耳（左右），尻尾の付け根を手動でラベル付けした．同じデータを使用し，手動ラベリングを 2 回行い，2 回の試行間の平均平方二乗誤差（RMSE : Root Mean Squared Error）を求めた．そして，これを"Human Variability"とした．

次に，DeepLabCut による自動ラベリングの実験を行った．DeepLabCut による自動ラベリングの流れは，用意した動画の全フレームのうち，一部を訓練データとして使用し，手動でラベリングする．ラベル付けされた訓練データによって DeepLabCut のネットワークを学習させ，学習済み DeepLabCut で残りフレームを自動ラベリングさせる．自動ラベリング結果と正解ラベルのそれぞれのピクセル値間の RMSE を Human Variability と比較することにより，DeepLabCut による自動ラベリングと人間による手動ラベリングとの精度を比較した．

人間の変動性の定量化実験と同じデータを使用し，データの 8 割を訓練データとしてネットワークを学習させ，残りのデータを DeepLabCut によって自動ラベリングさせたところ，鼻と尻尾は人間レベルで検出ができていることが確認できた．

また，訓練データの割合を変化させると検出精度にどの程度影響するかを調査する実験が行われた．訓練データの割合を 1%，5%，10%，20%，50%，80%とした 6 種類のネットワークを学習させた．訓練データ数の変化による自動ラベリングの RMSE を図 2-1 に示す．自動ラベリングの RMSE を実線で示し，Human Variability を破線で示している．訓練データの割合が 80%の場合，人間と同じレベルの検出精度となっていることがわかる．訓練データの割合が 10%を超えると RMSE は 5 ピクセル以下の値をとっている． 実験で使用した動画中のマウスの鼻のサイズは約 30 ピクセルであり，5 ピクセル以下の誤差になっていることは高い精度で検出ができたと言える．つまり，動画の全フレームの 10%程度を訓練データとして，DeepLabCut を学習させると，指定した特徴点を高い精度で検出できる．

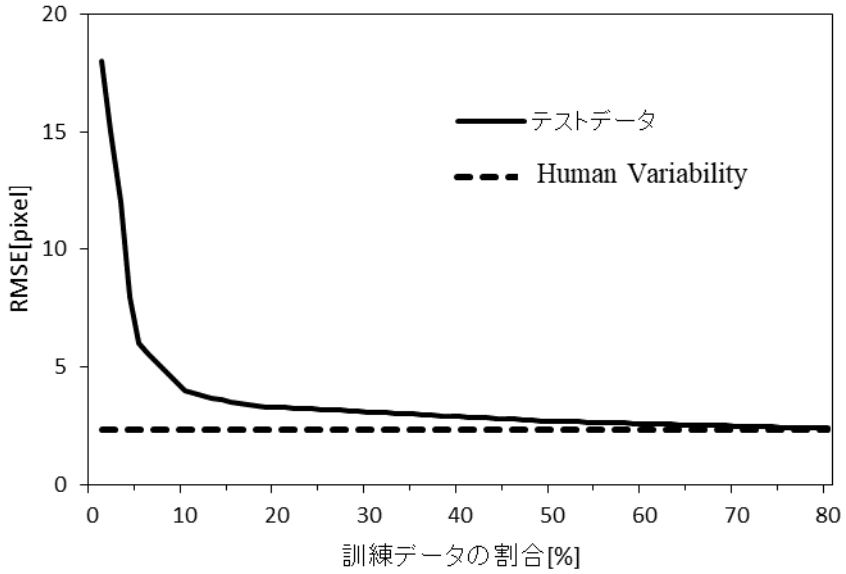


図 2.2.1 訓練データ数と RMSE の関係

出典 : Math, PranavMamidanna1"Markerless tracking of user-defined features with deep learning" arXiv preprint arXiv:1804. 03142, 2018. P. 3

マウスを追跡対象とした実験では，動き回るマウスを真上から撮影した動画が使用されており，検出する特徴点とした鼻，耳そして尻尾の付け根は常にカメラから隠れることはなかった．DeepLabCut のさらなる柔軟性を実証するために，立体環境中を自由に行動するショウジョウバエを追跡対象とした実験も行われた．固定カメラからショウジョウバエを見ると，体の一部がカメラから見えないフレームが存在した．また立体環境中を自由に動き回っているため，体の向きが大きく変化していた．しかし，そのような条件でも指定した身体部分を検出できることが確認された．つまり，身体部位が一時的に隠れることや体の向きの変化は DeepLabCut の精度に大きな影響を与えないことが示されている．

2.2.3 DeepLabCut による複数個体の姿勢推定

2.2.1 節と 2.2.2 節で述べた DeepLabCut は，動画像から 1 個体の関節位置を推定し，追跡するためのツールである．しかし，生物学の多くの実験では，複数の個体間の相互作用を測定する必要がある．複数個体の姿勢推定は，一般的に以下の処理フローで行われる．

A. キーポイントの検出
B. 各個体の識別
C. 各個体の時間軸上の追跡

しかし，DeepLabCut では処理フローA のキーポイントの検出のための構造しか備わっていない. そこで B と C を可能にするために DeepLabCut を改良した DeepLabCut 2.2 を，2021 年に Jessy Lauer らが発表した.

DeepLabCut2.2 の仕組みや構造について簡単に説明する.キーポイントの検出のために，DeepLabCut2.2 ではマルチタスク畳み込みニューラルネットワーク（CNN）を用いる．この CNN ではスコアマップ，局所改良領域，部位関係領域，個体 ID がキーポイントごとに出力される.

この四つの出力について説明する．スコアマップの目的は，画像上のどこにキーポイントが存在するかの判定にある．この判定のために，画像上の画素ごとにキーポイントが存在する確率が算出される.

局所改良領域の目的は，ダウンサンプリングによるズレの補正にある．DeepLabCut2.2 のネットワークではダウンサンプリングが行われる．その際に，画像上のキーポイントの位置がずれるため，補正値を推測する.

部位関係領域の目的は，キーポイント同士の繋がりの表現にある．この技術は OpenPose から着想を得ている[7]．各キーポイントがどの個体に属するかの確率を算出される.

個体 ID では各個体のバウンディングボックスが出力される．後に，各個体の軌道追跡にも利用される．DeepLabCut2.2 では，CNN を用いてこれら四つを求めることで，各キーポイントを検出する.

次に，各個体の時間軸上の追跡について説明する．追跡は，「前フレームと比較して最も近い個体を同一個体とする」という仮定で行われる．具体的には，各個体を長方形で囲み，前フレームの長方形と重なった面積が一定以上なら同一個体とされる.

しかし，オクルージョンの問題などで，追跡できず軌道が途切れることがある．この途切れた軌道をつなぎ合わせるために，グラフ最適化が用いられている．バラバラの軌道は，グラフのノードとして表現される．そしてノードの始点と終点の距離を重みとし，移動方向の情報を利用することで有向グラフを作成する．さらに，それぞれのノードを滑らかに繋ぐためのコスト関数を導入することで，グラフ内の結合を最適化している.

以上の手法により，DeepLabCut2.2 では，複数個体の姿勢推定と追跡を実現している.

2.3 転移学習

転移学習とは，ある領域の知識を，関連する別の領域の学習に転用させる技術である[8]. 転移学習を行うことで，少量のデータしかない領域でも高精度のモデルを作成できる．また，学習時間も短縮できる，などの利点がある.

転移学習の一つであるドメイン適応について 2.3.1 節で説明し，ドメイン適応を動物に活用した例を 2.3.2.節で述べる.

2.3.1 ドメイン適応

ドメイン適応では，ソースドメインから得られた知識をターゲットドメインに適用する．ここで，ソースドメインとは知識の元となる十分な教師ラベルを持つドメインである．また，ターゲットドメインとは十分に情報がないドメインである．ドメイン適応により，情報がないターゲットドメインでも高い精度で働く識別器などを学習できる．

ターゲットドメインに正解ラベルが付いている場合は教師ありドメイン適応，付いていない場合は教師なしドメイン適応と呼ぶ．

2.3.2 ドメイン適応の動物への活用例

独ルプレヒト・カール大学ハイデルベルク、米 Facebook の AI Research、独マックス・プランク進化人類学研究所による研究チームにより，Transferring Dense Pose to Proximal Animal Classes が開発された[9]．これは，動画からチンパンジーの姿勢推定を行い，三次元形状を取得する深層学習のフレームワークである．2.2.1 節で示しているドメイン適応を用いて，人間のデータセットで学習されたモデルを動物に対して適応した．

具体的には，5 万人分のラベル付き人物データセットである DensePose-COCO で学習されたモデルから，チンパンジー画像に対して疑似ラベルを生成する．この疑似ラベルを使用し，モデルの再学習を行う．この再学習を繰り返すことで，動画からチンパンジーを検出し，セグメンテーションマスクの作成が可能なモデルが生成される．

このように，人間とチンパンジーのような骨格構造が近位の動物間では，転移学習を用いることでターゲット動物の姿勢推定をすることに成功している．しかし，人間と四足歩行の動物のように，骨格構造の共通点が少ない動物間の転移学習は困難である．

2.4 IsolationForest

IsolationForest は Liu らによって提案された決定木を拡張させた手法である[10]．この方式では，異常値は正常値よりも少ないであろうという前提をもとに外れ値を検知する．観測値からランダムに値を選択し，選択した値を分割する値をランダムに決め，観測値を分割していく．この分割する値は，ランダムに選んだ観測値の最小値と最大値の間から選ばれる．

観測値の分布と，観測値を分割する木構造の例を図 2.4.1 に示す．異常値は正常値よりも少ないという前提であるため，最も浅い木に分割された値が異常値であるとする．図 2.4.1 のような木構造（IsolationTree）を複数作成し，終端ノードまでのパス長の平均を最終的な異常スコアとする．

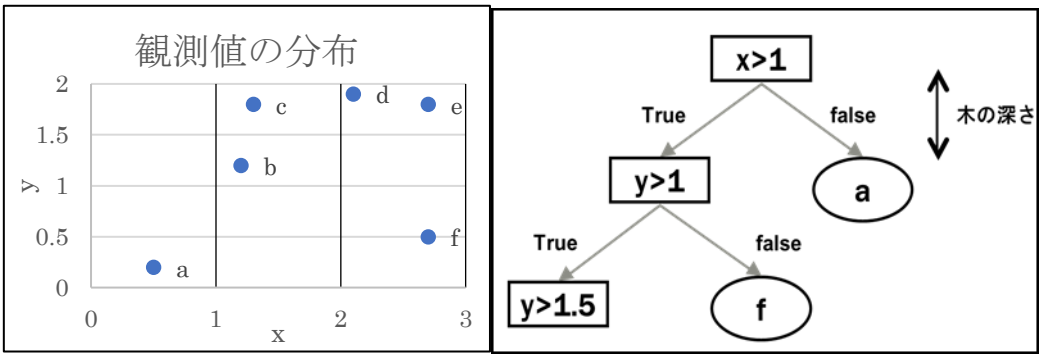


図 2.4.1 IsolationForest による観測値の外れ値検知

2.4.1 IsolationTree の作成方法

IsolationTree は二分木である．二分木とは，葉以外のすべてのノードが二つ以下の子を持つ木構造である．子を持たないノードを葉または外部ノードと呼び，葉ではないノードを内部ノードと呼ぶ．

IsolationTree のノードを T とすると，T は子を持たない外部ノード，または二つの子を持つ内部ノードである．観測値の分布から，n 個のデータ $X = \{x_1, x_2, x_3, \ldots, x_n\}$ が与えられたとき，属性 q と分割値 p がランダムに選択される．そして，X を次の条件になるまで分割する．

A. 木の高さが事前に決めた最大値と同じ高さになる．
B. X=1 になる
C. X が全て同じ値になる

すべてのノードが葉になると木の構成は終了する．この時，外部ノードの数は n,内部ノードの数は n-1 となり，IsolationTree のノードの総数は 2n-1 となる．

2.4.2 異常スコア

異常スコアは、観測値が外部ノードになるまでの平均の深さをそのまま用いるのではなく、二分探索する際の平均の深さで正規化されるため、0 から 1 の値に収まる．

点 x のパス長 h(x)は，ルートノードから外部ノードまでの木を辿り，エッジを数えることで算出される．また，調和数 H(i)は式（2.4.1）で表される．

$$H(i) = log(i) + 0.57721 \qquad (2.4.1)$$

この調和数を用い,IsolationTree 内でのパス長 h(x)の正規化 c(n)は式(2.4.2)で表される.

$$c(n) = H(n-1) - \frac{2(n-1)}{n} \tag{2.4.2}$$

全ての IsolationTree のパス長 h(x)の平均値を E(h(x))とすると, x の異常スコア s(x,n)は, 式（2.4.3）で表される.

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2.4.3}$$

この異常スコアは 1 に近いほど異常である確率が高く，異常スコア 0.5 未満では異常ではないと判定される．初期設定では，異常スコアが 0.5 より大きい値は，異常だと判定される．さらに，全ての異常スコアが 0.5 の場合は異常なデータが含まれていない.

2.5 線形補間

Alexandre M. Bayen, Timmy Siauw らによる『An Introduction to MATLAB® Programming and Numerical Methods for Engineers』の中で，線形補間は次のように定義される [11].

線形補間では，推定点は左右の最近接点を結ぶ線上にあると仮定する．データが昇順に並んでいる，つまり，$x_i < x_{i+1} \quad (x \in \mathbb{R})$ だとする．この時，式（2.5.1）を満たす点 x での線形補間は，式（2.5.2）のように表される.

$$x_i < x < x_{i+1} \tag{2.5.1}$$

$$\hat{y}(x) = y_i + \frac{(y_{i+1} - y_i)(x - x_i)}{x_{i+1} - x_i} \tag{2.5.2}$$

2.6 Total Variation

Total Variation（TV）とは，データの総変化量を意味する．N 点の一次元信号 u(t)の TV は次のように定義できる.

$$V\big(u(t)\big) = \sum_{t=1}^{N} |u(t) - u(t-1)| \tag{2.6.1}$$

入力信号 u(t)が与えられた時, TV によるノイズ除去の目的は, u(t)よりも TV が小さく, u(t)の近似値を見つけることである．この近似値を v(t)とすると，u(t)と v(t)が近似しているかの指標として二乗誤差の総和を用いる．

$$\mathbf{E}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{N}}(\boldsymbol{v}(\boldsymbol{t}) - \boldsymbol{u}(\boldsymbol{t}))^{\mathbf{2}} \qquad \textbf{(2.6.2)}$$

つまり，TV によるノイズ除去は，信号 v(t)に対する評価式（2.6.3）を最小にすることを目標にする．

$$\boldsymbol{E}(\boldsymbol{u},\boldsymbol{v}) + \boldsymbol{\lambda}\boldsymbol{V}(\boldsymbol{v}(\boldsymbol{t})) \qquad \textbf{(2.6.3)}$$

ここで，$\boldsymbol{\lambda}$は平滑化の程度を決めるパラメータであり，ノイズが大きいほど，大きな値を設定する必要がある

2.7 RandomForest

RandomForest とは，2001 年に Leo Breiman によって提案されたアンサンブル学習の一種である[12].

RandomForest はパターン識別や回帰，クラスタリングに用いられる．複数の決定木を用いて森を構成して識別などを行う．個々の決定木が高い識別能力を持つわけでは無いが，複数用いることで高い性能を得る．

複数の決定木から構成されるが学習過程ではそれぞれの決定木で独立して学習が行われる．アルゴリズムは次の処理フローである．

A. ランダムサンプリング

与えられたデータに対して, ランダムサンプリングを行いN組のデータセットに分ける．

B. 各データセットから決定木の作成

A で分けられたデータセット中に p 個の変数があるとする．この時，p 個全ての変数をノード作成に使用するのではなく，一部のみ使用する．$m = \sqrt{p}$ 程度の m 個の変数をランダムに選び，決定木を作成する．

C. 出力の決定

（回帰のとき）B で作成された N 本の決定木の出力の平均値が最終的な出力.

（識別のとき）B で作成されたN本の決定木の出力で多数決を取り最終的な出力を決定.

2.8 むすび

本章では，本研究で用いる技術である DeepLabCut，IsolationForest，線形補間，Total Variation について述べた．また，有効検証のために実験に用いた RandomForest について述べた．

第3章 予備実験（1）

3.1 まえがき

本章では，DeepLabCut による馬の姿勢推定の実験結果について述べる．そして，実験結果から DeepLabCut をデータセット作成に用いる際の課題を述べる．

実験は，運動時と静止時に分類して行った．本論文における運動時とは歩行のように「動き続けている状態」を指す．一方，静止時とは食事のように「動きはあるがその場で行われる動作」を指す．静止時には，足を一歩前に出す，のような動きも含まれる．

3.2 評価指標

姿勢推定の評価指標である，Percentage of Correct Key-points（PCK）スコアについて述べる．PCK スコアは，推定したい関節位置を正しく推定できた割合である．推定した関節位置と正解の誤差が閾値以内の場合に，推定結果が正しいとされる．そして，この正解率を評価指標とする．

総フレーム数 T の動画における PCK スコアは式（3.2.1）のように表される．

$$\mathbf{PCK@n} = \frac{1}{T}\sum_{i=1}^{T} \delta(\|X_i - Y_i\|_2 < Threshold) \tag{3.2.1}$$

$$Threshold = \sigma \times n \tag{3.2.2}$$

$$\delta(x) = \begin{cases} 1\ (x = true) \\ 0\ (x = false) \end{cases} \tag{3.2.3}$$

ここで，X_iは特徴点の推定値，Y_iは正解値を表し，σは閾値を決める際の基準値を表す．閾値は式（3.2.2）で表され，例えば PCK@0.3 の場合は基準値の 0.3 倍の距離が閾値となる．動物の場合，目と鼻の距離を基準値とする例が多くあるため，本実験でも目と鼻の距離を基準とした．

3.3 運動時の姿勢推定

運動時の実験で使用したデータセットと実験結果について述べる．

3.3.1 データセット

1920×1080[pel]サイズの三つ動画を使用した. 全て歩行中の馬を撮影したカラー動画で, 異なる種類の馬の動画を使用した．それぞれの動画を運動 1，運動 2，運動 3 として，総フレーム数を表 3.3.1 に示す．歩行の速度は，運動 1 が最も早く，運動 3 が最も遅い.

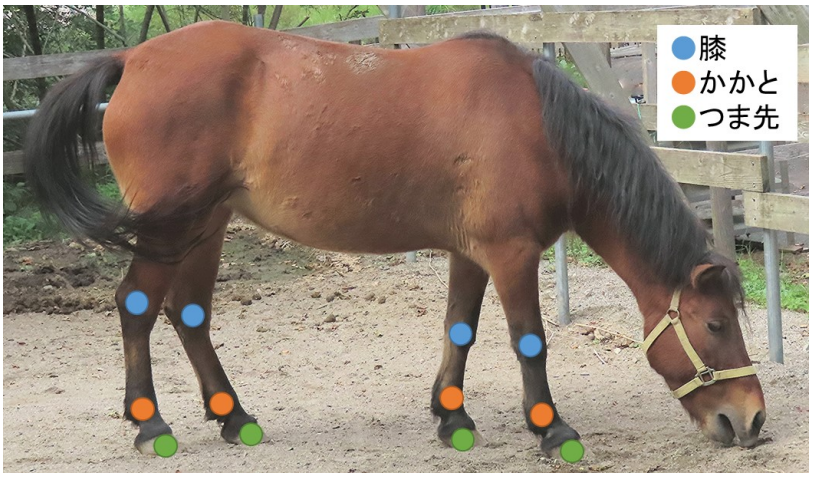
表 3.3.1 運動時のデータセット

動画	フレーム数
運動 1	116
運動 2	121
運動 3	146

3.3.2 実験結果

本実験では，馬の脚のつま先，かかと，膝を DeepLabCut によって取得する特徴点として指定した．具体的には，図 3.3.1 のように左前脚，右前脚，左後脚，右後脚の各足 3 ヶ所ずつ，合計 12 ヶ所の座標データを，DeepLabCut を用いて取得した．そして，被写体である馬の目と鼻の距離を基準とした時の，運動 1，運動 2，運動 3 の PCK スコアを計測した．各計測結果を表 3.3.2，表 3.3.3，表 3.3.4 に示す.

表 3.3.2 から，運動 1 の左前脚と右後脚の膝における PCK スコアは 0.7 を下回っていることがわかる．つまり，DeepLabCut は 3 割以上のフレームで特徴点を正しく推定できなかった．また．表 3.3.3 より，運動 2 の左後脚のかかとにおける PCK スコアも 0.7 を下回っていることがわかる.



出典：写真 AC（https://www.photo-ac.com/）

図 3.3.1 取得する特徴点

表 3.3.2 運動 1 の PCK@0.3

	左前脚	右前脚	左後脚	右後脚
膝	0.776	0.698	0.819	0.698
かかと	0.871	0.845	0.853	0.974
つま先	0.871	0.802	0.707	0.871

表 3.3.3 運動 2 の PCK@0.3

	左前脚	右前脚	左後脚	右後脚
膝	0.884	0.860	0.766	0.784
かかと	0.810	0.843	0.694	0.793
つま先	0.851	0.818	0.802	0.766

表 3.3.4 運動 3 の PCK@0.3

	左前脚	右前脚	左後脚	右後脚
膝	0.912	0.934	0.897	0.986
かかと	0.949	0.890	0.973	0.877
つま先	0.882	0.875	0.836	0.747

3.4 静止時の姿勢推定

静止時の実験で使用しデータセットと実験結果について述べる．

3.4.1 データセット

運動時とは異なる 1920×1080[pel]サイズの三つカラー動画を使用した．それぞれの動画を静止 1，静止 2，静止 3 として，総フレーム数を表 3.4.1 に示す．また，それぞれの動画の特徴を説明する．

静止 1 は総フレーム 235 のうち，初めの 180 フレームは各脚に大きな動きはない．その後の残りの 55 フレームで，各脚一歩ずつ前に動かしている．

静止 2 は，右前脚，左後脚，左前脚，右後脚の順番で一歩ずつ前に踏み出している．一歩踏み出すのにかかるフレーム数は約 15 フレームで，残りのフレームは静止している．

静止 3 は総フレーム 150 のうち，初めの 40 フレームで右前脚と左後脚を一歩前に動かしている．その後 100 フレーム目までは各脚に動きがなく，100 フレーム目以降に左前脚と右後足を一歩前に動かしている．

以上のように 3 本の動画は，静止が中心であるが，各脚が一度は前に踏み出すという動きを含んでいる．

表 3.4.1 静止時のデータセット

動画	フレーム数
静止 1	235
静止 2	100
静止 3	150

3.4.2 実験結果

運動時と同じ 12 ヶ所を特徴点とし，姿勢推定を行った．3 本の動画に対する PCK スコアをそれぞれ表 3.4.2，表 3.4.3，表 3.4.4 に示す．静止 1 の右後脚のつま先のみ PCK スコアが 0.9 を下回っているが，それ以外の特徴点では 0.9 を上回っている．

また，静止 2 の DeepLabCut による推定結果と正解データ（GT）の x，y 座標をそれぞれ図 3.4.1，図 3.4.2 に示す．静止 2 の目と鼻の距離を基準値とした時の PCK@0.3 の閾値も図中に示す．

静止 2 の目と鼻の距離 d は，d=125.45 である．そのため，PCK@0.3 で評価する場合は，GT との誤差が 37.635 のとき，DeepLabCut の推定結果は正しいと判定される．図 3.4.1，図 3.4.2 より，例で示した 40 フレームは推定結果と GT の誤差は PCK@0.3 の閾値以内であり，推定結果は正しいと判定されることが分かる．

表 3.4.2 静止 1 の PCK@0.3

	左前脚	右前脚	左後脚	右後脚
膝	0.996	0.987	0.932	0.94
かかと	0.987	0.991	0.974	0.919
つま先	0.932	0.974	0.974	0.898

表 3.4.3 静止 2 の PCK@0.3

	左前脚	右前脚	左後脚	右後脚
膝	0.900	0.990	0.960	1.000
かかと	0.910	0.960	0.930	0.930
つま先	0.920	0.950	0.940	0.910

表 3.4.4 静止 3 の PCK@0.3

	左前脚	右前脚	左後脚	右後脚
膝	0.920	1.000	0.987	1.000
かかと	0.953	0.973	0.960	1.000
つま先	0.960	0.973	0.960	1.000

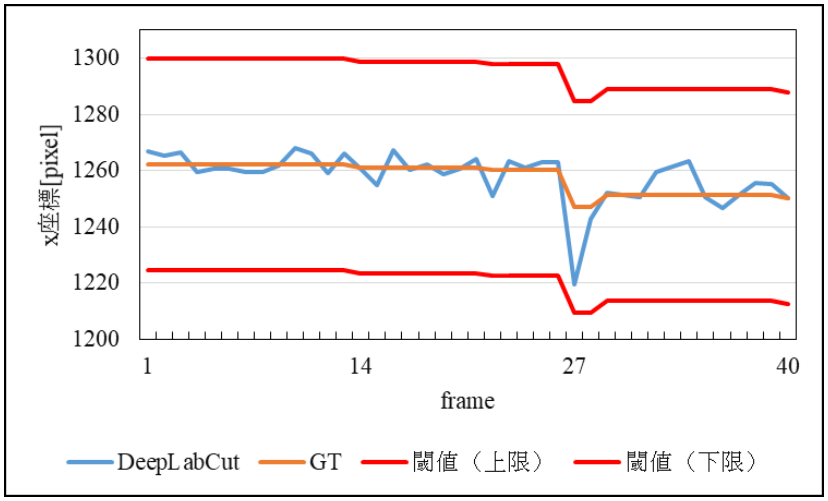


図 3.4.1 ブレの例（x 座標）

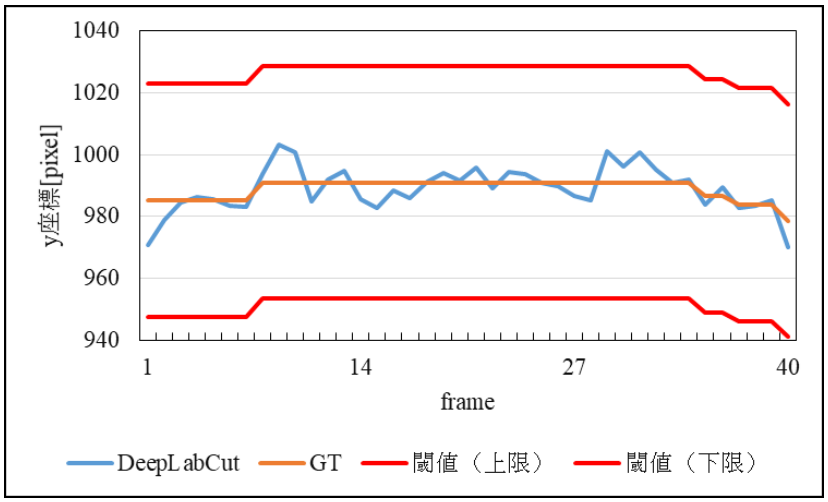


図 3.4.2 ブレの例（y 座標）

3.5 DeepLabCut の課題

使用した 6 本の動画における，全特徴点の PCK スコアの平均を表 3.5.1 にまとめる．運動時の 3 動画における PCK スコアの平均は 0.839 である．つまり，DeepLabCut は平均 16.1%のフレームで特徴点の推定に失敗していることがわかる．これは誤検出が原因である．誤検出とは，取得したい特徴点ではないポイントを誤って検出してしまうことである．この誤検出が運動時の PCK スコアを下げた原因である．

一方, 静止時の 3 動画における PCK スコアの平均は 0.958 で, 運動時と比較すると PCK スコアは高い値となった．しかし，図 3.4.1 と図 3.4.2 で示す GT と推定結果を比較すると，GT で座標変化が起きていないフレームでも，推定結果では座標の変化が起きている

ことがわかる．このように，本来特徴点の動きが無いフレームでも動きがあるように推定されている．本論文では，この動きを「ブレ」と呼ぶ．

この予備実験により，DeepLabCut には「誤検出」と「ブレ」の二つの課題があることが分かった．

表 3.5.1 全特徴点の平均 PCK@0.3

動画	PCK@0.3
運動 1	0.815
運動 2	0.806
運動 3	0.896
静止 1	0.959
静止 2	0.942
静止 3	0.974

3.6 むすび

本章では，DeepLabCut による馬の姿勢推定の実験結果について述べた．そして，実験結果から DeepLabCut をデータセット作成に用いる際の課題として「誤検出」と「ブレ」が存在することを述べた．

第4章 予備実験（2）

4.1 まえがき

本章では，提案手法で用いる線形補間の補間区間を決めるための予備実験について述べる．線形補間について 2.5 節で述べた．

4.2 実験の概要

第 3 章で，「誤検出」が DeepLabCut の課題だと述べた．線形補間を用いることで，この誤検出を補正する．線形補間をする際は，誤検出フレームの前後の正常フレームの座標値を直線で結ぶことで補間する．しかし，各特徴点の動作は直線的ではないため，特徴点座標の正解値の波形における振幅が大きい場合には，線形補間による補正値と正解値の誤差が大きくなる可能性がある．つまり，誤検出フレームが連続し，その誤検出フレーム間における正解値の波形の振幅が大きい場合，線形補間に失敗する可能性がある．

そのため，予備実験（2）では線形補間により補正可能な区間を求めた．予備実験（2）の処理フローは以下の通りである．

A. DeepLabCut の推定結果が，連続で「正しい」と判定された区間を取り出す．
B. A で取り出した区間の内，連続した数フレームを欠損値として扱う．
C. B で定めた欠損値を線形補間によって補間する．
D. 線形補間後の PCK スコアを算出する．
E. 線形補間前後で PCK スコアに変化が無い場合，線形補間に成功したとする．

B で欠損値として扱うフレームを増やしていき，補正可能な区間を求めた．

4.3 実験条件

予備実験（1）では 6 本の動画を対象として，それぞれ 12 ヶ所の特徴点を DeepLabCut により推定した．予備実験（2）では，予備実験（1）で得られた姿勢推定の結果を用いて線形補間を行った．

線形補間を行う区間の定め方について説明する．まず，予備実験（1）で特徴点の姿勢推定を行った結果が，連続で「正しい」と判定された区間を取り出す．次に，取り出した区間のうち，数フレームを欠損値として扱い線形補間を行う．欠損値として扱うフレームを 1 から増やしていき，線形補間に失敗するフレーム数を求める．各動画 12 ヶ所の全ての特徴点で，線形補間可能な区間を求めた．

4.4 実験結果

6 本の動画に対して，線形補間を行う区間を増やしながら，線形補間が可能かどうかを求めた結果をそれぞれ表 4.4.1，表 4.4.2，表 4.4.3，表 4.4.4，表 4.4.5，表 4.4.6 に示す．線形補間に成功した場合は値を「S」とし，失敗した場合は値を「F」とする．表 4.4.1 より，運動 1 では補間フレーム数を 7 に設定すると，線形補間に失敗する特徴点があることがわかる．よって，線形補間可能な区間は 6 フレームと設定する．同様に他の 5 本の動画でも，線形補間に失敗する特徴点がない最大の補間フレーム数を，フレーム補間可能な区間に設定する．各動画のフレーム補間可能な区間を表 4.4.7 に示す．

表 4.4.1 運動 1 の予備実験（2）

補間フレーム		5	6	7	8	9	10
	膝	S	S	S	S	S	S
左前脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	S
右前脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	S
左後脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	**F**	**F**	**F**
右後脚	かかと	S	S	S	S	S	S
	つま先	S	S	**F**	**F**	**F**	**F**

表 4.4.2 運動 2 の予備実験（2）

補間フレーム		5	6	7	8	9	10
	膝	S	S	S	S	S	**F**
左前脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	**F**
右前脚	かかと	S	S	**F**	**F**	**F**	**F**
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	S
左後脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	S
右後脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S

表 4.4.3 運動 3 の予備実験（2）

補間フレーム		5	6	7	8	9	10
	膝	S	S	S	S	S	S
左前脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	S
右前脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S
	膝	S	S	S	S	S	S
左後脚	かかと	S	S	S	**F**	**F**	**F**
	つま先	S	S	S	S	S	**F**
	膝	S	S	S	S	S	S
右後脚	かかと	S	S	S	S	S	S
	つま先	S	S	S	S	S	S

表 4.4.4 静止 1 の予備実験（2）

補間フレーム		1	2	3	4
	膝	S	S	S	S
左前脚	かかと	S	S	S	S
	つま先	S	**F**	**F**	**F**
	膝	S	S	S	S
右前脚	かかと	S	S	S	S
	つま先	S	S	S	S
	膝	S	S	S	S
左後脚	かかと	S	S	S	S
	つま先	S	S	S	S
	膝	S	S	**F**	**F**
右後脚	かかと	S	S	S	**F**
	つま先	S	S	S	S

表 4.4.5 静止 2 の予備実験（2）

補間フレーム		1	2	3	4
	膝	S	S	S	**F**
左前脚	かかと	S	S	**F**	**F**
	つま先	S	S	S	S
	膝	S	S	S	S
右前脚	かかと	S	**F**	**F**	**F**
	つま先	S	S	**F**	**F**
	膝	S	S	S	S
左後脚	かかと	S	S	S	S
	つま先	S	S	**F**	**F**
	膝	S	S	S	S
右後脚	かかと	S	S	S	S
	つま先	S	S	S	**F**

表 4.4.6 静止 3 の予備実験（2）

補間フレーム		1	2	3	4
左前脚	膝	S	S	S	S
	かかと	S	S	S	S
	つま先	S	S	S	S
右前脚	膝	S	S	S	S
	かかと	S	S	S	S
	つま先	S	S	**F**	**F**
左後脚	膝	S	S	S	**F**
	かかと	S	S	S	S
	つま先	S	S	S	S
右後脚	膝	S	S	S	S
	かかと	S	S	S	S
	つま先	S	S	S	S

表 4.4.7 各動画の線形補間可能な区間

	線形補間可能な区間 [フレーム数]
運動 1	6
運動 2	6
運動 3	7
静止 1	1
静止 2	1
静止 3	2

4.5 むすび

本章では，線形補間可能な区間を実験により求めた．本研究では，DeepLabCut の誤検出を線形補間により補正することを目的とする．線形補間を行う区間を増やし，線形補間に失敗することは，誤検出を増やすことにつながる．そのため，誤検出フレームを増やさずに誤検出の補正を行うという条件では，線形補間に失敗する特徴点がないような区間を設定する必要があることがわかった．

第5章 提案手法

5.1 まえがき

本章では，DeepLabCut の課題である「誤検出」と「ブレ」を補正するための手法を提案する．

5.2 提案手法

本研究では，3.5 節で述べた DeepLabCut の課題である「誤検出」と「ブレ」を補正することで，DeepLabCut を用いてデータセット作成するための手法を検討する．

一般的な物体検出や姿勢推定では，正解データと推定結果の多少のズレが許容される傾向にある．例えば，物体検出では Intersection over Union（IoU）を評価指標として用いることが多い．IoU は，推定した検出対象のバウンディングボックスと正解のバウンディングボックスがどの程度重なっているかを表す指標である．IoU=1.0 では，推定結果と正解データは完全に一致していることを意味し, IoU=0.0 では重なりが無いことを意味する．このような評価指標を用いることで，推定結果が正解と多少ずれていても検出に成功したと判定される．

しかし，本研究ではデータセットの作成を目的としている．作成されたデータセットを教師データとして分類器などの学習をさせる際には，教師データにおけるアノテーションの正確さが分類器の分類精度などの品質に影響を与える．そのため，本研究ではラベル付けにおいて生じうる座標の細かいズレであっても補正の対象とする．

提案手法の処理フローを図 5.2.1 に示す．まず，DeepLabCut により入力動画の姿勢推定を行う．姿勢推定によって得られた座標値を用いて，フレーム間の座標変化率を算出する．この座標変化率を特徴量として用いて，IsolationForest による異常検知を行う．異常検知により誤検出フレームを抽出し，線形補間により補正する．その後，静止時のみ TV によるブレ補正を行う．

3.5 節で述べたように，ブレは本来特徴点の動きがないフレームを動きがあるように検出してしまうことを意味する．そのため，常に特徴点が動き続けている運動時はブレ補正を行わない．

誤検出フレームの抽出方法，誤検出フレームの補正方法，ブレの補正方法についてそれぞれ 5.3 節，5.4 節，5.5 節で述べる．

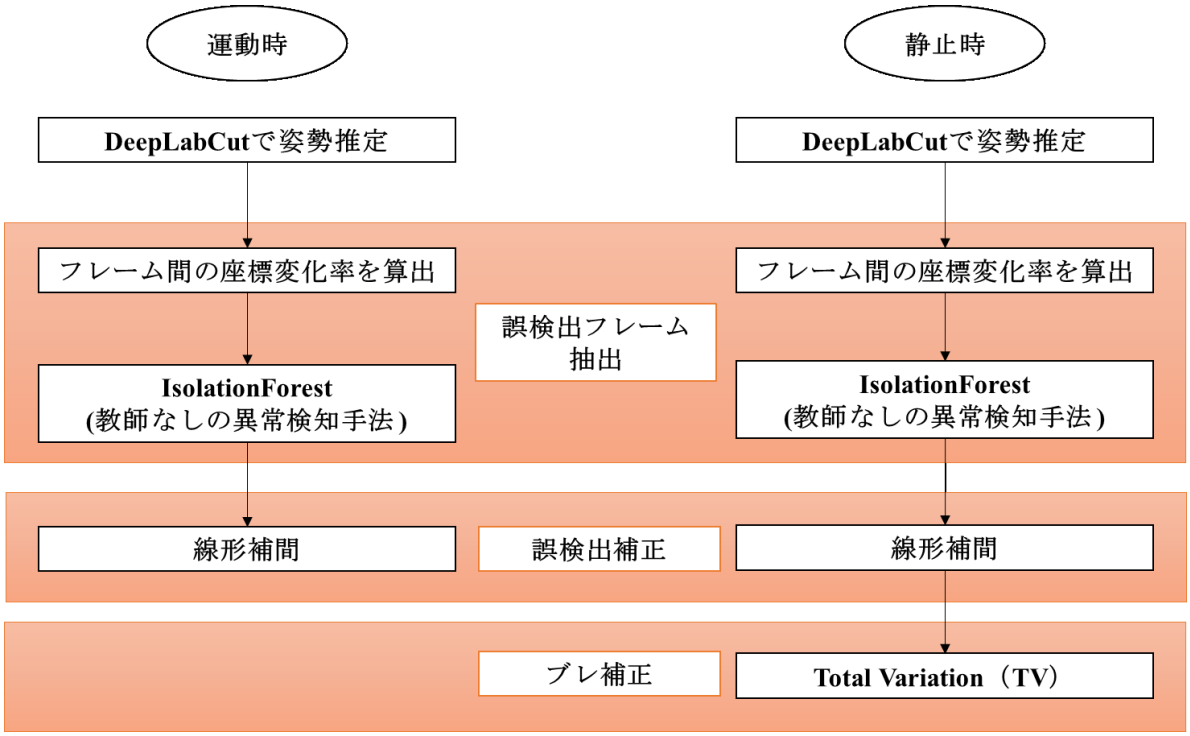


図 5.2.1 提案手法の処理フロー

5.3 誤検出フレームの抽出

2.4 節で述べた IsolationForest を用いて，誤検出フレームの抽出を行う．異常検知から誤検出フレームの抽出までの処理フローを以下に示す．

A. フレーム間の座標変化率を特徴量として，IsolationForest による異常検知を行う．
B. 異常フレームのペアを作成する．
C. フレーム N と M がペアになった時，フレーム N からフレーム M-1 を誤検出フレームとする（N<M）．

誤検出が起きた場合，一つ前のフレームと比較して急激な座標変化が発生する．つまり，誤検出が起きていない場合と比較すると座標変化率が大きくなる．この正常時と比較して大きい座標変化率を「異常」と判定することで，誤検出フレームを抽出する．

2.4 節で述べたように，IsolationForest は異常スコアが 0.5 より大きい値を異常だと判定する．今回，誤検出フレームは正常フレームよりも少ないため，誤検出が起き，急激な座標変化が起きたフレームが異常と判定される．つまり，座標変化率を特徴量として異常検知をした際に，異常スコアが 0.5 より大きいフレームでは座標変化率が大きく，異常スコアが 0.5 未満のフレームでは座標変化率が小さいことを意味する．

誤検出時の座標変化の例を図 5.3.1 に示す．この例では，フレーム 5 からフレーム 7 の全 3 フレームが誤検出である．図 5.3.1 よりフレーム 4 からフレーム 5 と，フレーム 7 からフレーム 8 の座標変化が大きくなっていることがわかる．この時，座標変化率を特徴量として，IsolationForest による異常検知を行うと，表 5.3.1 のような結果になる．

表 5.3.1 より全 10 フレームのうち，フレーム 5 とフレーム 8 が異常検知により「異常」と判定されたことがわかる．この二つのフレームを異常フレームのペアとし，フレーム 5 からフレーム 7 を誤検出フレームとする．

また，表 4.4.7 に示している線形補間可能な区間を，連続の誤検出フレーム数の上限にする．異常と判定されたが，ペアが存在しないフレームは，誤検出ではないとして補正を行わない．

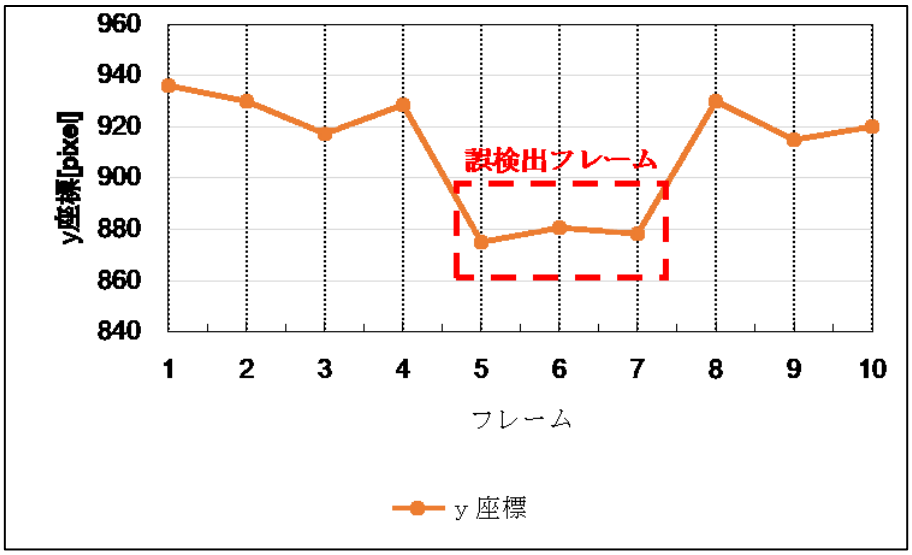


図 5.3.1 誤検出の例

表 5.3.1 IsolationForest による判定結果

フレーム	1	2	3	4	5	6	7	8	9	10
IsolationForest による判定	—	正常	正常	正常	異常	正常	正常	異常	正常	正常

5.4 誤検出フレームの補正

誤検出の補正には，2.5 節で示した線形補間を用いる．誤検出フレームの前後の正常フレームの座標値を線形補間で結ぶことで，補正を行う．

図 5.3.1 の例だと，正常フレームであるフレーム 4 と 8 の座標値を直線で結び，その間の誤検出フレームを補正する．

5.5 ブレの補正

ブレの補正には，2.6 節で示した TV を用いる．平滑化パラメータ λ は 16 を用いる．今回使用した 6 動画では，λ=16 を用いることで，PCK スコアを下げずにブレが補正できることを確認している．

5.6 むすび

本章では，DeepLabCut の課題である「誤検出」と「ブレ」を補正するための提案手法について述べた．

第6章 提案手法による誤検出・ブレの補正実験

6.1 まえがき

本章では，提案手法による誤検出とブレの補正実験について述べる．静止時と運動時のそれぞれのデータセット，実験結果および考察について述べる．

6.2 運動時の誤検出補正実験

提案手法を用いて運動時の誤検出補正を行った．実験に使用したデータセット，実験結果および考察について述べる．

6.2.1 データセット

本実験では，第 3 章の予備実験（1）と同様のデータセットを用いた．運動時のデータセットに関する詳細は 3.3.1 節で述べた．

6.2.2 実験結果

運動 1，運動 2，運動 3 の 3 本の動画を対象として，提案手法による誤検出補正を行った．DeepLabCut による姿勢推定の結果を補正前，DeepLabCut による姿勢推定結果に対して誤検出補正を行った結果を補正後として，補正前後の PCK スコアを比較した．この比較結果を表 6.2.1，表 6.2.2，表 6.2.3 に示す．それぞれの表は 12 ヶ所の特徴点および全特徴点の平均の PCK@0.3 の値を比較している．

表 6.2.2 より，運動 2 の左前脚のつま先では補正前後で PCK スコアが変化していないことがわかる．また，表 6.2.3 より右後脚の膝も補正前後で PCK スコアが変化していない．この二つの特徴点以外では，PCK スコアが上がったことがわかる．全特徴点の平均 PCK スコアを比較すると，運動 1 では 21.1%，運動 2 では 12.3%，運動 3 では 7.0%，PCK スコアが上がったことがわかる．

表 6.2.1 提案手法による運動 1 の PCK スコア

		補正前	補正後
	膝	0.776	1.000
左前脚	かかと	0.871	1.000
	つま先	0.871	1.000
	膝	0.698	1.000
右前脚	かかと	0.845	0.983
	つま先	0.802	0.957
	膝	0.819	1.000
左後脚	かかと	0.853	1.000
	つま先	0.707	0.905
	膝	0.698	1.000
右後脚	かかと	0.974	1.000
	つま先	0.871	1.000
平均		**0.815**	**0.987**

表 6.2.2 提案手法による運動 2 の PCK スコア

		補正前	補正後
	膝	0.884	0.959
左前脚	かかと	0.810	0.893
	つま先	**0.851**	**0.851**
	膝	0.860	0.959
右前脚	かかと	0.843	0.909
	つま先	0.818	0.868
	膝	0.766	0.928
左後脚	かかと	0.694	0.955
	つま先	0.802	0.883
	膝	0.784	0.874
右後脚	かかと	0.793	0.982
	つま先	0.766	0.802
平均		**0.806**	**0.905**

表 6.2.3 提案手法による運動 3 の PCK スコア

		補正前	補正後
	膝	0.912	0.993
左前脚	かかと	0.949	0.971
	つま先	0.882	0.949
	膝	0.934	0.971
右前脚	かかと	0.890	0.919
	つま先	0.875	0.941
	膝	0.897	1.000
左後脚	かかと	0.973	0.993
	つま先	0.836	0.897
	膝	0.986	0.986
右後脚	かかと	0.877	0.993
	つま先	0.747	0.897
平均		**0.896**	**0.959**

6.2.3 考察

本実験では，提案手法を用いて誤検出フレームを補正することを目的としていた．6.2.2 節の実験結果より，提案手法による誤検出の補正前後を比較すると，補正後は全特徴点の平均の PCK スコアが上がったことがわかる．しかし，PCK スコアが 1.000 の値を取ったのは，本実験で用いた全 36 ヶ所の特徴点のうち 9 ヶ所のみだった．つまり多くの特徴点で，全ての誤検出の補正には成功せず，誤検出フレームが残る結果となった．

補正に失敗した誤検出の特徴としては，誤検出が連続していて誤検出フレームのペアの間隔が, 4.4 節で求めた線形補間可能な区間を超えている場合が多かった. したがって，誤検出フレームが多く連続した場合の補正手法も検討する必要がある．

6.3 静止時の誤検出補正・ブレ補正実験

提案手法を用いて静止時のブレと誤検出の補正を行った．実験に使用したデータセット，実験結果および考察について述べる．

6.3.1 データセット

本実験では，第 3 章の予備実験（1）と同様のデータセットを用いた．静止時のデータセットの詳細は 3.4.1 節で述べた．

6.3.2 実験結果

静止 1，静止 2，静止 3 の 3 本の動画を対象として，提案手法による誤検とブレの補正を行った．DeepLabCut による姿勢推定の結果，DeepLabCut による姿勢推定結果に対して誤検出補正を行った結果および誤検出補正後にブレ補正を行った結果の三つの結果を比較した．この比較結果を表 6.3.1，表 6.3.2，表 6.3.3 に示す．それぞれの表は 12 ヶ所の特徴点および全特徴点の平均の PCK@0.3 の値を比較している．

線形補間前後の全特徴点における平均 PCK スコアを比較すると，線形補間後の PCK スコアの上昇率は静止 1 では 1.98%，静止 2 では 3.08%，静止 3 では 2.05%であった．静止時は，運動時と比較すると各特徴点の動きが少ない．そのため、DeepLabCut による推定結果における PCK スコアは，静止時は運動時よりも高い値となった．その結果，提案手法による誤検出補正前後の PCK スコアの上昇率は，運動時よりも低い値となった．

また，DeepLabCut による姿勢推定後をブレ補正前として，ブレ補正前後の TV を比較した．この比較結果を表 6.3.4，表 6.3.5，表 6.3.6 に示す．TV は，式（6.3.1）を用いて算出した．

$$\mathbf{TV} = \sum_{t=1}^{N} \left[\sqrt{\{x(t) - x(t-1)\}^2 + \{y(t) - y(t-1)\}^2} \right] \qquad (6.3.1)$$

ここで，N は総フレーム数を表す．式（6.3.1）のように，特徴点のフレーム間の移動距離を算出する．そして，全フレームにおけるフレーム間の移動距離の総和を TV とする．

表 6.3.1 提案手法による静止 1 の PCK スコア

		DeepLabCut	誤検出補正後	ブレ補正後
	膝	0.996	0.996	0.996
左前脚	かかと	0.987	0.987	0.987
	つま先	0.932	0.974	0.991
	膝	0.987	0.987	0.987
右前脚	かかと	0.991	0.991	0.991
	つま先	0.974	0.979	0.979
	膝	0.932	0.987	0.987
左後脚	かかと	0.974	0.979	0.979
	つま先	0.974	0.983	0.983
	膝	0.940	0.979	0.983
右後脚	かかと	0.919	0.970	0.970
	つま先	0.898	0.923	0.923
平均		**0.959**	**0.978**	**0.979**

表 6.3.2 提案手法による静止 2 の PCK スコア

		DeepLabCut	誤検出補正後	ブレ補正後
	膝	0.900	0.950	0.960
左前脚	かかと	0.910	0.990	0.990
	つま先	0.920	1.000	1.000
	膝	0.990	0.990	0.990
右前脚	かかと	0.960	0.970	0.970
	つま先	0.950	1.000	1.000
	膝	0.960	1.000	1.000
左後脚	かかと	0.930	0.950	0.980
	つま先	0.940	0.950	0.950
	膝	1.000	1.000	1.000
右後脚	かかと	0.930	0.930	0.960
	つま先	0.910	0.920	0.930
平均		**0.942**	**0.971**	**0.978**

表 6.3.3 提案手法による静止 3 の PCK スコア

		DeepLabCut	誤検出補正後	ブレ補正後
	膝	0.920	0.987	0.993
左前脚	かかと	0.953	1.000	1.000
	つま先	0.960	1.000	1.000
	膝	1.000	1.000	1.000
右前脚	かかと	0.973	0.980	0.980
	つま先	0.973	0.980	0.973
	膝	0.987	1.000	1.000
左後脚	かかと	0.960	0.980	0.980
	つま先	0.960	1.000	1.000
	膝	1.000	1.000	1.000
右後脚	かかと	1.000	1.000	1.000
	つま先	1.000	1.000	1.000
平均		**0.974**	**0.994**	**0.994**

表 6.3.4 静止 1 のブレ補正前後の TV 比較

		ブレ補正前の TV[pixel]	ブレ補正後の TV[pixel]
	膝	1594.5	584.4
左前脚	かかと	2363.2	589.7
	つま先	20260.2	613.5
	膝	4917.9	522.8
右前脚	かかと	5480.8	605.8
	つま先	8717.2	437.3
	膝	15747.6	526.5
左後脚	かかと	3373.0	597.6
	つま先	6842.0	481.3
	膝	10199.4	570.6
右後脚	かかと	12549.5	652.0
	つま先	5560.9	449.0

表 6.3.5 静止 2 のブレ補正前後の TV 比較

		ブレ補正前の TV[pixel]	ブレ補正後の TV[pixel]
	膝	10115.3	448.3
左前脚	かかと	7141.7	563.7
	つま先	10630.7	567.5
	膝	811.3	416.1
右前脚	かかと	1053.6	607.6
	つま先	10399.3	614.5
	膝	5413.1	443.0
左後脚	かかと	4575.8	1606.7
	つま先	3452.3	443.2
	膝	855.8	334.4
右後脚	かかと	1106.7	540.0
	つま先	3288.8	503.1

表 6.3.6 静止 3 のブレ補正前後の TV 比較

		ブレ補正前の TV[pixel]	ブレ補正後の TV[pixel]
	膝	8120.4	410.4
左前脚	かかと	1743.4	525.1
	つま先	3454.1	504.0
	膝	1092.3	496.8
右前脚	かかと	3596.9	558.6
	つま先	1218.4	561.2
	膝	4687.8	450.0
左後脚	かかと	3419.8	755.0
	つま先	4817.5	542.6
	膝	1034.3	410.1
右後脚	かかと	1019.3	472.3
	つま先	1086.4	497.0

6.3.3 考察

6.3.2 節の実験結果からブレ補正前後の PCK スコアの変化率と TV の減少率を算出し比較した．3 本の動画に関する比較結果を表 6.3.7,

表 6.3.8，表 6.3.9 に示す．表から，ブレ補正後はブレ補正前と比較して PCK スコアが下がっていないことがわかる．また，TV は 3 動画で平均 78.94%減少した．このように DeepLabCut による姿勢推定の結果は，本来特徴点の動きが無いフレームでも動きがあるように推定されている．しかし，提案手法によってブレ補正を行うことで，より正解に近い特徴点の動きを推定できていることがわかる．

表 6.3.7 静止 1 のブレ補正前後における PCK スコアと TV の比較

		PCK スコアの変化率	TV の減少率
	膝	±0.00%	63.35%
左前脚	かかと	±0.00%	75.05%
	つま先	+6.33%	96.97%
	膝	±0.00%	89.37%
右前脚	かかと	±0.00%	88.95%
	つま先	+0.51%	94.98%
	膝	+5.90%	96.66%
左後脚	かかと	+0.51%	82.28%
	つま先	+0.92%	92.97%
	膝	+4.57%	94.41%
右後脚	かかと	+5.55%	94.80%
	つま先	+2.78%	91.93%
平均		**+2.09%**	**88.48%**

表 6.3.8 静止 2 のブレ補正前後における PCK スコアと TV の比較

		PCK スコアの変化率	TV の減少率
	膝	+6.67%	95.57%
左前脚	かかと	+8.79%	92.11%
	つま先	+8.70%	94.66%
	膝	±0.00%	48.71%
右前脚	かかと	+1.04%	42.33%
	つま先	+5.26%	94.09%
	膝	+4.17%	91.82%
左後脚	かかと	+5.38%	64.89%
	つま先	+1.06%	87.16%
	膝	±0.00%	60.92%
右後脚	かかと	+3.23%	51.20%
	つま先	+2.20%	84.70%
平均		**+3.82%**	**75.68%**

表 6.3.9 静止 3 のブレ補正前後における PCK スコアと TV の比較

		PCK スコアの変化率	TV の減少率
	膝	+7.93%	94.95%
左前脚	かかと	+4.93%	69.88%
	つま先	+4.17%	85.41%
	膝	±0.00%	54.51%
右前脚	かかと	+0.72%	84.47%
	つま先	±0.00%	53.94%
	膝	+1.32%	90.40%
左後脚	かかと	+2.08%	77.92%
	つま先	+4.17%	88.74%
	膝	±0.00%	60.35%
右後脚	かかと	±0.00%	53.66%
	つま先	±0.00%	54.25%
平均		**+2.05%**	**72.37%**

6.4 むすび

本章では，提案手法による誤検出とブレの補正実験について述べた．静止時と運動時のそれぞれのデータセット，実験結果および考察について述べた．その結果，提案手法により DeepLabCut の課題である，「誤検出」と「ブレ」の補正が可能であることが確認できた．

第7章 有効性検証のための実験

7.1 まえがき

本章では，提案手法を用いて作成されたデータセットの有効性を確認するための実験について述べる．実験の概要，実験結果の評価指標，運動時と静止時のそれぞれのクラス分類で用いる特徴量，実験結果および考察について述べる．

7.2 実験の概要

第 6 章では，提案手法により DeepLabCut の課題である「誤検出」と「ブレ」の補正が可能であることが確認できた．しかし，提案手法では全ての誤検出は補正できず，誤検出フレームが残る結果となった．そのため，提案手法によって作成した姿勢情報付きのデータセットの有効性を確認するための実験を行った．

1.1 節で，動物園で飼育動物の行動分類が自動で可能になれば，飼育員の作業負担の軽減に繋がると述べた．そのため，本実験では提案手法によって作成したデータセットを用いて，馬の行動分類を行う．分類手法には，2.7 節で述べた RandomForest を用いた．DeepLabCut による姿勢推定結果，提案手法により作成されたデータ，正解データ（GT）の三つのデータセットに対して行動分類を行い，それぞれの結果を比較する．ここで，GT とは馬のデータセットに対して人手で付与されたラベルを意味する．本実験では，ラベル付きのデータセットを使用したため，このラベルを GT として使用した．

運動時と静止時のそれぞれの実験で使用する特徴量や分類するクラスは，7.4 節と 7.5 節で述べる．

7.3 評価指標

クラス分類の評価指標である，正解率（Accuracy），適合率（Precision），再現率（Recall），F 値（F-measure）について述べる．

2 値分類における予測値と真値の関係を表す混同行列を図 7.3.1 に示す．

真値＼予測値	正	負
正	True Positive (TP)	False Negative (FN)
負	False Positive (FP)	True Negative (TN)

図 7.3.1 2 値分類の混同行列

正解率は全サンプルのうちサンプル正しく予測ができたサンプルの割合を表す．正解率は式（7.3.1）によって表される．

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}} \quad \textbf{(7.3.1)}$$

入力サンプルに正が多いなど偏りがある場合には，正解率以外の指標が使われることが多い．

適合率は，精度と訳されることもあり．正と予測されたサンプル中の実際に正であるサンプルの割合を表す．適合率は式（7.3.2）で表される．

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \quad \textbf{(7.3.2)}$$

再現率は，真値が正であるサンプルを正と判定できた割合を表す．再現率は式（7.3.3）で表される．

$$\boldsymbol{Recall = \frac{TP}{TP + FN}} \quad \textbf{(7.3.3)}$$

適合率と再現率はトレードオフの関係にあり，適合率が高いシステムは，再現率が低く，再現率が高いシステムは適合率が低い．そのため適合率と再現率の調和平均である，F 値で分類精度を評価することもある．F 値は式（7.3.4）で表される．

$$\boldsymbol{F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}} \quad \textbf{(7.3.4)}$$

7.4 運動時における提案手法の有効性

提案手法によって作成した運動時の姿勢情報付きデータセットの有効性について述べる．

7.4.1 使用する特徴量，分類クラス

RandomForest を用いてクラス分類をする際に使用する特徴量と，分類するクラスを表 7.4.1 に示す．実験に使用するデータセットから二つの特徴量を算出し，RandomForest によって 2 クラス分類を行う．

運動 1，運動 2，運動 3 のデータセットは，3.3.1 節で述べたように歩行中のデータである．四足動物の歩行や走行は，4 本足が地面につくタイミングの違いで分類される．そのため，各脚が地面に付いているか，付いていないかの 2 クラス分類を行う．

また，3 本の動画では，右側の前脚と後脚は同じ軌道を通る．同様に左側の前脚と後脚も同じ軌道を通る．そのため，データを右側の脚と左側の脚の二つに分けて，それぞれのデータに対してクラス分類の実験を行った．

表 7.4.1 運動時の分類するクラス，使用する特徴量

分類するクラス	使用する特徴量
脚が地面に付いている	かかとの関節角度
脚が地面に付いていない	つま先の y 座標

7.4.2 実験結果

3 本の動画を対象に，それぞれ GT，DeepLabCut による姿勢推定結果および提案手法によって作成されたデータの三つのデータを使用し，RandomForest によるクラス分類を行った．

クラス分類の結果を表 7.4.2，表 7.4.3，表 7.4.4 に示す．また，左右のデータに対するクラス分類精度の平均値をまとめた結果を表 7.4.5，表 7.4.6，表 7.4.7 に示す．

表 7.4.2 運動 1 のクラス分類精度（左側，右側）

	左側				右側			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.880	0.893	0.862	0.877	0.860	0.905	0.905	0.905
DeepLabCut	0.660	0.655	0.655	0.655	0.590	0.600	0.429	0.500
提案手法	0.970	0.800	0.800	0.800	0.900	0.944	0.773	0.850

表 7.4.3 運動 2 のクラス分類精度（左側，右側）

	左側				右側			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.880	0.897	0.921	0.909	0.980	1.000	0.967	0.983
DeepLabCut	0.830	0.868	0.868	0.868	0.910	0.969	0.886	0.925
提案手法	0.860	0.941	0.842	0.889	0.950	0.971	0.943	0.957

表 7.4.4 運動 3 のクラス分類精度（左側，右側）

	左側				右側			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.920	0.929	0.929	0.929	0.940	0.962	0.962	0.962
DeepLabCut	0.850	0.878	0.857	0.867	0.850	0.904	0.887	0.895
提案手法	0.910	0.933	0.903	0.918	0.930	0.951	0.951	0.951

表 7.4.5 運動 1 のクラス分類精度（平均）

	正解率	適合率	再現率	F 値
GT	0.870	0.899	0.883	**0.891**
DeepLabCut	0.625	0.628	0.542	**0.578**
提案手法	0.935	0.872	0.786	**0.825**

表 7.4.6 運動 2 のクラス分類精度（平均）

	正解率	適合率	再現率	F 値
GT	0.930	0.949	0.944	**0.946**
DeepLabCut	0.870	0.919	0.877	**0.897**
提案手法	0.905	0.956	0.893	**0.923**

表 7.4.7 運動 3 のクラス分類精度（平均）

	正解率	適合率	再現率	F 値
GT	0.930	0.945	0.945	**0.945**
DeepLabCut	0.850	0.891	0.872	**0.881**
提案手法	0.920	0.942	0.927	**0.935**

7.4.3 考察

7.4.2 節では，GT，DeepLabCut による推定結果，提案手法によって作成したデータの三つのデータを使用しクラス分類を行った．表 7.4.5，表 7.4.6，表 7.4.7 の太字で記された値が示すように，それぞれのクラス分類結果を F 値で比較すると，3 動画全てで GT を使用した場合が最も良い精度となった．

GT の作成時は全フレームを人手でラベル付けしている．一方，DeepLabCut と提案手法では，動画の全フレームのうち一部を人手でラベル付けし，DeepLabCut のネットワークに学習させることで，残りのフレームを自動でラベル付けしている．このようにラベル付けの負担を減らすことができる点が，DeepLabCut を使用する利点である．そのため，ク

ラス分類の精度だけではなく，データセット作成時の負担を評価指標として加え，三つのデータを評価する．

ここで，全フレームのうち，人手でラベル付けをしているフレームの割合をラベル付けの負担と設定する．つまり，GT のラベル付け負担は 1.0 となる．運動 1，運動 2，運動 3 のラベル付け負担を表 7.4.8 に示す．

3 本の動画に対して，クラス分類の際の F 値とラベル付け負担を比較し評価した図をそれぞれ図 7.4.1，図 7.4.2，図 7.4.3 に示す．それぞれの図から，提案手法によるデータセットは，DeepLabCut よりも F 値が高く，GT よりもラベル付け負担を削減できることが分かる．

表 7.4.8 運動時データのラベル付け負担

動画	総フレーム数	ラベル付けフレーム数	ラベル付け負担
運動 1	235	40	0.170
運動 2	256	40	0.156
運動 3	288	40	0.139

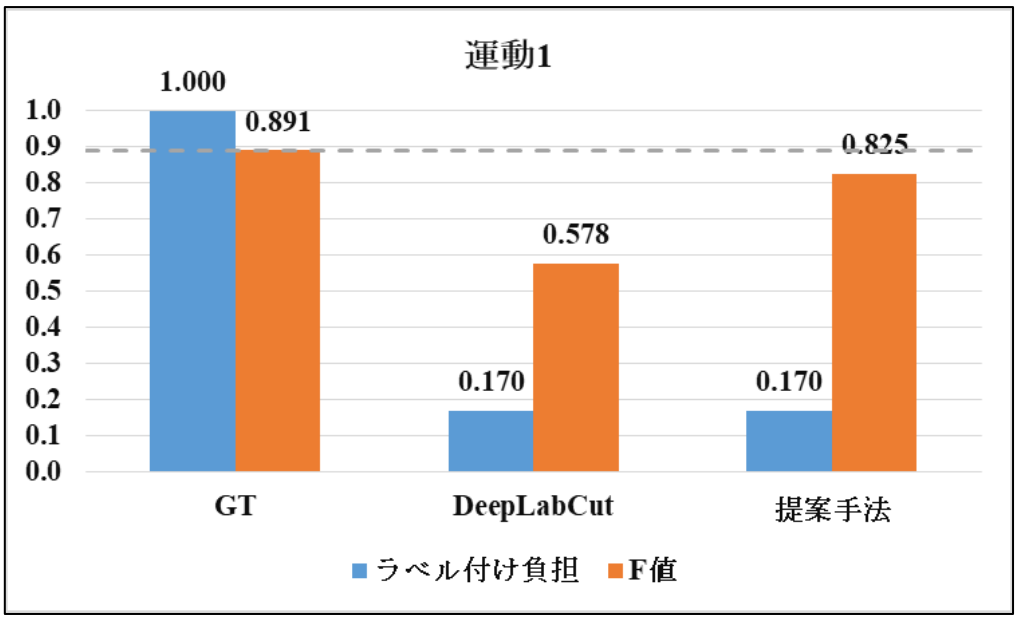


図 7.4.1 運動 1 のデータセットの有効性

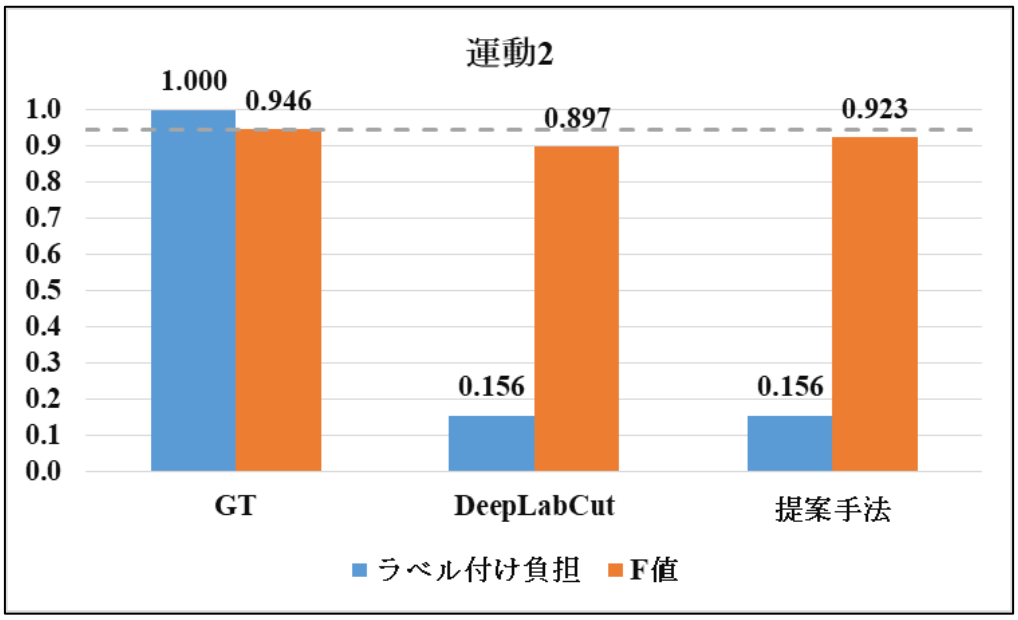


図 7.4.2 運動 2 のデータセットの有効性

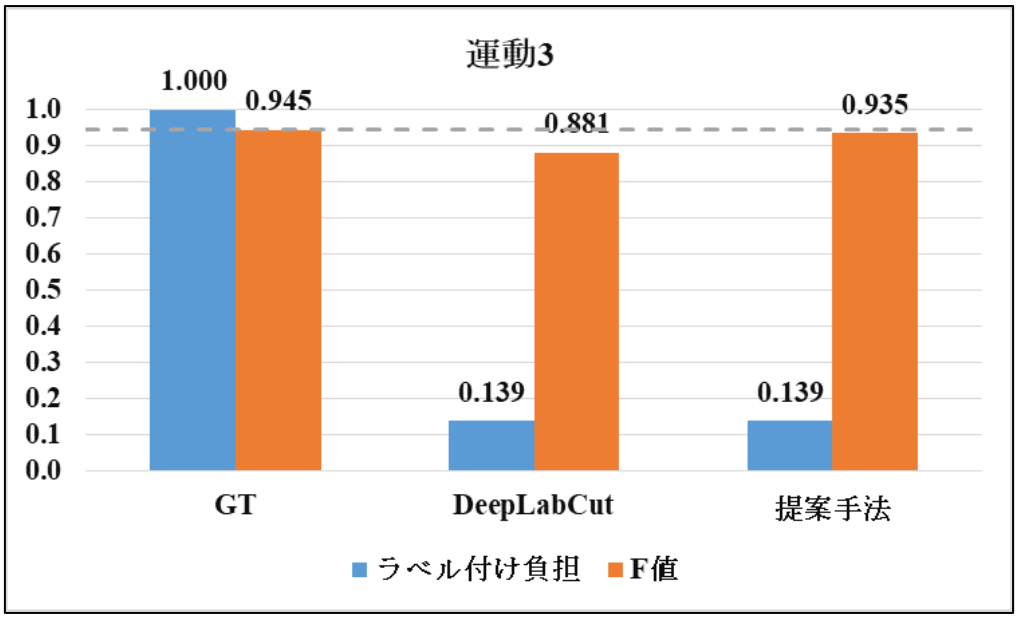


図 7.4.3 運動 3 のデータセットの有効性

7.5 静止時における提案手法の有効性

提案手法によって作成した静止時の姿勢情報付きデータセットの有効性について述べる.

7.5.1 使用する特徴量，分類クラス

RandomForest を用いてクラス分類をする際に使用する特徴量と分類するクラスを表 7.5.1 に示す．表 7.5.1 が示すように，使用するデータセットから二つの特徴量を算出し，RandomForest によって 2 クラス分類を行う．

静止 1, 静止 2, 静止 3 のデータセットは, 3.4.1 節で述べたように静止が中心であるが，各脚が「一歩前に踏み出す」のような動きを含む．そのため脚が動いているか，動いていないかの 2 クラス分類を行う．

4 本の脚それぞれの座標データから使用する特徴量を算出し, RandomForest による 2 クラス分類を行う．

表 7.5.1 静止時の分類するクラス，使用する特徴量

分類するクラス	使用する特徴量
脚が動いている	かかとの関節角度
脚が動いていない	かかとの y 座標

7.5.2 実験結果

3 本の動画を対象に，それぞれ GT，DeepLabCut による姿勢推定結果および提案手法によって作成されたデータの三つのデータを使用し，RandomForest によるクラス分類を行った．その結果を表 7.5.2，表 7.5.3，表 7.5.4 に示す．また，4 本の脚のデータそれぞれに対する，クラス分類精度の平均値をまとめた結果を表 7.5.5，表 7.5.6，表 7.5.7 に示す．

表 7.5.2 静止 1 のクラス分類精度

	左前脚				右前脚			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.980	1.000	0.857	0.923	0.980	1.000	0.800	0.889
DeepLabCut	0.930	1.000	0.571	0.727	0.900	0.600	0.600	0.600
提案手法	0.951	1.000	0.714	0.833	0.950	1.000	0.600	0.750
	左後脚				右後脚			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.980	1.000	0.750	0.857	0.950	0.900	0.900	0.900
DeepLabCut	0.900	0.500	0.500	0.500	0.880	0.857	0.600	0.706
提案手法	0.950	1.000	0.500	0.667	0.950	1.000	0.800	0.889

表 7.5.3 静止 2 のクラス分類精度

	左前脚				右前脚			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	1.000	1.000	1.000	1.000	0.970	1.000	0.857	0.923
DeepLabCut	0.900	1.000	0.500	0.667	0.867	0.800	0.571	0.667
提案手法	1.000	1.000	1.000	1.000	0.967	1.000	0.857	0.923
	左後脚				右後脚			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.967	1.000	0.857	0.923	1.000	1.000	1.000	1.000
DeepLabCut	0.833	1.000	0.286	0.444	0.933	1.000	0.667	0.800
提案手法	0.967	1.000	0.857	0.923	0.967	1.000	0.833	0.909

表 7.5.4 静止 3 のクラス分類精度

	左前脚				右前脚			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.980	1.000	0.800	0.889	1.000	1.000	1.000	1.000
DeepLabCut	0.844	0.250	0.200	0.222	0.933	0.857	0.750	0.800
提案手法	0.960	0.800	0.800	0.800	1.000	1.000	1.000	1.000
	左後脚				右後脚			
	正解率	適合率	再現率	F 値	正解率	適合率	再現率	F 値
GT	0.978	0.909	1.000	0.952	0.910	0.571	0.800	0.667
DeepLabCut	0.956	0.900	0.900	0.900	0.933	0.750	0.600	0.667
提案手法	0.978	1.000	0.900	0.947	0.933	0.750	0.600	0.667

表 7.5.5 静止 1 のクラス分類精度（平均）

	正解率	適合率	再現率	F 値
GT	0.973	**0.975**	0.827	**0.892**
DeepLabCut	0.903	0.739	0.568	**0.633**
提案手法	0.950	**1.000**	0.654	**0.785**

表 7.5.6 静止 2 のクラス分類精度（平均）

	正解率	適合率	再現率	F 値
GT	0.984	**1.000**	0.929	**0.962**
DeepLabCut	0.883	0.950	0.506	**0.644**
提案手法	0.975	**1.000**	0.887	**0.939**

表 7.5.7 静止 3 のクラス分類精度（平均）

	正解率	適合率	再現率	F 値
GT	0.967	**0.870**	0.900	**0.877**
DeepLabCut	0.917	0.689	0.613	**0.647**
提案手法	0.968	**0.888**	0.825	**0.854**

7.5.3 考察

運動時と同じくクラス分類の精度とラベル付け負担を評価指標として, GT, DeepLabCut, 提案手法によるデータを比較した．静止 1，静止 2，静止 3 に対して，クラス分類の際の F 値とラベル付け負担を比較し，評価した図をそれぞれ図 7.4.1，図 7.4.2，図 7.4.3 に示す.

運動時の比較結果と同様に，提案手法によるデータセットは，DeepLabCut よりも F 値が高く，GT よりもラベル付け負担を削減できることが分かる.

また，表 7.5.5，表 7.5.6，表 7.5.7 の赤字で記された値が示すように，適合率を比較すると，GT と提案手法は等しい，または提案手法のほうが高い値だとわかる．本実験結果における適合率は，「脚が動いている」と推定されたフレームのうち，実際に動いていたフレームの割合を指す. 適合率が高いということは, 動いていないフレームを動いていると誤分類することは少ないが，動いているフレームを動いていないと誤分類している割合が高いということである.

提案手法では，全フレームに対して TV によるブレ補正を行った．ブレ補正を行ったことで, 動きがあるフレームとないフレーム間での座標変化が滑らかになり, 動きの有無でクラスを分けた際に，その境界が曖昧になった可能性がある．そして，動きがあるにも関わらず，動きがないと判定されるフレームの割合が増えたと考えられる.

したがって, 全フレームに対してブレ補正を行うのではなく, 動きがあるフレームではブレ補正を行わないなどの手法を検討する必要がある.

表 7.5.8 静止時データのラベル付け負担

動画	総フレーム数	ラベル付けフレーム数	ラベル付け負担
静止 1	236	20	0.085
静止 2	1601	20	0.012
静止 3	538	20	0.037

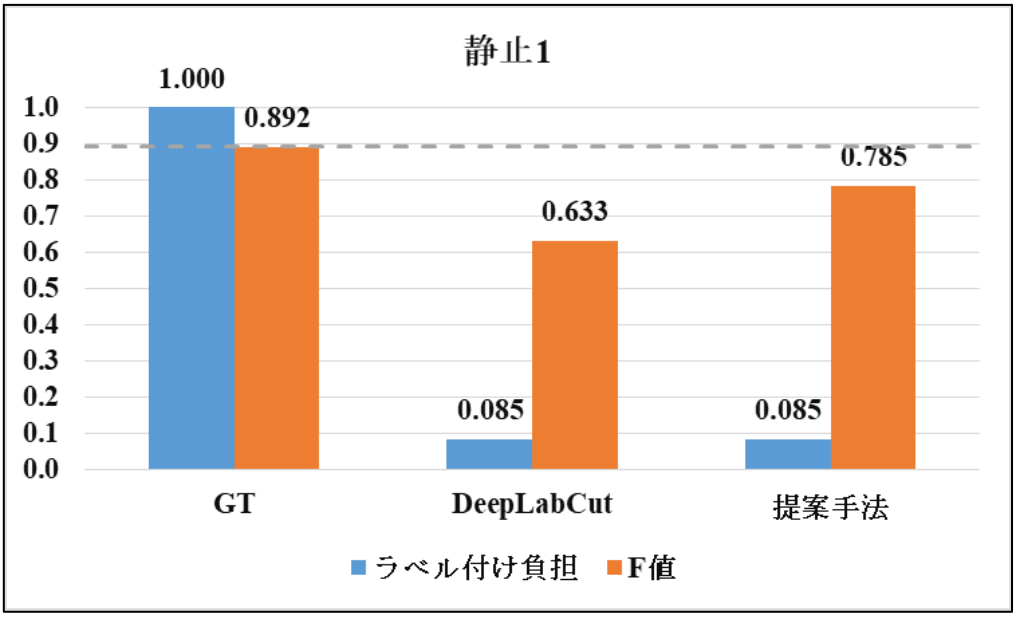


図 7.5.1 静止 1 のデータセットの有効性

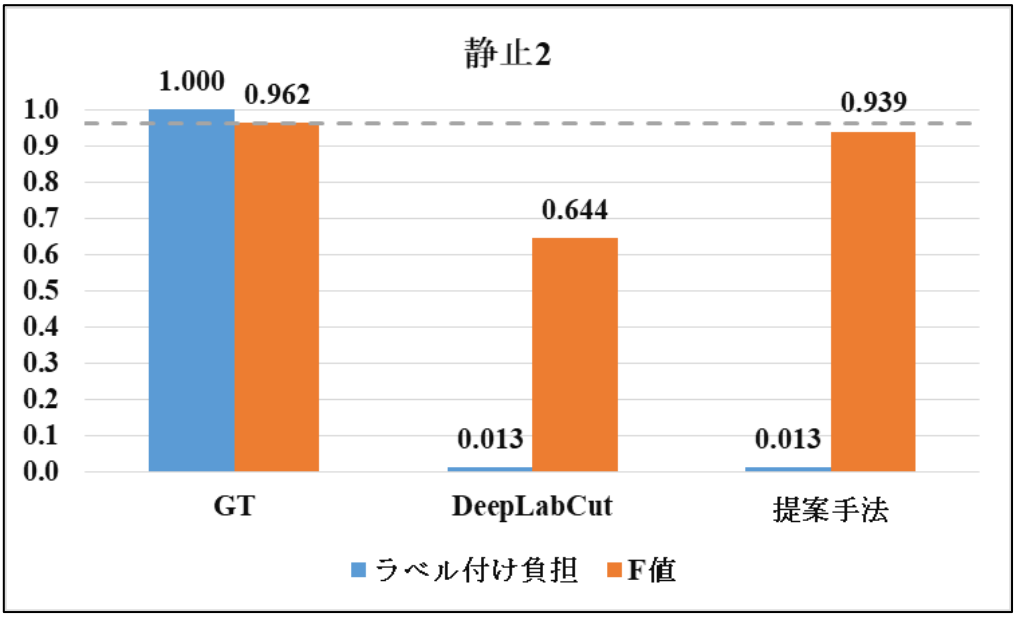


図 7.5.2 静止 2 のデータセットの有効性

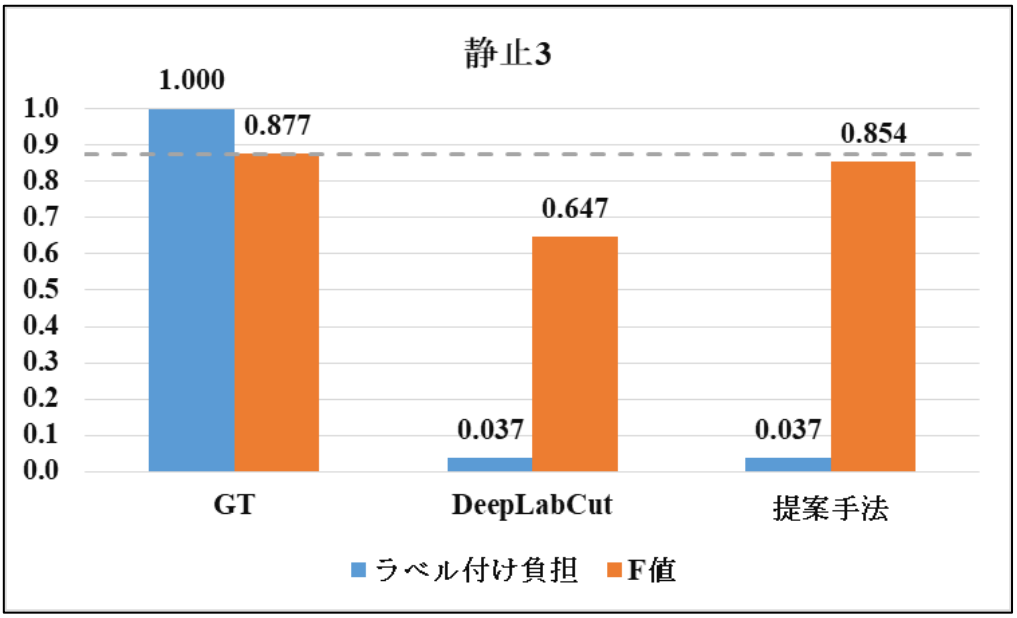


図 7.5.3 静止 3 のデータセットの有効性

7.6 むすび

本章では，提案手法を用いて作成されたデータセットの有効性を確認するための実験について述べた．その結果，提案手法により作成されたデータを用いて馬の行動分類を行うと，DeepLabCut の推定結果を用いた場合よりも，クラス分類の精度が上がることが確認できた．また，GT を使用してクラス分類をする場合と比較すると，分類精度は劣っているが，データセット作成時の負担は削減できることが確認できた．

第8章 結論

8.1 結論

本研究では,動物の姿勢推定技術である DeepLabCut を使用する際の課題である「誤検出」と「ブレ」を補正することで，動物画像のデータセット作成の負担を軽減することを目的とした．動物の個体識別や姿勢推定の成功事例は一部の動物に限られているが，データセット作成時の負担が軽減されデータセットが増えることにより，より多くの動物で姿勢推定が可能になることが考えられる．

提案手法では，誤検出に対して線形補間を用い，ブレに対して TV を小さくすることで補正した．DeepLabCut の姿勢推定結果の誤検出に対して線形補間を行うと，補正前後で姿勢推定の結果が正しいかを示す評価指数である PCK スコアが上昇した．その上昇率は，運動時のデータを使用した場合は平均 13.5%, 静止時のデータを使用した場合は平均 2.4%だった．つまり，誤検出に対して線形補間を行うことで誤検出フレームを補正できることが確認できた．

また，静止時のデータに対してブレ補正を行うと，特徴点の全フレームにおける総変化量を示す評価指数である TV は 3 動画で平均 78.8%削減することができ，PCK スコアを下げずに TV を削減できることがわかった．このことから，提案手法を用いることで，DeepLabCut の姿勢推定結果よりも GT に近い特徴点の動きを捉えられることが確認できた．

さらに，提案手法により作成されたデータの有効性を検証する実験では，GT と提案手法により作成されたデータを使用し，それぞれのデータで馬の行動に関するクラス分類を実施した．クラス分類精度を F 値で比較すると，GT を使用した場合のほうが良い精度となることが示された．その一方で，データセット作成時のラベル付け負担を比較すると，提案手法では GT より負担を軽減でき，その削減率は，運動時は平均 84.5%，静止時は平均 95.5%だった．よって提案手法を用いることでデータセット作成時の負担を軽減できると言える．

8.2 今後の課題

本研究では，提案手法により作成されたデータを使用したクラス分類の精度は，GT を使用した場合よりも低い結果となった．これは，提案手法により作成されたデータ中に誤検出フレームが残っていることが原因の一つだと考えられる．また誤検出の補正に失敗した際の特徴として，誤検出が連続し，誤検出フレームのペアの間隔が線形補間可能な区間を超えている場合が多かった．そのため，提案手法により作成れたデータの有効性を更に高めるためには，誤検出が連続した場合の補正手法を検討する必要がある．

より多くの誤検出を補正する手法を検討することで，提案手法により作成されたデータセットを使用した際のクラス分類精度が向上すると考えられる．そして，クラス分類の精度が向上し，動物園などの動物の行動分類に活用することが可能になれば，飼育員の負担軽減に繋がると考えられる．

謝辞

本研究の方向性等の丁寧かつ熱心なご指導を頂いた渡辺裕教授と早稲田大学国際情報通信センターの石川孝明様に心から深謝いたします．また，日頃から御意見やアドバイスをくださった研究室の皆様に御礼申し上げます．

最後に，これまで育てていただき，学業に専念させてくださった家族に心より感謝致します．

参考文献

[1] D. Schofield, A. Nagrani, “ Chimpanzee face recognition from videos in the wild using deep learning, “ Science Advances Vol. 5, eaaw0736, 2019.

[2] A. Mathis, P. Mamidanna1, ” Markerless tracking of user-defined features with deep learning, ” arXiv preprint arXiv: 1804. 03142, 2018.

[3] AlexEMG/DeepLabCut, https://github.com/AlexEMG/DeepLabCut, GitHub, Inc. (2022 年 1 月 11 日アクセス)

[4] E. Insafutdinov, L. Pishchulin, “DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model, ” arXiv preprint arXiv: 1605.03170, 2016.

[5] L. Pishchulin, E. Insafutdinov, “DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation, ” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), December. 2016.

[6] K. He, X. Zhang, “Deep Residual Learning for Image Recognition, ”The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June. 2016.

[7] Z. Cao, G. Hidalgo, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, ” arXiv preprint arXiv: 1812. 08008, 2018.

[8] S. Pan, Q. Yang, “A Survey on Transfer Learning, ” The IEEE Transactions on Knowledge and Data Engineering (TKDE), October. 2010.

[9] A. Sanakoyeu, V. Khalidov, “Transferring Dense Pose to Proximal Animal Classes, ” arXiv preprint arXiv: 2003. 00080, 2020.

[10] F. Liu, K. Ting, Z. Zhou, “Isolation forest, “ The IEEE International Conference on Data Mining (ICDM), December. 2008.

[11] T. Siauw, A. Bayen, An Introduction to MATLAB® Programming and Numerical Methods for Engineers, Academic Press, Boston, 2015, pp. 211-233.

[12] B. Leo, “Random Forests, “ Machine Learning, October. 2001.

図一覧

表一覧

研究業績

[1] S. Fujimori, T. Ishikawa, H. Watanabe, “Animal Behavior Classification Using DeepLabCut, “ 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), October, 2020.