H-036

シングルボードコンピュータで動作できる手の動作予測モデルの作成と検討

岩田雄介†　　田村仁†

日本工業大学院工学研究科　機械システム工学専攻†

1. はじめに

近年，急速に進化するロボティクス技術は，人間の動きに対応するロボットの応用範囲を拡大している．特に，自動運転技術や球技を行うロボットなどでは，人間の動作を予測し，それに適切に反応することが求められている．ロボットシステムを実現するためには，単なるカメラ映像の解析だけでは不十分で，実時間の映像を待つだけでなく，予測に基づいて動作を決定する必要がある．ロボットの駆動には時間がかかるため，目標となる物体の動きを予測し，実時間よりも前に動作を決定することで，駆動に必要な時間を確保することが重要である．

例えば，じゃんけんの場合，相手が手を出す前にその手を予測し，その予測に基づいて動作するようにロボットを駆動させる必要がある．

本研究では，深層学習を用いて相手の動作を予測し，実時間でロボットの動作を間に合わせることを目指している．特に，ロボットに搭載することを想定し，シングルボードコンピュータ上で深層学習を実施することに焦点を当てる．シングルボードコンピュータを用いて，ロボットの駆動時間を確認し，実際の運用に適したシステムを構築することが本研究の目的である．

2. 関連研究

関連研究として，東京工業大学の動作予測システム[1]が挙げられる．この研究では，格闘訓練において 0.5 秒後の動きをリアルタイムで予測することが可能とされている．このシステムは VR 技術を活用して動作を再現しするため，高性能デスクトップ PC を必要とする．楽天のロボット[2]は高速度カメラを使用した高度な動作予測を行っている．このように．　これまでの研究では主に高速度カメラやデスクトップ PC を使用することが一般的であり，小型で普通のカメラを利用した動作予測についての報告は見当たらない．

3. 提案手法

手の予測には図 1 のような時間の流れになっている．

カメラ遅延が a 秒，実際の映像を関節座標に変換する時間が b 秒，予測処理時間が b 秒，そしてロボットの動作時間が c 秒かかる．したがって，手の形を予測するためには，これらの時間的な要素を総合的に考慮した時間予測が必要だ．この合計時間は a+b+c+d 秒先の手の形を予測することに対応している．

そのため，本研究では,カメラ遅延と予測処理時間を測定し，ロボットの動作可能時間を求める．

本研究では予測結果が分かりやすいじゃんけんにしぼって実験する．じゃんけんの手が出る時間は自分で確認したところ 300 ミリ秒であることが分かった．

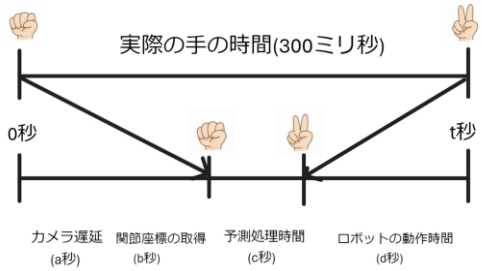

図 1　手の予測をするときの時間の流れ

実機には小型 PC かつ機械学習に特化した Jetson orin nano を使用し，Mediapipe[3]を介して関節座標を取得する．

対象となるデータは，じゃんけんの手を出す動画をフレームごとに分割し，各画像において Mediapipe を用いて関節座標を抽出する．これらの関節座標データを CSV ファイルにまとめ,学習データとして用いる．

学習には LSTM（Long Short Term Memory）[4]と Transformer[5]を採用する．LSTM は RNN

Development and Evaluation of a Hand Movement Prediction Model on a Single-Board Computer
†Yusuke Iwata
†Hitoshi Tamura
‡Nippon Institute of Technology, Graduate School, Mechanical System Engineering Major

（Recurrent Neural Network）の一種であり，時系列データに対して長期的な依存関係を考慮することが可能である．入力ゲート，忘却ゲート，出力ゲートの 3 つのゲート機構により，情報の流れを制御し，重要な情報を保持しつつ不要な情報を削除する．これにより，長期間の依存関係を効果的に学習できる．

一方，Transformer は全てのタイムステップを同時に処理できる点が特徴である．自己注意機構により，各トークンの重要度を計算し，重要な情報をより深く処理する．並列処理が容易であり，大規模なデータセットのトレーニングも効率的に行える．

4 評価実験

4.1 カメラの遅延の確認(a 秒)

先行研究[6]では，カメラの遅延時間を確認するための実験が行われいる．この実験では，一般的に使用される USB カメラ，具体的にはソニーのデジタルカメラ（960fps 60p 50m）を使用し，Jetson Orin Nano に接続されたモニターに表示される手の映像と現実の手を同じ画面で撮影している．その後，この撮影された動画を 1 フレーム毎に分割し，カメラの遅延時間を計測した．実験結果から，カメラの遅延は 2.3 ミリ秒であることが確認できる．(図 2)

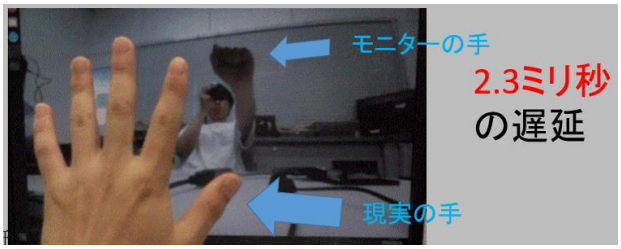

図 2　カメラ遅延の確認

4.2 関節座標に変換する時間の確認(b 秒)

先行研究[6]では，カメラ映像を関節座標に変換する際の時間を確認するための実験が行われている．この実験では，カメラから取得した映像をリアルタイムで MediaPipe に変換するプログラムを用いて，その処理時間を測定した．MediaPipe を使用して手の関節座標に変換する時間を計測した結果，処理時間は 100 ミリ秒であることが確認されている．

4.3 手の予測の時間測定(c 秒)

次に，予測処理時間 b 秒の計測を行う．

手の予測の時間測定では以下の条件で行った．

① 3 フレーム学習させ予測

テストデータは，予測したいフレームから 3 フレームを入力データとし，LSTM へ入れ予測結果を出力させる．(図 3)

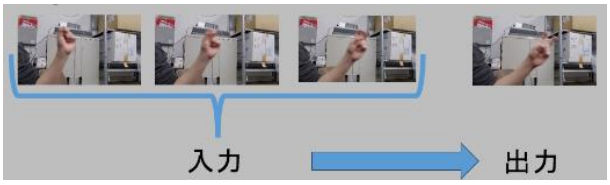

図 3　学習フレーム数

② 学習データの適正化

学習に使うデータは手の形の評価をする為，CSV ファイルの手首の関節座標（landmark_0）を(0.5,0.5)に変更して学習させる．(図 4)

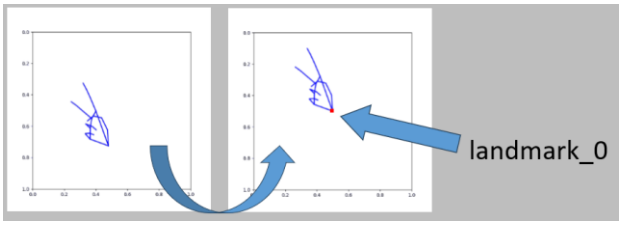

図 4　データの適正化

③ 実験回数

本実験では 1,3,5 フレーム先の関節座標を各 10 回出力する．

④ テストデータ

じゃんけんの手の動きをディープラーニングで学習し，未来の手の動きを予測するためのプログラム作成する．

学習データとして，自分で撮影したじゃんけんの「最初はグーじゃんけんポン」の動画からフレームごとに抽出された関節座標データを使用する．学習データはグー・チョキ・パーを各 100 回，テストデータは 1 回撮影したものを使用する．画像数は学習用が 2712 枚，テストデータは 14 枚になる．

4.3.1 LSTM

まず，LSTM を使用する．CSV ファイルに保存された関節座標データを読み込む．

学習条件の値は以下のように設定した．

- 入力層：567 ノード
- 隠れ層:16 ノード（1 層）
- 出力層:63 ノード
- エポック:100
- バッチサイズ:36

結果は平均で 1.0 ミリ秒予測に時間が掛かっ

ている．（図 5）

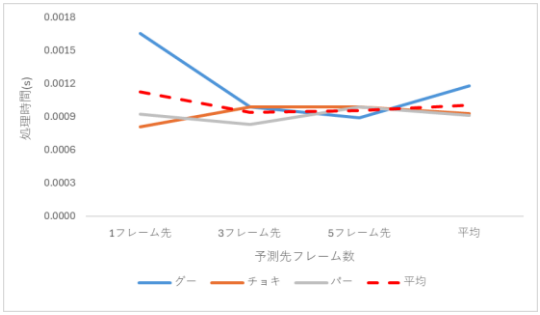

図 5　LSTM の処理時間

4.3.2Transformer

LSTM で行った実験を Transformer で行う．

学習では LSTM と同様に Transformer を使用し，CSV ファイルに保存された関節座標データを読み込む．

学習条件の値は以下のように設定した．

・隠れ層：63
・Transformer の層:4
・埋め込み層の次元：64
・エポック:100
・バッチサイズ:36
・ヘッドアテンション:3

結果は平均で 52 ミリ秒予測に時間が掛かっている．（図 6）

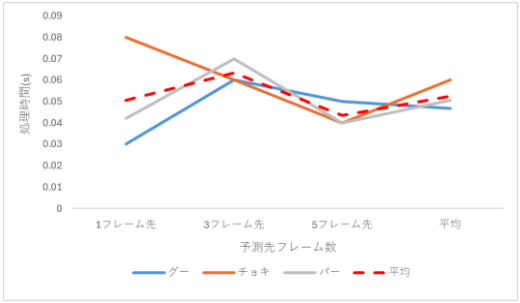

図 6　Transformer の処理時間

4.4 精度評価

精度の比較を行いたいので予測値と実測値の平均二乗誤差で評価する．（式 1）

$$\frac{(実測値 - 予測値)^2}{データ数} \quad (1)$$

4.4.1LSTM

LSTM での予測精度評価を行った結果が図 7 である．予測するフレームをより前にした時ほど誤差が大きくなっていることが分かる．

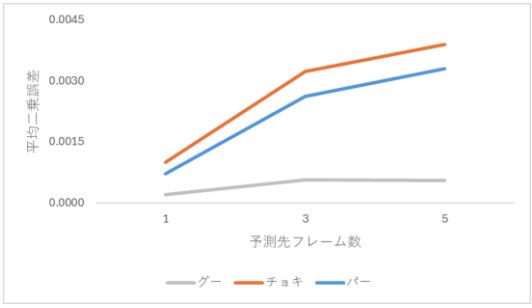

図 7　LSTM の平均二乗誤差

予測結果は画像として保存，表示される．画像の赤い線が予測した関節座標，青の線が予測したい正しい数フレーム先の関節座標である．（図 8）予測しているが，チョキ・パーの指先が伸びていないことが分かる．

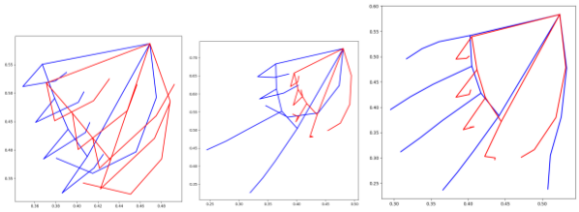

図 8　LSTM の出力結果

4.4.2 Transformer での精度評価

Transformer での予測精度評価を行った結果が図 9 である．誤差が予測先フレーム数に依存していないことが分かる．

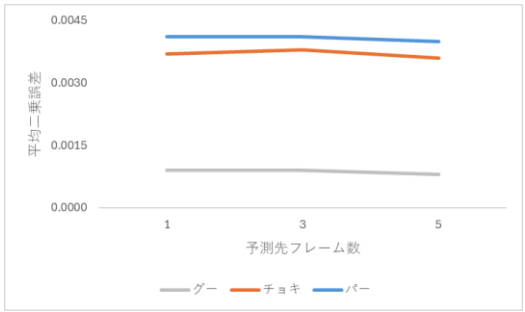

図 9　Transformer の平均二乗誤差

LSTM と同様，予測結果は画像として保存，表示される．画像の赤い線が予測した関節座標，青の線が予測したい正しい数フレーム先の関節座標である．（図 10）予測はしているがグー・チョキ・パーそれぞれで同じ出力結果になっている．

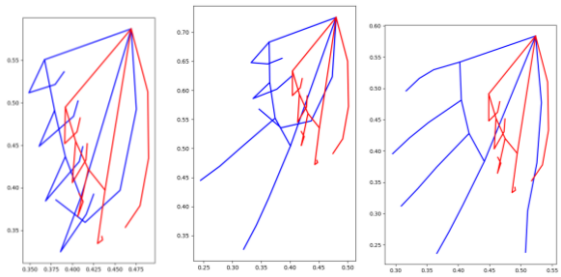

図 10　Transformer の出力結果

5 考察

先行研究[6]から a が 2.3 ミリ秒,b が 100 ミリ秒と分かった.じゃんけんの手が出る時間は 300 ミリ秒なので,予測処理とロボットの動作にかけて良い時間は 197.7 ミリ秒となる．手の予測は 30fps で考えた時,約 10 フレーム先の手を出力することが求められる.

LSTM の予測時間は 1.0 ミリ秒と処理が速いことが分かる．それに比べて，Transformer は 52 ミリ秒と遅いことが分かる.

実験結果から LSTM を使用するとロボットの動作時間は 196.7 ミリ秒確保できる．Transformer を使用するとロボットの動作時間は 145.26 ミリ秒確保できる．ロボットの動作時間は先行研究[6]から 150～300 ミリ秒で動作することが分かっているので，処理速度の面では LSTM はじゃんけん予測が可能だが Transformer ではできない.

原因として，GPU を使用していないため Transformer 本来の処理速度を出せてないことや,トークン，位置エンコーダなどの Transformer の構造を効率よく使用できていないことが挙げられる．さらに，ヘッドアテンションやバッチサイズなどの学習条件を適切に設定できていない可能性が考えられる.

テストデータの数が少ない為，軽い処理ならば LSTM などの単純処理の方が速い可能性も考えられる.

精度に関しては，LSTM の予測ではグー・チョキ・パーそれぞれに対応して予測していることが図 8 から分かる．しかし，図 10 から，Transformer ではグー・チョキ・パーそれぞれに同じような結果が出力されている．平均二乗誤差でも予測先フレーム数を変更すると LSTM では,誤差が大きくなるの対し，Transformer では大きな変化が見られない

原因として，学習データ，テストデータ共に時系列処理されておらず，目的の処理がされていない可能性がある.

6 おわりに

LSTM と Transformer の比較実験を行ったが LSTM の方が速く処理が終わり，フレームごとの予測ができていることが分かった.

Transformer 本来の性能を発揮させることが出来ていない可能性がある．対策として，GPU の使用や学習条件の見直し，データセットの変更などを今後行っていきたい.

Transformer には時系列データに強いモデルがあるので今後の実験で検証していきたい．さらに，モデルへのデータの渡し方を指一本や一点の関節座標（x,y,z）に変更してデータセットの取り扱いにも注視していきたい.

参考文献

[1] Hideki Koike, FuturePose - Mixed Reality Martial Arts Training Using Real-Time 3D Human Pose Forecasting With a RGB Camera, 07-11 January 2019

[2] 山川雄司 石川正俊 勝率 100%じゃんけんロボットの開発 画像ラボ Vol. 24, No. 6, , (pp. 1-8

[3] Mediapipe, “GitHub -googlesamples/mediapipe”, (最終閲覧日 2024/1/12)

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. http://doi.org/10.1162/neco.1997.9.8.1735

[5] Ashish Vaswani et al., ″Attention is all you need″, Advances in neural information processing systems, vol. 30, 2017.

[6] 岩田 雄介，田村 仁,深層学習による人の動作予測，情報処理学会第 86 回全国大会講演論文集 2 分冊,pp.289-290(2024)