C-017

姿勢推定モデル「MoveNet」における回路動作の最適化

Optimization of circuit operation in the posture estimation model "MoveNet"

田中 康雅‡　　中西 知嘉子‡
Yasumasa Tanaka　Chikako Nakanishi

1. はじめに

近年，姿勢推定技術の AI モデルである姿勢推定モデルは，姿勢の改善や不審な人物の検知の必要性から，スポーツの分野や医療現場，監視カメラなどで使用されている．そこで，姿勢推定を小型でネットワークを用いず，手元のエッジ端末上で AI 技術を動作させることのできるエッジ端末で動かすことができれば，精密機械のある医療現場などでもさらに扱いやすく，監視カメラなどにも組み込みやすくなるのではないかと考えた．しかし，リアルタイム性が要求されるような応用では姿勢推定モデルは演算量が多すぎるため，性能の低いエッジ端末では処理ができない問題点がある．

そこで本研究では，CPU と FPGA が同一チップ上に存在する SoC FPGA ボードである Ultra96v2 を使用する．そして，一部の機能の処理を行うことができる回路を作成することにより低リソースで推論処理を行いつつ，CPU で処理をするソフト部と FPGA で処理をする回路部を協調動作させることにより高速化を図った．また，ソフト部と回路部のデータ共有の仕方に関しては，共有メモリを使用する．

2. 現状の回路動作

データの共有方法について説明する．図 2.1 のように現在の層から演算結果を共有メモリに送信し，その後 DMA 転送によって回路部に送信を行う．そして，回路部で実行を行った後，再度 DMA 転送によって共有メモリに返信し，その後ソフト部へのメモリに返信を行うといったことをしていた．しかし，次の層が回路部での転送を行う層であった場合はソフト部へのメモリに返信を行う時間は無駄となる．

そこで，データの共有にかかる時間を削減するため，回路化を行った Conv2D 層が連続する箇所は図 2.2 のように，ソフト部のメモリに返信をする処理を削減し，共有メモリから回路に必要な処理結果を使用している．その結果，回路化で高速化した時間も含め総実行時間が約 16023[ms]から約 1141[ms]となり，約 14 倍の高速化をすることができた[1].

また，更なるデータの共有にかかる時間を削減するために，Add 層と呼ばれる，データの加算処理を行う層に注目し，その層もソフト部で共有メモリのデータを読み出しながら加算するように，変更を行ったことにより全体の処理時間を 1151[ms]から 1113[ms]と約 38[ms]の減少に成功した[2].

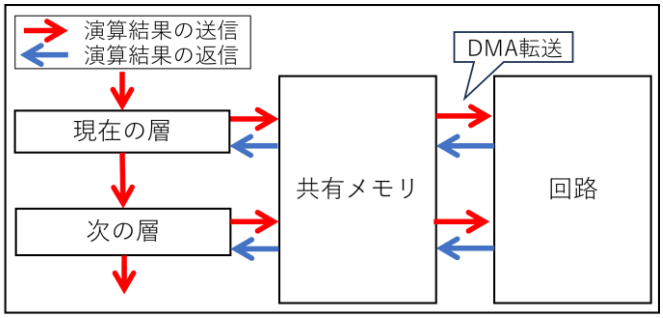

図 2.1　データ共有の流れ

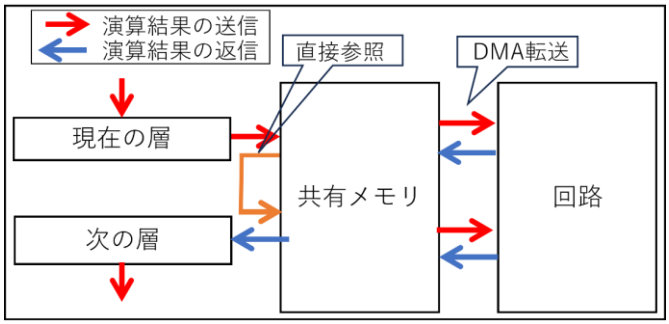

図 2.2　データ共有の短縮

3. 現状の問題点

本研究では 2021 年に Google から公開された，画像処理 AI の一つである姿勢推定モデル「MoveNet」[3]を対象として行っている．

そして，前項で述べたデータの活用は，「MoveNet」の層構成において，現在の層と次の層が回路で処理できる場合でのみ適用することができる．そのため，次の層がソフト部で処理をする場合は，今までと同じく CPU 側のメモリに共有メモリからデータをコピーすることによって行っていた．そのため，表 3.1 のように回路の実行時間のうちデータの読み出しと書き込みに約 51%と半分以上の時間がかかってしまっていた．

そこで，「MoveNet」の層構成の中で Conv2D 層に続く層を回路化することができれば，上記手法でデータのコピーをする時間を短縮することができ，更なる高速化が見込めるのではないかと考える．また，回路化することにより，演算速度も上げることができれば高速化につながるのではないかと考える．

表 3.1　回路の実行時間[ms]

書き込み時間	97.2	23%
読み出し時間	120.3	28%
回路	130.0	31%
その他	77.7	18%
総実行時間	425.2	

4. 提案手法

「MoveNet」では，図 4.1 のような Conv2D 層，DepthWiseConv2D 層(DW_Conv2D 層)，Conv2D 層の順番となっているような層構成が約 23 個ある．そのため，DW_Conv2D 層を回路化することができれば，更なる高速化を図れると考える．

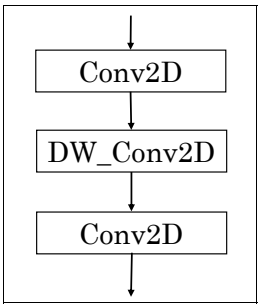

図 4.1　「MoveNet」の層の一例

4.1 DW_Conv2D 層

DW_Conv2D 層とは，畳み込み演算をする層の一種であり，通常の Conv2D 層よりも精度を上げたいときや，演算量を減らしたい時などに使用されることがある[3].

4.2 DW_Conv2D 層の回路化

現状の Conv2D 層の回路化は，汎用的な Conv2D 層用の回路[4]を「MoveNet」用に最適化させ使用している．そこで Conv2D 層用の回路に，DW_Conv2D 層用の処理をプログラムに追加を行い，現状の処理が Conv2D 層か DW_Conv2D 層かを flag で判別させ制御することによって，DW_Conv2D 層の回路を実装することとした．

5. 結果

DW_Conv2D 層を回路化した際の実行時間を表 5.1 に示す．

表 5.1　DW_Conv2D を回路化した実行時間[ms]

書き込み時間	202.6	28%
読み出し時間	202.1	28%
回路	195.5	27%
その他	118.2	16%
総実行時間	718.5	

表 5.1 より DW_Conv2D 層を回路化した結果，表 3.1 で示した回路の総実行時間より 293.3[ms]秒増加した．そして，書き込み時間，読み出し時間が増加しているのは共有メモリの削減の途中であることが原因である．

また，今までのソフト側の実行時間なども含めた全体の総実行時間と DW_Conv2D を回路化した際の全体の総実行時間を表 5.2 に示す．

表 5.2　実行時間の比較[ms]

	Conv2D層のみ	DW_Conv2D層の回路化
全体の実行時間	1066.6	995.2

表 5.2 より総実行時間は 1066.6[ms]から 995.2[ms]秒と約 71.4[ms]秒短縮した結果となった．

表 5.1，5.2 より DW_Conv2D を回路化したことにより回路の総実行時間は増えたが，全体で見た時の実行時間は減るといった結果となった．

6. 今後の展望

表 3.1，5.1 より回路の総実行時間が増えるといった結果となった．これは DW_Conv2D 層を回路に組み込んだことにより，回路の動作回数が増えたことも大きな要因と考えられるが，もう一つの要因として，書き込み，読み出し時間がかなり増えていることが挙げられる．これは，今回 DW_Conv2D 層を回路化したが，データの共有の削減が途中であることが原因である．そこで，1 で述べたデータの共有にかかる時間を削減することにより，書き込み時間，読み出し時間を削減し更なる高速化を図る．

7. まとめ

本研究では，DW_Conv2D 層を回路化することによる速度の変化と，回路化することによるメリットを示した．今後は，書き込み，読み出し時間を減らすことによる速度の変化を比較していく．

参考文献

[1] 田中康雅，中西知嘉子，”姿勢推定モデル「MoveNet」における高速化の手法の提案”，FIT2023
[2] 田中康雅，中西知嘉子，” 姿勢推定モデル「MoveNet」における処理データの活用方法”，電子情報通信学会総合大会(2025)
[3] MoveNet：動きの激しい動画向け骨格検出モデル - axinc - Medium
[4] https://qiita.com/kenichiro-yamato/items/60affeb7ca9f67c87a17 (2020 年 11 月 20 日)
[5] 大戸彰馬, 中西知嘉子, "推論処理における畳み込み処理の回路化の検討",電子情報通信学会総合大会(2022).

† 大阪工業大学 情報科学研究科 情報科学専攻
Graduate School of Information Science and Technology
Osaka Institute of Technology
‡ 大阪工業大学 情報科学部 情報知能学科
Department of Information and Computer Science Osaka Institute of Technology