C-008

エッジ AI による YOLOX の動作高速化の検討

Investigation of YOLOX operation acceleration by edge AI

池田　才慈　　中西　知嘉子‡

Saiji Ikeda†　　Chikako Nakanishi

1. はじめに

近年，IoT 技術の進展に伴い，エッジコンピューティングが注目を浴びている．エッジコンピューティングは，ユーザ近傍のデバイス（エッジデバイス）で処理を行う技術である．そして従来データセンターとの通信が必要だった処理において通信が不要となる特徴がある．これにより，通信遅延がなくなることで，リアルタイム性が向上し，セキュリティ上のリスクが低減する．また，IoT 機器の急増に伴い，クラウド上でのデータ処理が一極集中して混雑することを回避できるなど，エッジコンピューティングには様々な処理がエッジデバイス上で実現可能となることが期待されている．これは AI 処理にも当てはまるが，エッジデバイスはサーバー用コンピュータと比べて処理能力が制限されるため，高速な AI 処理が難しいという課題がある．

そこで本研究では CPU によるソフトウェア処理と FPGA によるハードウェア処理の両方を活用し，処理の大半は CPU 上で処理を行い，演算量が多く高負荷な箇所のみを FPGA による専用回路で処理を行うことで推論時間の高速化を図った．

2. 使用デバイス・使用モデル・開発環境

2.1. Ultra96-V2[1]

エッジデバイスとして，Avnet 社の Ultra96-V2 を用いた．Ultra96-V2 は SoC FPGA ボードであり，CPU と FPGA が 1 つのチップに統合された製品である．特徴として，CPU-FPGA 間の高速なデータ転送が可能な点，高効率な専用回路を作成することで FPGA による高並列で高速な処理を行える点などが挙げられる．OS は Ultra96-V2 向けに公開されている Debian 11 を使用した．

2.2. YOLOX[2]

YOLOX は，物体検出アルゴリズムである YOLO シリーズより，2021 年に発表されたモデルである．YOLOX には高精度向けのスタンダードモデルと，エッジデバイス向けのライトモデルが存在するが，本研究ではスタンダードモデルである， YOLOX-s を採用した．

2.3. 開発ツール

2.3.1. 回路生成ツール

FPGA 上に実装する回路は，C++言語で記述した回路作成用のコードを高位合成することで生成した．高位合成には Vitis HLS 2022.2[3]，回路設計には Vivado Design Suite 2022.2[4]を使用した．Ultra96-V2 への回路実装には Debian のデバイスツリー・オーバーレイを用い，FPGA のカスタムハードウェアを Linux カーネルが正確に認識し，必要な設定やドライバを動的に適用可能とした．

2.3.2 Ceras[5]

Ceras は弊研究室で開発された、学習済み ONNX モデルの推論を C++言語で実行するライブラリである。また，使用する回路における出力結果の再利用を可能にするハードウェア情報をモデルファイルに付加することができる．一般的な AI は Python 言語で創られている。しかし、Python での処理はブラックボックスな一面が大きいため、開発の検証が容易になる Ceras を開発に用いた。

2.4. 先行研究の回路[6]

2.4.2 出力データの再利用

使用回路は Conv2D 層の処理に対応した汎用的な回路である．CPU と FPGA でデータを共有する場合、両者がアクセス可能なバッファ領域（以降，共有メモリと呼ぶ）を使用する．また，共有メモリを活用した出力結果を再利用する特徴を持つ．図 1 に出力結果の再利用の図を示す．先行研究の回路では，共有メモリをブロックに分割してデータを CPU が管理する．各層がどのタイミングでどのデータを使用するかをあらかじめ把握し，使用が終了したデータを上書きすることが可能である．そして，再利用が必要なデータは上書きされないようにしている．これにより，必要な各層のデータが共有メモリ内にあることを保証でき，各層の実行結果を CPU 側の変数にコピーする必要がなくなる．

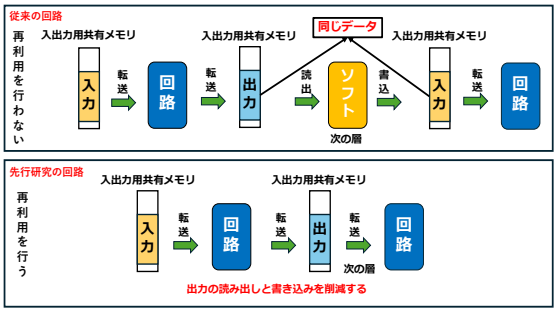

図 1　出力結果の再利用

2.4.3. データ管理の具体例

図 2 に示すように，今後使用予定のないデータについては共有メモリ上で上書きを行うことで，メモリを効率的に利用する．ただし，共有メモリで管理されるデータは，回路で実行される特定の層のデータに限定される．

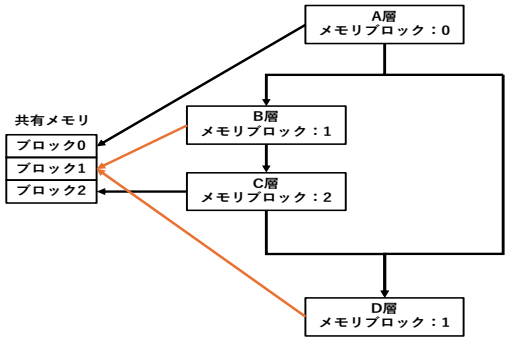

図 2　データ管理の方法

この手法により，モデル構造中の A 層の出力結果が保持されていることが保証されるため，D 層の演算時に A 層の出力結果を再利用できる．これにより，ソフトウェア側の出力用配列の領域確保，共有メモリへの書き込み，共有メモリからの読み出し処理の削減を行える．
この先行研究[6]の回路をベースに YOLOX に特化した回路を作成した．

3. 提案手法

YOLOX には Concatenate 層が 16 箇所ある．そのうち 13 箇所では次の層が Conv2D 層であり，出力データは必ず次の層である Conv2D 層で使用される．また、13 箇所のうち 12 箇所では前の層が回路で実装されている．図 3 にその層構成の一例の流れを示す．

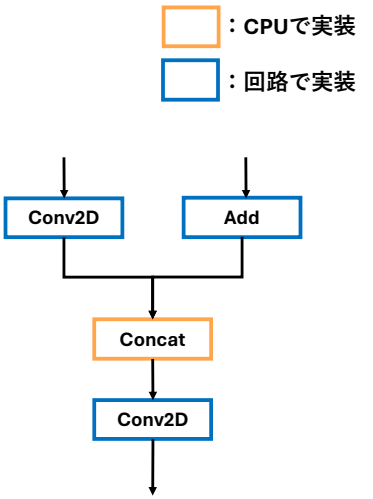

図 3 Concatenate 層の前後の層の流れ

Conv2D 層，Add 層[7]は回路で実装されており，その次の層である Concatenate 層はソフトで行われている．ソフトで処理する場合，CPU 側の変数にコピーする必要がある．CPU の実行結果は CPU 側の変数上にあり，共有メモリ上に存在せず出力結果の再利用を行うことができない．そして，処理に無駄が生じる．そこで，本研究で共有メモリを利用した Concatenate 処理を実装することで，出力結果の再利用を図った． Concatenate 層に入力される複数のデータブロックのうち、それらとは異なるブロックに対して各入力データを順次連結することで、Concatenate 処理を可能にした．CPU で行っていた Concatenate 処理を共有メモリを利用した処理で行うことにより，共有メモリの再利用を行い，ソフトウェア側の出力用配列の領域確保，共有メモリへの書き込み，共有メモリからの読み出し処理の削減を図った．

4. 結果

提案手法の効果の確認には，提案手法を適用する前である Resize 層と Add 層の回路化[7]を適用した場合と提案手法を適用した場合の推論時間を比較することで行う．時間はそれぞれを 5 回ずつ実行し，平均した値を参照する．図 4 は提案手法を適用する前である Resize 層と Add 層の回路化[7]を適用した場合と提案手法を適用した場合の推論時間を示したものである．

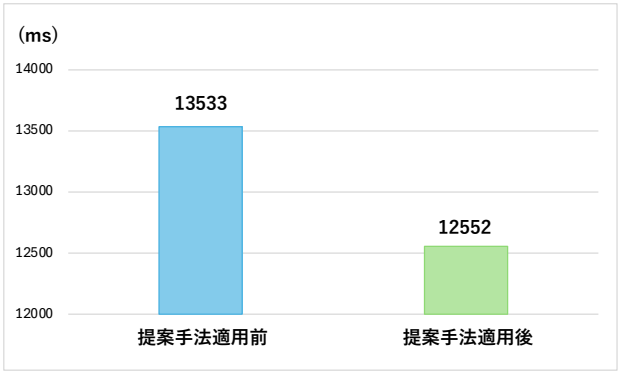

図 4 提案手法の適用前と適用後との比較

図 4 の結果から，提案手法を適用した場合，全体の推論時間は約 1026ms 短縮したことが分かる．

5. まとめ

本研究では，SoC FPGA を用いて YOLOX の高速化を行った．その結果，提案手法では約 1026ms の高速化が可能になった． 今後は，弊研究室で開発された起動回数の少ない特徴を持つ回路[8]と現在使用している回路を組み合わせることでさらなる処理時間の削減に取り組む．

参考文献

[1] AVNET, ULTRA96-V2,
<https://www.avnet.com/opasdata/d120001/medias/docus/193/5365-pb-ultra96-v2-v4a.pdf>
[2] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun, “YOLOX:Exceeding YOLO Series in 2021”, arXiv preprint arXiv:2107.08430(2021)
[3] Vitis HLS,
<https://japan.xilinx.com/support/documentation-navigation/design-hubs/2020-2/dh0090-vitis-hls-hub.html>
[4] Vivado Design Suite,
<https://www.xilinx.com/content/dam/xilinx/support/documents/sw_manuals_j/xilinx2020_2/ug910-vivado-getting-started.pdf>
[5] 西岡駿，中西知嘉子，″機械学習ライブラリの C 言語化の実現″，電子情報通信学会ソサイエティ大会(2021)
[6] 田嶋夏己，中西知嘉子，“SoCFPGA による深層学習モデル「RegNet」高速化手法の検討”，FIT2023(2023)
[7] 池田才慈，中西知嘉子，“エッジ AI による YOLOX の動作高速化の検討”，電子情報通信学会総合大会(2025)
[8] 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