Vol.2020-EMB-55 No.5
2020/12/11

高位合成によるFPGA向けDNN推論器に対する前処理の検討

田中彬義[1]　山本椋太[1]　伊藤慎治[2]　本田晋也[3]　枝廣正人[1]

概要：近年，コンピュータビジョンに対する需要が組込みシステムでも高まっており，中でも Deep Neural Network（DNN）推論はコンピュータビジョンの手法の 1 つとして注目されている．推論器の利用のために原画像データを直接扱うのではなく前処理が必要な場合がある．特定の用途向けに最適化されたハードウェア推論器を設計する方法として，C 言語記述から高位合成を用いてハードウェアを生成する方法が注目されている．しかしながら，既存手法は推論器のみに着目し，前処理について扱っている研究はない．そこで本研究では DNN 推論器の前処理を対象に高位合成を用いた設計を実施し，前処理部が DNN 推論器の処理全体にどのような影響を与えるのか調査した．結果として，SW 向けに設計した C 言語記述に最適化手法を適用することにより前処理部について LUT 数を約 55%削減し，実行時間を約 17%低減した．

1. 研究背景と目的

近年，自動運転などへの応用を背景としてコンピュータビジョンに対する需要が組込みシステムでも高まっており，中でも Deep Neural Network（DNN）推論はコンピュータビジョンの手法の 1 つとして注目されている．しかし，Huang らが述べているように原画像データを直接扱えない場合，グレイスケール化や 2 値化などの前処理が必要となる [1]．また，他にも画像のリサイズが必要となりうる．

我々の研究グループでは高位合成による FPGA 向けの DNN 推論器の研究 [2] を進めてきたが，前処理を含めた DNN 推論器の設計を行っていなかったため，本研究では前処理を含めた FPGA 向け DNN 推論器について実装，調査を行う．

我々の研究グループがすでに設計した DNN 推論器 [2] をベースとして，DNN 推論器の前処理部を C 言語にて設計し，C 記述に対する高位合成での最適化を行うことにより，**図 1** に示す前処理部を含めた DNN 推論器の設計を行う．さらに，その設計における前処理部が DNN 推論器全体に与える影響と前処理部自身の実行時間について評価する．本研究の最終目標は，我々が実現した DNN 推論器の設計 [2] において DNN 推論器で可能としている約 3800fps のスループットに対し，前処理部が悪影響を及ぼさないことで設計全体のボトルネックにならないようにすることである．

本研究では，DNN 推論器の前処理部を C 言語で記述し，システムレベル設計環境である SystemBuilder[3] を用いて HW/SW 協調設計を行う．このとき高位合成ツールとして CyberWorkBench 8.1（CWB）を用いて FPGA 向けに高位合成を行う．

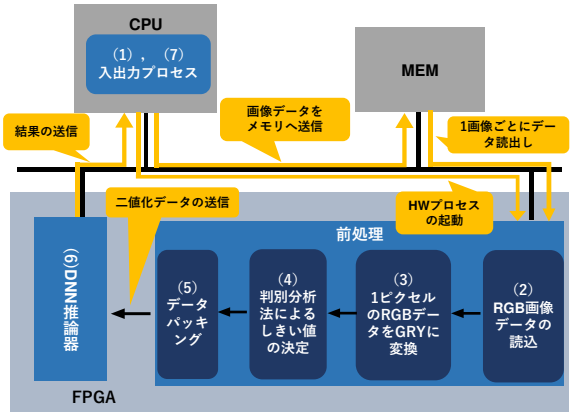

図 1　前処理部を含めた DNN 推論器の設計概要

2. 前処理部の設計

本研究では，システム全体への入力として，推論器が要求する画像サイズと同じ 32x32 の RGB 画像を想定する．また，推論器は 2 値画像を要求しているため，前処理として 2 値化が必要である．本研究における前処理部を含む DNN 推論器の構成を図 1 と以下に示す．

(1) SW 側からメモリに画像データを書き込む．書き込み後，SW が HW を起動するための同期通信を行う．

(2) HW プロセス起動後，メモリからデータを 1 画像分読み込む．

(3) Red，Green，Blue の 8bit データから構成される 1 ピクセルのデータを 8bit の GRAY データに変換する．

(4) GRAY データに対して，判別分析法 [4] を適用する．

(5) （3）にて得られたしきい値を用いて 2 値化後，データを 32bit にパッキングして，32bit ごとに FIFO を使用して DNN 推論器にデータ送信を行う．

(6) 受け取った 2 値化データを DNN 推論器が処理する．

[1] 名古屋大学
[2] システムアイ
[3] 南山大学

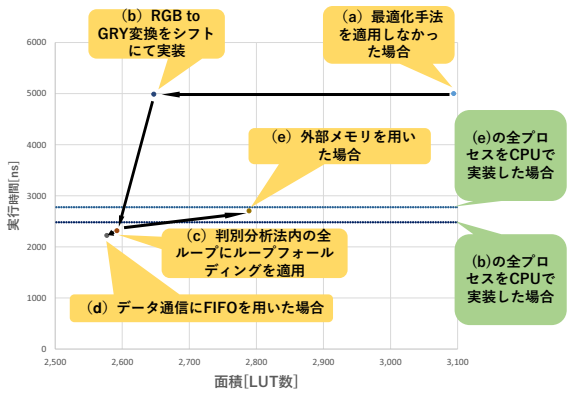

図 2 LUTs と実行時間の関係性

(7) DNN 推論器から送られた結果を出力する．

また、本研究で高速化と面積削減のため適用した手法を以下に示す．

2.1 RGB to GRY 変換でのシフト演算の利用

RGB から GRAY へのデータ変換処理において浮動小数点演算の代わりにシフト演算を採用した Kumara らによる RGB to GRY 変換アルゴリズム [5] の適用を試みる．この適用により，このデータ変換処理における面積の削減を図り，レイテンシの向上を検討する．

2.2 2 値化アルゴリズムの最適化

画像処理の 2 値化手法として広く知られる判別分析法 [4] を 2 値化アルゴリズムのベースとした．判別分析法は，しきい値によって 2 クラスに分離されたクラス分離度を最大とするしきい値を最適なしきい値として算出する手法である [4]．本研究では，判別分析法 [4] を C 言語で実装した後，ループ展開とループフォールディングの 2 つの最適化手法を適用し，高速化や面積の変化を検討する．ループ展開は，ループ内の処理を並列に行う手法である．ループフォールディングは，1 つのループが終了する以前に別のループを開始させることでループ内の処理速度を高め，処理全体の実行時間を短縮する手法の 1 つである．

3. 評価

前章で設計した前処理を含めた DNN 推論器を Xilinx 社製 SoC ZYNQ-7020 を用いて実現した．FPGA については動作周波数 100MHz で実装を行い，CPU では動作周波数 667MHz として実装した．本研究では，**図 2** に示す 7 種類の実装を以下のように行った．

(a) RGB to GRY 変換を浮動小数点演算で実装を行った．
(b) （a）の実装に対し，シフト演算を用いて実装を行った．
(c) （b）の実装に対し，図 1 （3）での判別分析法 [4] のプロセスにループフォールディングを適用した．
(d) 内部メモリ（BRAM）を用いた（c）の実装を FIFO により直接 SW から HW へデータを受け渡す実装に変更した．
(e) （c）の実装を外部メモリ（SDRAM）を使用する実装に変更した．

また，CPU のみを用いた実装の中で実行時間が最小の実装（b）と最大の実装（e）を図 2 に示す．FPGA を用いた 5 つの実装と全プロセスを CPU で行った実装とのレイテンシを比較するため点線を用いて記載した．図 2 に LUT の数と実行時間の関係性を表す．

RGB to GRY 変換でシフト演算を適用し，浮動小数点演算を用いた場合と比較した．図 2 における（a）と（b）では，レイテンシの低減は大幅には見られなかった．その一方で，面積の削減に対しては効果が見られ，具体的には浮動小数点演算による実装と比較して前処理部での LUT 数は約 16%の削減，レジスタ数においては 2%前後の削減がシフト演算による実装で確認された．

本研究では特に図 1（3）での判別分析法 [4] のプロセスにループフォールディングを適用することにより，前処理部での処理時間を図 2 の（b）と比較して約 53%までの削減が確認された．

メモリアクセスがレイテンシの増大につながるか確認するため，(c)，(d)，(e) を比較した．今回の実装では，FIFO を用いた実装が面積，実行速度共に優位性がある結果が得られたが，得られた画像データが直接メモリに格納されるなど BRAM 以外のメモリの利用が見込まれる場合では，メモリアクセスの隠蔽手法適用を検討する必要がある．

全プロセスを CPU にて処理する実装と FPGA を用いる実装との比較を行う．（d）と（e）では，CPU と FPGA での動作周波数の差が 6 倍程度あるにも関わらず，CPU での実装のレイテンシを下回る結果が確認された．しかし，適切な高速化手法を適用しなければ，全プロセスを CPU で行う実装にレイテンシに関して劣ることも確認された．

4. 今後の課題

本研究で設計した DNN 推論器のための前処理部は，最終目標とするスループットに間に合っていないため，今後の課題としてさらなる高速化が挙げられる．本研究では，判別分析法 [4] を C 言語で記述し，2 章で述べた手法を適用した．しかし，判別分析法 [4] は，1 枚の全画像データに対し，逐次的に処理を行う必要があるため，データ依存性が高く，処理の並列性を高めることに限度があると考えられる．こうした結果から今後最終目標に向けた前処理部の設計を行うには処理の並列性を高めるため，2 値化アルゴリズム自体の見直しが必要であると考える．

参考文献

[1] Huang, K.-W., Lin, C.-C., Lee, Y.-M. and Wu, Z.-X.: A Deep Learning and Image Recognition System for Image Recognition, *Data Science and Pattern Recognition*, Vol. 3, No. 2, pp. 1–11 (2019).
[2] 岡本卓也，山本椋太，本田晋也，中本幸一，若林一敏：高位合成による小規模 FPGA 向け DNN 推論器の設計，情報処理学会研究報告 (2019).
[3] 本田晋也，冨山宏之，高田広章：システムレベル設計環境：SystemBuilder，電子情報通信学会論文誌，Vol. 88, No. 2, pp. 163–174 (2005).
[4] 大津展之：判別および最小 2 乗基準に基づく自動しきい値選定法，電子情報通信学会論文誌 D，No. 4, pp. 349–356 (1980).
[5] Kumara, K., Kumar, R. and Nandanb, D.: Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC, *Procedia Computer Science*, Vol. 171, pp. 2008–2015 (2020).