2H-01

データフロー構成による高演算効率 DCNN を用いた高速移動物体の識別

李　寧 †　　高木　俊平 †　　富岡　洋一 ‡　　北澤　仁志 †
† 東京農工大学　　‡ 会津大学

1　はじめに

近年，Deep Convolutional Neural Network (DCNN) による物体識別を高速に行うことを目的とし，FPGA を用いた様々なハードウェアが提案されている [1][2][3]．多くのハードウェアは DCNN の全ての層を一度に計算できず，バッファが必要のため，メモリへのアクセスがボトルネックになる．本報告ではバッファを使用せず，データフロー構成のハードウェアを提案する．提案のハードウェアでは画像入力から最後の識別結果の出力まで全ての処理回路を一つのチップに実装し，データフローを一度も止めず，全ての層の計算を並列に行うことで，既報告の最速のものの約 2 倍の処理速度が得られた．車載カメラ映像を用いた実験では，1600 フレーム/秒以上の処理速度があり，追突危険状態の識別率は 99% に達した．

2　提案手法

提案ハードウェアは，各層に合わせて設計した演算ユニットを層ごとに配置したデータフロー構成で，バッファを使用せず，全ての層の計算を一貫して並列に行う．FPGA の実装に適するように 3D 畳み込み演算回路と Global Summation を提案実装し，最新の Recurrent Convolutional Neural Network(RCNN)[4] にも対応できる．

2.1　畳み込み層の演算回路

本研究では文献 [3] の回路をベースに 3 次元フィルタに対応する演算ユニットを設計した．積和演算回路とレジスタによって構成される Processing Element(PE) は FIFO と一本のチェーンに繋ぎ，入力データはその中で計算されながらシフトされて行く (図 1)．FIFO の深さは画像 (特徴マップ) とフィルタの幅の差によって決まる．必要な PE の数はフィルタの重みの数に等しいため，十分深い FIFO が用意できれば，任意サイズの画像を処理できる．

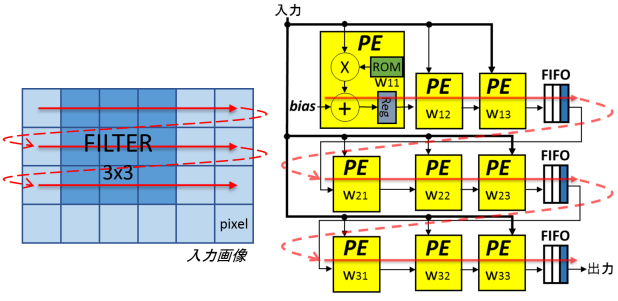

図 1　畳み込み演算回路

2.2　Max Pooling 層の演算回路

Max Pooling 層の回路は図 1 と同様な構造で，PE の積和演算回路の代わり，コンパレータを使用する．Pooling の後の層では，Pooling 層の出力の一部のみを使用するため，レジスタのチェーンを増やし，クロックごとに異なるチャンネルを順次に計算することで稼働率の低下を防ぐ (図 2)．

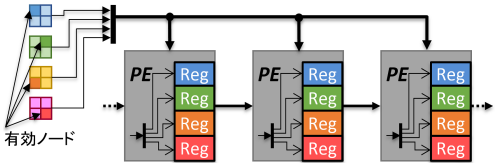

図 2　各 PE が 4 チャンネルの特徴マップを計算する (stride=2)

2.3　分類識別層の演算回路

DCNN の最終段は畳み込み層で抽出された特徴を用いて画像を識別分類する．通常はいくつかの Dense 層によって構成される．全結合の Dense 構造は膨大な量の DSP とメモリを使用するため，本研究では Global Average Pooling[5] をハードウェア実装に適するように改良した Global Summation を提案する．Global Summation 直前の層で識別種類の数と同枚数の特徴マップを生成する．Global Summation 層で各特徴マップ内のノードの総和を求め出力し，値が最も大きい特徴マップのクラスが識別結果となる．重みを使用しないため，乗算が行われず，メモリと DSP 資源が節約できる．

2.4　Recurrent Convolutional Neural Network への拡張

RCNN はフィルタ F で一回畳み込んで得られた特徴マップに，その特徴マップをさらに別のフィルタ f で t 回畳み込んだ結果を重ねあわせて出力とすることで，少数の重みで Neural Network の段数を擬似的に増やす手法である (図 3)．$t = 2$ の場合，2.1 節の回路を図 4 のように接続することで，提案のハードウェアは RCNN に対応できる (式 1)．

$$Output_{t=2} = (f^2 + f + 1) \cdot (F \cdot Input) \tag{1}$$

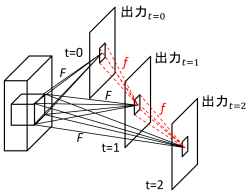

図 3　RCNN の概念図

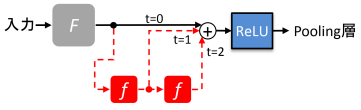

図 4　RCNN 回路の構成

3　実装結果

高速物体識別のため，図 5 の DCNN を提案した．Global Summation などを用いることで，画像入力から識別結果の出力まで全ての処理回路 (図 6) をワンチップ (Altera Stratix V

An implementation of DCNN with dataflow architecture for high speed moving object recognition
†Ning LI　†Shunpei TAKAKI　†Hitoshi KITAZAWA
‡Yoichi TOMIOKA
†Tokyo University of Agriculture and Technology
‡The University of Aizu

5SGSMD5K2F40C2) に実装できた．論理合成の結果を表 1 に示す．提案ハードウェアは最終層で 32bit，その他の層では 16bit の固定小数点を用いて計算を行う．FIFO のサイズを調整することのみで，様々なサイズの入力画像に対応できる．

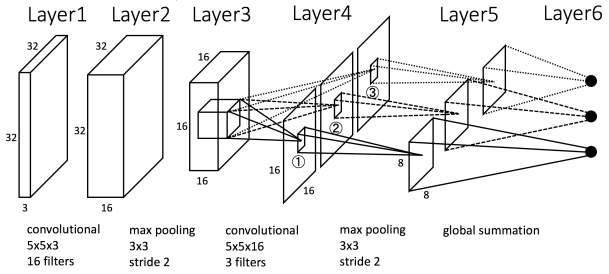

図 5　ハードウェアに実装した DCNN の構造

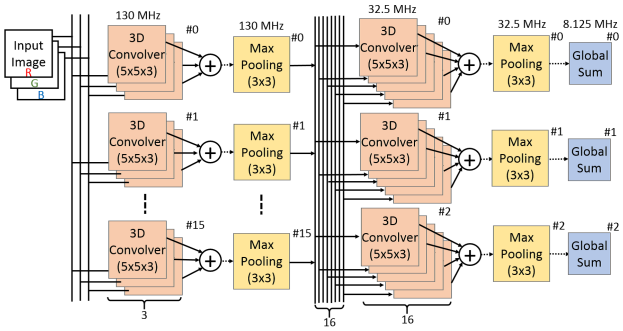

図 6　実装したハードウェアのブロック図

表 1　提案ハードウェアの資源使用量と動作速度

入力画像サイズ	32×32	32×32(RCNN)	320×240	320×240(RCNN)
ALMs	23750 (14%)	26563 (15%)	25130 (15%)	28232 (16%)
レジスタ	46339 (7%)	51945 (7%)	49964 (7%)	56450 (8%)
Block Memory Bits	196112 (<1%)	233776 (<1%)	1684496 (4%)	2044720 (5%)
DSP (27bit×27bit)	1590 (100%)	1590 (100%)	1590 (100%)	1590 (100%)
動作速度	130.63 MHz	108.65 MHz	134.64 MHz	108.27 MHz
消費電力	1113.88 mW	1132.45 mW	1116.70 mW	1136.70 mW

文献 [1] のハードウェアは 61.62 giga-operations per second(GOPS) と述べられている．ImageNet 1K[6] において，文献 [2] は [1] の約 3 倍の速度とされているため，ここでは 200 GOPS と予測する．提案したハードウェアはデータフローのアーキテクチャを使用しており，全ての層と各層の全ての特徴マップを並列に計算する．スループットは 1 pixel/1 clock のため，処理 time step は画像の pixel 数に等しい．現在，pooling 層の後の層の計算回路でデータフローが流れるチェーンの 4 多重化が行う前の値であり，有効な演算回数が 1/4 に低下するが，提案のハードウェアは既に [2] より 2 倍以上のパフォーマンス 409.62 GOPS を達成できた (表 2)．

表 2　提案ハードウェアのパフォーマンス

Layer	PEs	Equivalent Frequency	Operation	GOPS
Layer1	1200	130.000 MHz	2	312.00
Layer2	144	130.000 MHz	1	18.72
Layer3	1200	32.500 MHz	2	78.00
Layer4	27	32.500 MHz	1	0.88
Layer5	3	8.125 MHz	1	0.02
TOTAL	**2574**	-	-	**409.62**

提案 DCNN とハードウェアの精度を評価するために，いくつかのデータセットの画像を左右反転，切り出しなどの加工を行い，独自のデータセットを二つ作成した．その詳細は表 3 にまとめた．表 1 の動作速度により，提案ハードウェアは追突防止のデータセットに対し 1600 フレーム/秒以上の処理能力を持っており，高速運転のシチュエーションでも十分の能力を発揮できる．

表 3　データセットの詳細と識別率

データセット	物体識別 (図 7)	追突防止 (図 8)
画像サイズ	32×32 (RGB)	320×240 (RGB)
クラス	人，車，背景	NEAR，(FAR & 道路)
学習枚数 (各クラス)	2 万枚	700 枚
プレディクト枚数 (各クラス)	2 千枚	200 枚
画像出処	CIFAR-10, NICTA Pedestrian	Caltech Cars (Rear)
識別率	**96.2%**	**99.0%**

図 7　物体識別

図 8　追突防止

4　複数チップ構成

より大規模な DCNN に対応するためには，複数の FPGA チップを用いて処理する必要がある．提案のデータフロー型のハードウェアは，演算ユニット間の転送路の本数が少なく，書き戻す操作も必要としないため，複数チップ構成への拡張が容易である．

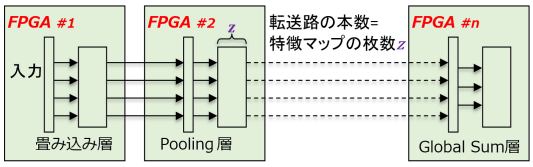

図 9　複数チップを用いる DCNN 処理ハードウェア

5　おわりに

提案手法は複数の演算ユニットを層ごとに配置し，同時に動かすことで高い並列度が得られ，既存手法の約 2 倍のパフォーマンスに達成した．今後はデータフローが流れるチェーンの多重化を行い，より複雑な DCNN の実装を目指す．また，複数チップへの拡張も今後の課題とする．本研究の一部は科研費基盤 (C)26330060 による．

参考文献

[1] C. Zhang, et al, Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks, FPGA2015.
[2] K. Ovtcharov, et al, “Accelerating Deep Convolutional Neural Networks Using Specialized Hardware.” in Microsoft Research Whitepaper, vol. 2, 2015
[3] C. Farabet, et al, “CNP: An FPGA-based Processor for Convolutional Networks.” FPL 2009.
[4] M. Liang, et al, “Recurrent Convolutional Neural Network for Object Recognition.” In CVPR, 2015.
[5] M. Lin, et al, “Network In Network.” arXiv:1312.4400, 2014
[6] Alex Krizhevsky, et al, ImageNet classification with deep convolutional neural networks. In NIPS, 2012.