

SIMD アレイ型 FPGA を用いた Multi-Stream Tracking における DCNN による物体識別

高木俊平[†] 李寧[†] 北澤仁志[†]

[†]東京農工大学

1 はじめに

防犯等を目的に多数の監視カメラが用いられるようになり、映像を有効に利用するためには撮影した映像を自動処理する技術が不可欠となってきている。そこで本研究室では FPGA を用いて数十台のカメラ映像を集中処理する Multi-Stream Tracking システム [1] を提案している。本研究では更に Deep Convolutional Neural Network(DCNN) を使用し抽出された物体の識別機能を実装する。DCNN のハードウェア実装にはデータフロー型や GPU の利用があるが、本システムでは SIMD Array 型並列ハードウェアによる移動物体抽出を行うので、同じハードウェアにより DCNN も実行する。そのために SIMD アレイ上での効率的な DCNN 実行手法やハードウェアに適した DCNN アーキテクチャを提案する。DCNN のソフトウェア実験においては歩行者・車両・背景の識別が 94% の精度で実現できた。

2 SIMD アレイ型ハードウェア

図 1 に Multi-Stream Tracking システムのハードウェア構成を示す。Processing Element(PE) 数は 20×15 個実装した。各 PE は 8×8 ピクセルのブロックを $2 \times 2 = 4$ ブロック処理でき、PE アレイ全体で 320×240 ピクセルの画像のトラッキング処理が可能である。

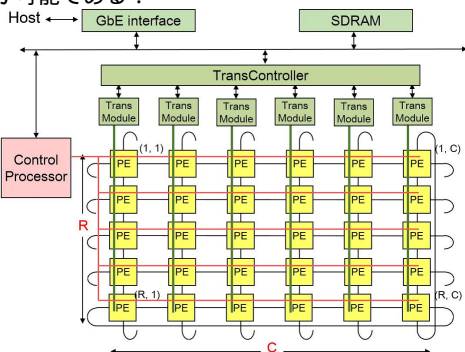


図 1 Multi-Streaming Tracking システムのハードウェア

3 DCNN アーキテクチャ

図 2 に実装する DCNN のアーキテクチャを示す。この DCNN は歩行者・車両・背景の 3 クラス識別を行うものである。入力層のサイズは PE アレイのブロック数と同じ 40×30 であり、各ブロックが 1 ピクセルを担当する。本アーキテクチャではソフトウェアでの識別率の実験により得られた結果から各層のフィルタ数(特徴マップ数)を最小限に抑えると共に、最終層での識別に Dense 層ではなく Global Summation[2] を使用することにより weight 数の大幅な削減を実現した。こ

れにより各 PE のメモリに DCNN 全体の weight を持たせることが可能となり、DCNN 実行中に外部メモリとの weight 転送を必要としないため高速化が期待できる。図 2 のアーキテクチャにおける実行に必要な weight と Bias の数は 1171 個であり、数値は 2byte で保持するので各 PE は 2342byte のメモリ領域を必要とする。

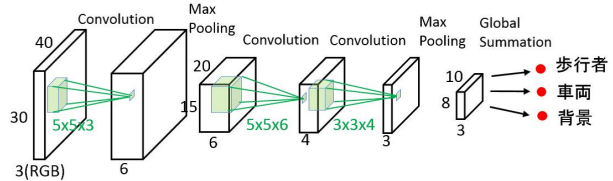


図 2 DCNN アーキテクチャ

4 SIMD アレイ上での DCNN 実行手法

4.1 PE メモリの構成

DCNN 内の演算は 16bit 固定小数点数で行う。計算中に使うメモリ領域は各ブロック毎にダブルバッファとして 2 領域確保し交互に入出力を切り替えながら層を計算していく。領域サイズは特徴マップ数の最大値分とデータ転送時のヘッダ領域を確保する。上記の weight も合わせると、1つの PE における DCNN 実行に必要なメモリ構成は図 3 のようになる。

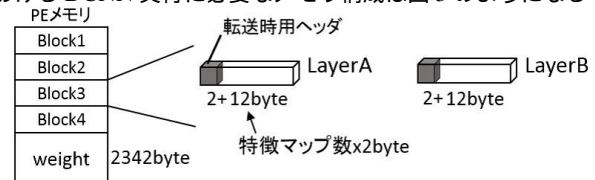


図 3 PE の DCNN 実行用メモリ領域

4.2 DCNN 入力のスケーリングと位置調整

背景差分法により抽出された移動物体を DCNN へ入力するために図 4 のように PE アレイ上で移動物体をフルスケールに拡大する。移動物体の位置をブロードキャストすることにより各ブロックが自らが取得すべきピクセル座標を求めたのち、データ転送することで拡大を行う。余白部分はゼロで埋めることで DCNN のフィルタが反応しないようにする。抽出される物体が複数のときは大きい方から指定された数取り出し、拡大処理と DCNN の実行を順次行う。なお 1 フレーム内での識別対象物体の上限は 2 個としている。

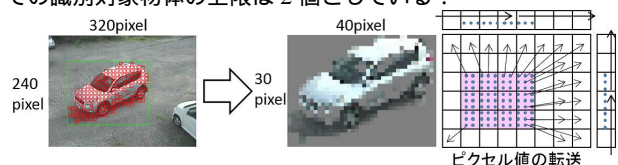


図 4 抽出された移動物体の拡大

4.3 Convolution 層

フィルタの Convolution 演算は同期シフトデータ転送 [3] を用いて図 5(a) のように各ブロックが周囲のブロックのデータ

Object Recognition using Deep Convolutional Neural Network on 2D SIMD Array Multi-Stream Tracking FPGA
[†] Shunpei Takaki, Li Ning, Hitoshi Kitazawa (Tokyo University of Agriculture and Technology)

と weight を畳み込むことにより行う．図 5(b) のようにデータを送ることで各ユニット値 x_i はフィルタ範囲内の全 PE に届く．各 PE が全ての weight を持っていることから 1 回の同期シフトデータ転送で全ての特徴マップについて $x_i w_{ij}$ を計算可能である．

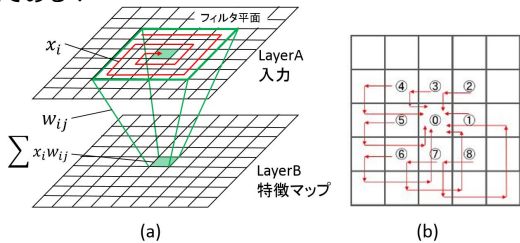


図 5 同期シフトデータ転送による Convolution 演算

4.4 Max Pooling 層

Convolution 層と同様に同期シフトデータ転送で近傍ブロックのデータを参照しながら，Pooling ウィンドウ内の最大値を選択する．なお Max Pooling はサブサンプリング処理であり次の層の平面サイズは小さくなる．図 2 の DCNN において最初の Max Pooling で層の平面サイズは 4 分の 1 になる．このとき平面サイズと PE アレイサイズが一致するので各 PE が 1 ブロックだけ処理をすれば良い．上記 DCNN は最初の Max Pooling 後に 2 回の Convolution を続けて行う．

4.5 Global Summation 層

Global Summation 層では特徴マップに表れる出力の総和の最も大きい特徴マップのクラスが識別結果となる．2 回目の Max Pooling 後にデータ転送を用いて出力の総和を求める．図 6 中の青いブロックが有効な値を持ったブロックである．PE アレイの横方向に全ブロック分シフトした後，縦方向にシフトすることで 1 つのブロックに特徴マップ毎の総和を求め，値を比較し識別結果を出力する．

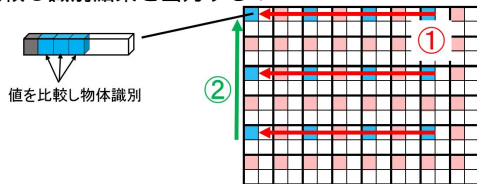


図 6 Global Summation のデータ転送順序

上記 DCNN において最終層を Dense 層とした場合との比較を表 1 で行う．Global Summation を使用することで識別率を大きく低下させることなく，メモリ量を大幅に削減できる．

表 1 Global Summation と Dense レイヤの比較

	weight 数	実行 step 数	識別率
Dense	1894	55.1kstep*	96.7%
Global Summation	1171	53.3kstep	94.0%

* PE メモリに全 weight が入りきると仮定した推測値

4.6 複数物体への対応

前述のように複数の物体が識別対象となることがあるが，図 7 のように特徴マップ方向に複数物体を入力することで 1 回の DCNN の実行で複数物体を識別出来る．メモリから読みだした weight を複数物体で共通使用することが出来るので実行 step 数を削減可能である．上記 DCNN で 2 物体を識別した際，1 物体のときの 1.6 倍の step 数で実行が完了した．なお 4.1 におけるバッファ領域は同時識別する物体数分だけ必要になる．

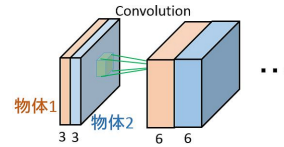


図 7 1 回の DCNN 実行における複数物体の同時識別

5 実行結果

5.1 実行 step 数および DCNN の識別性能

背景差分により抽出された 2 物体を識別する際，拡大処理が 101kstep，DCNN の実行が 84kstep であった．50MHz で FPGA を動作させた場合，3.7ms で識別処理が完了する．

図 8 にデータセットの一部を示す．NICTA Pedestrian Dataset[4] と CIFAR-10 Dataset[5] より歩行者・車両・背景の 3 クラスデータセットを作成し，学習データ 30000 枚とテストデータ 6000 枚を用意した．上記 DCNN においてテストデータに対する識別率は 94% であった．表 2 に各クラスの判別結果を示す．



図 8 データセットの一部

表 2 クラス別の判別結果

判別 \ 入力	歩行者	車両	背景
歩行者	1877	34	89
車両	44	1908	48
背景	93	54	1853

5.2 SIMD Array 型とデータフロー型の比較

SIMD Array 型とデータフロー型 [2] の比較を表 3 に示す．SIMD Array 型は汎用なプロセッサであるので並列度の低下を許せばどのような処理でも実現できる．一方，定型的な処理ではデータフロー型の方が高速である．両方の特性を踏まえ，チップ上への同時実装あるいはダイナミックリコンフィギュレーションの使用などによりこれらを組み合わせる使用するのが FPGA の効率的な使用方法であると考えられる．

表 3 SIMD Array とデータフローの比較

	特徴	利点	欠点	DCNN 実行速度 (s)
SIMD Array 型 (50MHz)	プログラム制御による PE のアレイ	移動物体抽出，フロー生成，スケールリングなど種々の機能に対応可能	PE メモリオーバーすると急激に速度低下	1.1×10^{-3}
データフロー型 (130MHz)	個々の機能に対応した専用回路のバイプライン	演算器稼働率が高い．大きな入力画像にも対応	機能は専用	7.9×10^{-6}

6 まとめ

SIMD アレイを用いて移動物体抽出を行うハードウェアにおいて，同時に DCNN による物体識別を行う手法を提案し，実行速度と DCNN の性能を示した．本研究の一部は科研費基盤 (c)26330060 による．

参考文献

- [1] R.Takasu, et al., "An FPGA Implementation of Multi-stream Tracking Hardware using 2D SIMD Array", FPGA2015.
- [2] 李寧ら, "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", 情処全大 78 回, 2016.
- [3] Y.Tomioka, et al., "FPGA Implementation of Exclusive Block Matching for ...", IEICE Vol.E97-D, 2014.
- [4] G. Overett, et al., "A New Pedestrian Dataset for ...", IEEE Intelligent Vehicles Symposium, 2008.
- [5] Alex Krizhevsky, "Learning Multiple Layers of ...", 2009.