

# HW/SW 協調設計による物体検出処理の高速化

## Realtime Implementation of Object Detection using HW/SW Co-Design

1225067 熊井 遼太 (集積システム研究室)  
(指導教員 密山 幸男 准教授)

### 1. はじめに

プログラマブルデバイスにプロセッサコアが搭載されるようになり、高性能化、高機能化が格段に進歩した。さらに、高位合成を用いた HW/SW 協調設計が容易になり、複雑な処理を 1 チップで容易に実現できるようになった。本研究では、Zynq UltraScale + MPSoC を搭載する AVNET 社製評価ボード Ultra96 を用いて、物体検出処理のリアルタイム実装を試みる。Zynq UltraScale + MPSoC は ARM CoretexA-53 を搭載したプロセッサ・システム(PS)部と論理回路を実装可能なプログラマブル・ロジック(PL)部がある。物体検出アルゴリズムとして YOLO( You Only Look Once ) [1]と HOG(Histograms of Oriented Gradients)特徴量を用いた SVM(support vector machine)を使用する。

### 2. 物体検出アルゴリズム

#### 2.1 YOLO [1]

YOLO はニューラルネットワークを用いた物体識別検出アルゴリズムである。図 1 に検出フローを示す。入力画像を分割したグリッドセル単位で処理する。Bounding Box と信頼度スコアを推測すると同時に、条件付きクラス確率を推定する。Bounding Box とクラス確率を合わせて物体を検出する。

YOLO の改良版に YOLOv2、YOLOv3 が存在する。YOLOv2 は入力画像のサイズを可変にすることで Multi-scale な学習を可能にした。YOLOv3 は、多オブジェクトの検出に弱かった課題を、異なる 3 つのスケールから特徴量を抽出することで解消した。

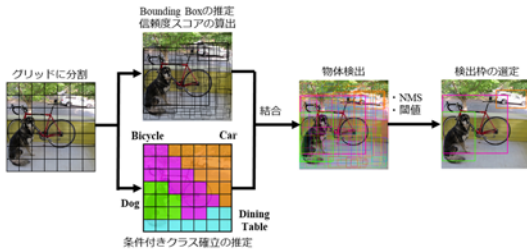


図 1 YOLO による物体検出処理フロー

#### 2.2 HOG 特徴量を用いた SVM[2]

本研究ではクラス分類に用いる特徴量に HOG を用いる。HOG 特徴量 [3]は、輝度勾配の算出、局所ヒストグラム化、局所ブロックによる正規化によって求められる。HOG 特徴量の算出フローを図 2 に示す。

算出した HOG 特徴量をもとに、SVM と呼ばれる機械学習の結果を用いた識別機器によりクラス分けを行い、人検出を実現します。

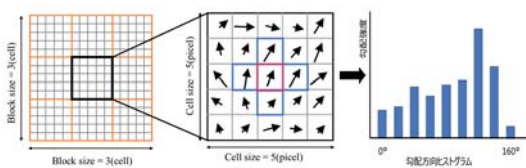


図 2 勾配強度ヒストグラムの算出方法

### 3. Tiny YOLO-v3 を用いた物体検出の高速化

YOLOv3 は演算量が多いため、本研究では YOLOv3 の演算量を削減した Tiny YOLOv3 を物体検出アルゴリズムとして使用した。Tiny YOLOv3 は、YOLOv3 に対して演算量を削減する反面、検出精度は劣る。

PetaLinux を用いて PS 部に Linux を実装し、768px×576px

のテスト画像に対して Tiny YOLOv3 による物体検出処理を行ったところ、物体検出に要した処理時間は 12.58 秒であった。このうち、浮動小数点行列乗算に要した時間は 9.71 秒であり、全体の約 81%を占めていた。そこで、処理時間を削減するために浮動小数点行列乗算のハードウェア化を行った。

Zynq UltraScale + MPSoC の全体像を図 3 に示す。Xilinx が公開している浮動小数点行列乗算 IP を PL 部に実装し、処理の高速化を行った。その結果、処理時間は 5.58 秒まで短縮できた。行列演算の処理時間は 3.24 秒まで短縮でき、約 66.63%削減できた。

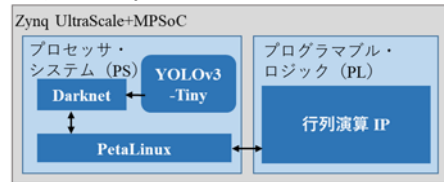


図 3 Zynq UltraScale + MPSoC の全体像

### 4. HOG・SVM を用いた物体検出の高速化

本研究ではまず SVM の高速化を行った。SVM のうち内積計算における乗算が処要の大部分を占めることが分かった[4]。そこで、内積計算における乗算を並列化して高速化を図った。

SVM の内積計算に用いるヒストグラムメモリを勾配方向別 (9 方向) に分離し、重み係数メモリと乗算器を 9 並列で実装した。図 4 に提案ブロック図を示す。テスト画像のサイズは BRAM 数の制約から 448×320 にした。以上で述べた回路を C 言語で記述し高位合成を行った。合成結果のリソース使用量を表 1 に示す。

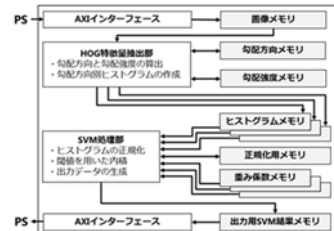


表 1 リソース使用量

Application	HOG_SVM	Available	Util [%]
BRAM18	349	432	80
DSP	70	360	19
FF	18,235	141,120	12
LUT	34,973	70,560	49

図 4 HOG 特徴量抽出・SVM 処理のブロック図

### 5. まとめ

Zynq UltraScale + MPSoC 搭載ボード Ultra96 を用いて、Tiny YOLOv3 ならびに SVM による物体検出の高速化実装を行った。Tiny YOLOv3 では浮動小数点行列乗算をハードウェア IP を用いて実装することで、処理時間を約 66.63%削減した。

SVM では内積における乗算処理をハードウェアリソース制約のもとで 9 並列化することにより処理の高速化を行った。

#### 参考文献

[1] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), May, 2016.  
[2] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," Journal of Machine learning, Sept. 1995.  
[3] N.Dalal and B.Triggs, "Histograms of Oriented Gradients for Human Detection", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2005.  
[4] 小松達也, 高位合成による設計最適化の評価とその応用, 高知工科大学, 2017年, 2月