

XSDK を用いた物体検出向け Non-Maximum Suppression の実装

1180062 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

Xilinx ソフトウェア開発キット (XSDK) は、Zynq All Programmable SoC およびソフトコアマイクロプロセッサを使用するエンベデッドアプリケーションを構築するための統合設計環境である。XSDK に含まれるツールを使用することで、ハードウェアとソフトウェアの最適な分割が可能となる。本稿では、HOG 特徴量と SVM による物体検出結果に対して、同一の検出物体に対する検出枠の重なりを削除する NMS (Non-Maximum Suppression) 処理の実装を行う。

2. 物体検出向け NMS 処理

2.1 物体検出処理

学習の際には、まず学習画像に対して HOG 特徴量を抽出し、抽出した HOG 特徴量を用いて SVM により学習する。HOG 特徴量の計算アルゴリズムは、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の 3 つの処理から構成されている。SVM とは教師あり学習の一つであり、二分類の識別器の代表的な手法である。

検出処理では、テスト画像の HOG 特徴量に対して、学習結果として得られた重みベクトルを用いて SVM を施す。SVM で得られた値に基づき、検出した座標に枠を描画し、出力する。この枠の出力する際に、同一の検出物体に対し重複して検出枠が生成されるため、NMS 処理により重複する検出枠を削除する。NMS の詳細は次節で述べる。図 1 に物体検出の処理フローを示す。

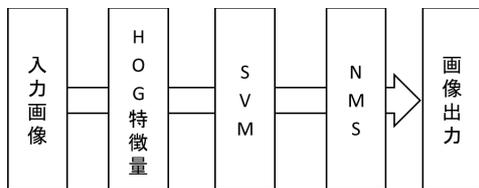


図 1: 検出処理フロー

2.2 Non-maximum Suppression

重複する検出枠の削除を、Non-maximum Suppression (NMS) と呼ばれる手法を用いて実現する。NMS は、同じクラスとして認識され、重なっている状態の領域を抑制するためのアルゴリズムである。IoU 値とは、画像の重なりを割る値を示す値である (図 2)。IoU 値のしきい値を 0.65 とすると、二つの検出枠において、SVM 処理で得た認識結果の値を比較し、重なりが 65% 以上であった場合、値が小さいほうの検出枠を削除することになる。

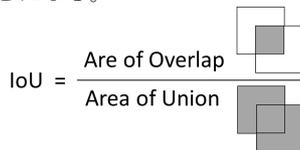


図 2: IoU 値

3. XSDK を用いた NMS 実装

Xilinx 社 Zynq-7000 (XC7Z020) をターゲットデバイスとし、XSDK を用いてハードコアプロセッサ上に NMS 処理を実装する。HOG 特徴量抽出に用いる画像情報としては輝度値しか扱えないため、カメラからの入力画像をグレースケール化する。グレースケール化した値を、AXI インタフェースを通して PL (Programmable Logic) 側に値を送る。AXI インタフェースとは、PS (Processing System) と PL 間を繋ぐメモリマッ

プ方式プロトコルである。PL で HOG 特徴量抽出と SVM 処理を行った結果は、AXI インタフェースを介して PS 側に渡される。PS では、得られた SVM 結果に対して、1 つの検出物体に対する検出枠の重なりを NMS により削除し、入力画像に最終的な検出枠を上書きして出力画像としてモニターに出力する。NMS 周辺の処理の流れを図 3 に示す。

XSDK を用いて PS 上に NMS を実装し、検出枠を削除した結果を図 4 に示す。SVM 処理結果として座標を指定した 7 個の値を生成し、うち 4 個が他の枠と 65% 以上重なるように生成した。重複した検出枠のうち、SVM 処理結果の値が最も高い検出枠を残し、削除できていることを確認した。

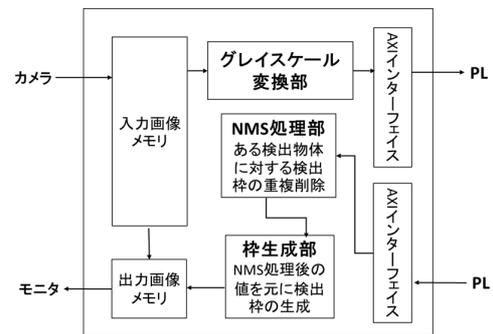
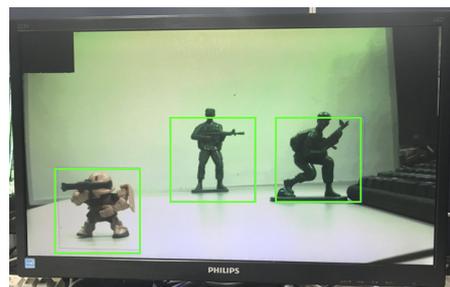


図 3: NMS 周辺の処理の流れ



(a) NMS 処理前



(b) NMS 処理後

図 4: NMS の適用例

4. まとめ

本研究では、XSDK を用いて HOG 特徴量と SVM による物体検出結果に対して、同一の検出物体に対する検出枠の重なりを削除する NMS 処理を、XSDK を用いて実装した。実行結果より、同一の検出物体に対し一定割合以上重なっている検出枠に対して、SVM から得たスコアが最も高い検出枠のみ残して削除できることを確認した。

今後の課題として、様々な大きさの物体検出をリアルタイムで処理するために、NMS 処理を関数化することで再帰呼び出しを可能にし、さらに AXI インタフェースを介した値の送受信方法についても再検討する。