

1. はじめに

本研究では、(1) 色の類似性に基づいた画像認識の高精度化と、(2) 物体検出の高速化に関する研究について取り組んだ。本稿では、(2) の物体検出の高速化に関する研究について述べる。

物体検出は入力画像に対して検出ウィンドウをラスタスキャンし、各検出ウィンドウから抽出した特徴ベクトルを線形 SVM 等の識別器に入力して判定することで実現している (図 1)。VGA の画像に対して、ラスタスキャンにより発生する検出ウィンドウは約 2 万個と多く、物体検出の高速化には特徴抽出と識別の各処理を高速に計算する必要がある。そこで、本研究では線形 SVM の近似計算法 [1] を導入し、識別器の演算量を削減することを目的とする。提案手法では検出ウィンドウに対して、近似計算結果による早期判定を導入することでラスタスキャンの高速化を実現する。また、入力特徴量のバイナリコードにおいて共起表現を用いることで、識別性能の高精度化を実現する。

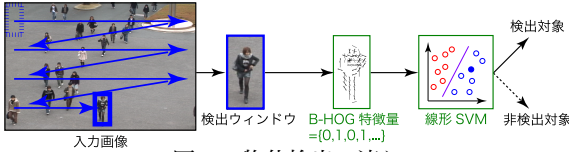


図 1: 物体検出の流れ

2. 物体検出の識別器の高速化

本研究では、線形 SVM を近似計算して識別器の高速化と、近似計算の打ち切りによるラスタスキャンの高速化を実現する。

2.1 近似計算法による識別器の高速化

線形 SVM により学習した識別器は式 (1) に示すように、重みベクトル w と入力特徴ベクトル x の内積計算により出力値 $F(x)$ を算出する。

$$F(x) = w^T x \quad (1)$$

このような実数ベクトル間の内積計算は計算コストが高い。そこで、Hare 等によって提案された近似計算法 [1] を利用して線形 SVM の高速な識別を実現する。図 2 に示す近似計算法により、重みベクトル w を実数ベクトル β と基底バイナリコード b に近似する。近似計算法は、ビット演算を利用して識別計算を行うことで高速な識別が可能となる。

1. 初期化：線形 SVM の重みベクトル w を実数ベクトル r に格納
 $r = w$
 2. w を実数ベクトル β と基底バイナリコード b に近似
For $i = 1, \dots, N_b$ // N_b : 基底数
 - ・ 実数ベクトル r を $\{-1, 1\}$ に 2 値化して基底バイナリコード b_i を算出
 $b_i = \text{sign}(r)$
 - ・ 実数ベクトル r と基底バイナリコード b_i の内積
 $\beta_i = \frac{\langle r, b_i \rangle}{\|b_i\|^2}$
 - ・ 近似による残差を実数ベクトル r に代入
 $r \leftarrow r - \beta_i b_i$
- End for
Return $\{\beta_i\}_{i=1}^{N_b}, \{b_i\}_{i=1}^{N_b}$

図 2: 重みベクトル w の近似アルゴリズム

線形 SVM は重みベクトル w を近似した実数ベクトル β と基底バイナリコード b を用いて、 $F(x) \approx f(x) = \sum_{i=1}^{N_b} \beta_i b_i x$ と近似することができる。さらに、式 (2) に示すように基底バイナリコード b を $b^+ \in \{0, 1\}^N$ と $\bar{b}^+$ に分解 ($b = b^+ - \bar{b}^+$) することで、内積 $\langle b_i^+, x \rangle$ とノルム $|x|$ を利用した式に変形できる。

$$f(x) = \sum_{i=1}^{N_b} \beta_i b_i x = \sum_{i=1}^{N_b} \beta_i (2 \langle b_i^+, x \rangle - |x|) \quad (2)$$

ここで、入力特徴ベクトル x がバイナリコードである場合、内積 $\langle b_i^+, x \rangle$ とノルム $|x|$ は論理積とビットカウントで計算することが可能である。バイナリコードの論理積は実数に比べて高速に演算することができる。また、ビットカウントは SSE4.2 から CPU に直接実装されている POPCNT 関数を用いることで、非常に高速な演算が可能である。

2.2 近似計算の打ち切りによるラスタスキャンの高速化

線形 SVM の近似計算では高次元の特徴ベクトルを扱う場合、基底数 N_b を多く用いる必要がある。基底数 N_b による識別精度と速度はトレードオフの関係にあり、膨大な検出ウィンドウを高速に処理するためには基底数 N_b をできるだけ削減したい。そこで、本研究では近似計算過程における計算結果から早期判定を導入することで、ラスタスキャンの高速化を行う。

少ない基底数 N_b で近似した出力値が関数マージンより離れているサンプルは、基底数 N_b を増やしても識別結果が変わることはない。そこで、図 3 のアルゴリズムにより識別器の近似計算を打ち切り、識別結果の早期判定を行う。

1. 入力画像 I に対して検出ウィンドウをラスタスキャン
For $k = 1, \dots, K$ // K : 検出ウィンドウの総数
 - ・ 検出ウィンドウ $I(k)$ から特徴ベクトル x_k を抽出
 2. 識別器の出力値を算出
For $i = 1, \dots, N_b$ // N_b : 最大基底数
 - ・ 線形 SVM の近似計算
 $f(x_k) += \beta_i (2 \langle b_i^+, x_k \rangle - |x_k|)$
 - ・ 計算結果を判定
If $f(x_k) > th^{pos}$ or $f(x_k) < th^{neg}$
Break // 近似計算の打ち切り
 3. 閾値 th により判定
 $y_k = \begin{cases} 1 & \text{if } f(x_k) > th \\ -1 & \text{otherwise} \end{cases}$
- End for // ラスタスキャン終了
Return $y_1, y_2, \dots, y_K$

図 3: 提案手法による物体検出アルゴリズム

近似計算を打ち切る閾値 th^{pos} , th^{neg} には、学習により求めた線形 SVM の識別境界 th から ± 1.0 の出力を持つサポートベクタを使用する。提案手法は、図 4 に示すように、識別器の出力が関数マージンから離れたサンプルの近似計算を打ち切り、関数マージン内の識別が困難なサンプルは基底数 N_b を増やし近似計算を継続する。これにより、膨大な検出ウィンドウに対して線形 SVM の性能を維持しつつ高速な識別計算が可能となる。

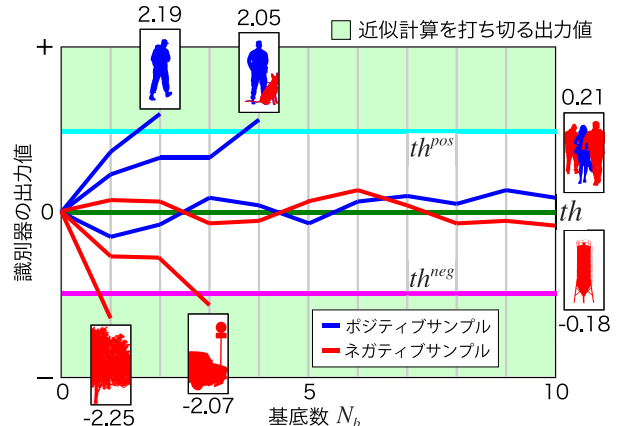


図 4: 線形 SVM の近似計算の打ち切り

2.3 B-HOG 特徴量の共起表現による高精度化

提案手法では、バイナリコードの特徴ベクトルに B-HOG 特徴量を利用する。B-HOG 特徴量は閾値処理により HOG 特徴量を 2 値化したバイナリコードで物体形状を表現する。B-HOG 特徴量は HOG 特徴量の 1/8 のメモリ量で特徴表現が可能であるが、2 値化の際の量子化誤差の影響により表現能力が低下する問題がある。そこで、図 5 に示すように、ブロック領域で論理演算を用いて共起表現したコードを追加して特徴表現能力を向上を目指す。2 つのセルの

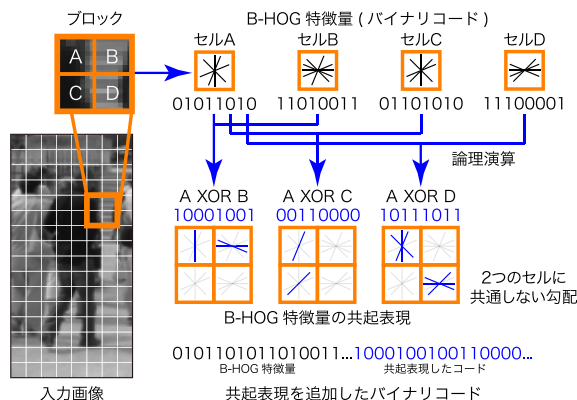


図 5: 論理演算による B-HOG 特徴量の共起表現

バイナリコードを論理演算することで、セル間の勾配の関係性を表現することが可能となる。論理演算子には論理積 (AND)、論理和 (OR)、排他的論理和 (XOR) を用いる。例えば XOR の場合、2 つのセルに共通しない勾配を“1”のビットで表現する。共起表現を追加した特徴量は、3,360 ビットの B-HOG 特徴量に対して、5,040 ビットのバイナリコードを結合するため、8,400 ビットのバイナリコードとなる。本研究では、近似計算法により高速な識別を行うため、B-HOG 特徴量を共起表現したバイナリコードを入力特徴ベクトル x として用いる。

3. 評価実験

提案手法の有効性を示すために識別精度と処理時間を従来法と比較する。

3.1 実験概要

データセットには INRIA Person Dataset を使用する。学習サンプルにはポジティブサンプル 2,416 枚、ネガティブサンプル 13,161 枚を使用する。評価サンプルにはポジティブサンプル 1,126 枚、ネガティブサンプル 9,060 枚を使用する。識別精度の評価には Detection Error Trade-off (DET) カーブを用いる。DET カーブは横軸に誤検出率、縦軸に未検出率を表しており、原点に近いほど高精度であることを表す。実験環境は、OS: Windows Server 2008 Enterprise x64, CPU: Intel Xeon CPU X7542 @ 2.67GHz, RAM: 256GB を用いる。評価実験では予備実験の結果より、基底数 N_b は 16 とする。

3.2 実験結果：識別精度の比較

実験結果の DET カーブを図 6 に示す。DET カーブより、線形 SVM を用いた B-HOG 特徴量は HOG 特徴量と比較して識別精度が低下している。一方、提案手法 (AND) 及び (OR) は B-HOG 特徴量と同様のバイナリコードの特徴量であるが、HOG と同等の識別率を得た。さらに、提案手法 (XOR) は誤検出率 0.1% において従来法 (HOG 特徴量+線形 SVM) と比較して約 8.1% 識別精度が向上している。これは、XOR による 2 つのセルに共通しない勾配の表現が人検出に効果的に働いたためと考えられる。

3.3 実験結果：処理速度の比較

表 1 に、共起表現 (XOR) した特徴ベクトルを用いたときの各識別器の検出率と処理時間を示す。近似計算法は基底数 $N_b = 16$ のとき、線形 SVM の検出精度と同等である。処理時間を比較すると、近似計算法は線形 SVM よりも約 3 倍高速な演算が可能である。提案手法は近似計算法よりもさらに速く、線形 SVM と比較して約 17 倍高速な演算を実現した。このとき、提案手法は平均で 2.17 回の基底数で近似計算を打ち切り早期判定している。基底数 $N_b = 2$

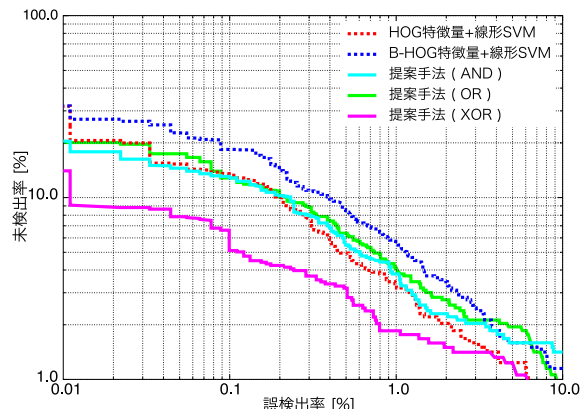


図 6: DET カーブ

のときの近似計算法は、提案手法と同じ処理時間となるが検出率は低下する。

表 1: 各識別器の検出率 [%] と処理時間 [ms]

識別器	検出率 [%]	処理時間 [ms]
線形 SVM	95.05	0.034
近似計算法 ($N_b=16$)	94.96	0.013
近似計算法 ($N_b=2$)	92.76	0.002
提案手法	94.96	0.002

図 7 は、近似計算の打ち切りの割合を示している。図 7 より、近似計算の打ち切りにより基底数 N_b が 5 までに約 5 割のポジティブサンプルが早期判定されている。また、ネガティブサンプルは基底数 N_b が 1 のときに約 9 割のサンプルが早期判定されている。図 7 に示す画像は、任意の基底数にて早期判定されたサンプル例と平均勾配画像である。基底数が少ない段階では、各クラスらしいサンプルの識別結果が早期判定され、異なったクラスに分類されやすいような識別が困難なサンプルは最大まで ($N_b = 16$) 継続して近似計算が行われる。このように、提案手法は入力サンプルに応じて適応的に識別結果を早期判定することで、線形 SVM の性能を維持しつつ高速な識別を実現した。

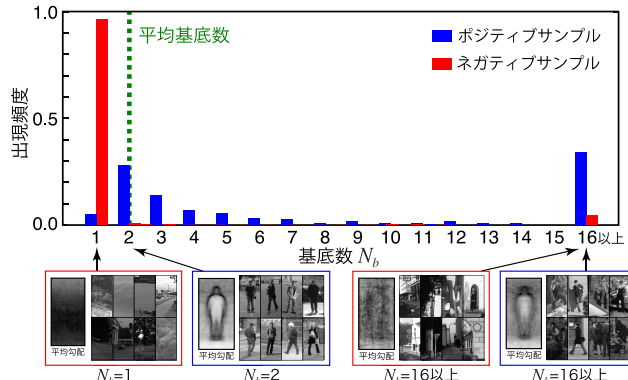


図 7: 近似計算の打ち切りの割合

4. おわりに

本研究では、入力サンプルに応じて適応的に線形 SVM を近似計算して識別結果を早期判定することで、線形 SVM の性能を維持しつつ高速な識別を実現した。今後は、バイナリコードによる特徴ベクトルの抽出過程を高速化し、特徴量と提案手法の識別器を組み合わせるリアルタイムな処理可能な物体検出器を設計する予定である。

参考文献

- [1] S. Hare, A. Saffari and P. H. S. Torr, “Efficient online structured output learning for keypoint-based object tracking”, Computer Vision and Pattern Recognition, pp.1894–1901, 2012.

研究業績

- [1] Yuhi Goto, Yuji Yamauchi and Hironobu Fujiyoshi. “CS-HOG : Color Similarity-based HOG”, 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV2013), GO2-2, pp.299–304, 2013.
(他 学会口頭発表 2 件, 研究会技術報告 1 件)

受賞

- [1] SSII2012 オーディエンス賞, 優秀学術賞