

1. はじめに

近年、人検出には Dalal らが提案した HOG 特徴量 [1] が用いられている。しかし、HOG 特徴量は特徴次元数が多いため、大量のメモリが必要となる。そこで、本研究では Real AdaBoost に基づく HOG 特徴量の効率的な削減を目的とする。

2. バイナリパターン化による HOG 特徴量の削減

提案手法は、HOG 特徴量を抽出後、閾値処理によりバイナリパターン化し、特徴量数の削減を行う。このとき、学習サンプルが少ないと識別時に用いる確率密度分布が疎になるという問題がある。そこで、本研究では Real AdaBoost に基づいたバイナリパターン化の統合を行い、密な確率密度分布を生成する方法を提案する。

2.1. HOG 特徴量のバイナリパターン化

HOG は、セル領域における輝度の勾配方向をヒストグラム化した特徴量である。1つの検出ウィンドウ (30 × 60 ピクセル) から求められる特徴量数は 2,880 個となり、1個の特徴量は 8 バイト (浮動小数点型) であるため、計 23,040 バイトのメモリを必要とする。

特徴量数とメモリを削減するために、1セルより得られる 8 方向の HOG 特徴量を固定閾値によりバイナリパターン化する (図 1 参照)。1つの検出ウィンドウの特徴量数は 360 個あり、1個の特徴量は 1 バイト (符号なし整数型) であるため、バイナリパターン化することにより計 360 バイトとなる。浮動小数点型と比べメモリは 64 分の 1 となる。

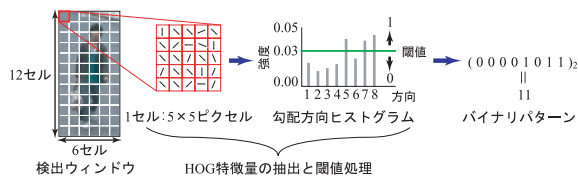


図 1: HOG 特徴量のバイナリパターン化

2.2. バイナリパターン化による問題

バイナリパターン化した特徴量を抽出後、Real AdaBoost により識別器を学習する。図 2(a) は、学習過程において生成された確率密度分布であり、確率の低いバイナリパターンが多く、確率密度分布が疎であることがわかる。これは、8 方向で 1 個のバイナリパターンを生成することにより、多数のパターン (256 パターン) が得られるためである。これにより、以下の 2 つの問題が発生する。

問題点 1: 弱識別器の極端な出力 ポジティブクラスとネガティブクラスの学習サンプルから生成される確率密度分布 W_+ , W_- において、一方のクラスの確率が 0 に近い値の場合、式 (1) で表わされる弱識別器の出力 $h(x)$ が極大値、もしくは極小値となる。

$$h(x) = \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon} \quad (\epsilon = 10^{-10}) \quad (1)$$

このとき、識別に用いる弱識別器の線形和が極大値や極小値に依存する可能性があるため、多くの未検出や誤検出が発生する恐れがある。

問題点 2: 疎な確率密度分布 図 2(a) に示す A や B, C のような確率が低く、クラス間の確率の差が小さいバイナリパターンは識別に不要である。このようなバイナリパターンが存在するために、確率密度分布が疎となる。

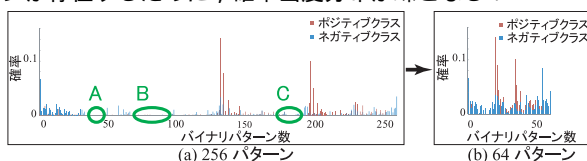


図 2: 1 ラウンド目の学習で選択された確率密度分布

2.3. Real AdaBoost に基づくパターンの統合

2.2. 節で述べた問題を解決するために、バイナリパターンを統合することにより、図 2(b) に示すような密な確率密度分布を生成する。

問題点 1 への対応 弱識別器の出力 $h(x)$ の絶対値が 2 以上である場合、出力値が極端な値であると判断し、他のバイナリパターンと統合する。

問題点 2 への対応 図 2(a) に示す A, B, C のような確率が低く、かつクラス間の確率の差が小さいバイナリパターンは、識別に不要であるため、確率の差の絶対値が最小であるバイナリパターンを他のバイナリパターンと統合する。

統合する際は、クラス間の確率の差を算出し、符号が変化しないバイナリパターンと統合する。提案する Real AdaBoost に基づく統合アルゴリズムを図 3 に示す。

```

For bin = BIN, BIN-1, BIN-2, ..., 64  BIN はバイナリパターン数
1. 削減対象のパターンの探索
   If max(|h_j(x)|) ≥ 2  j = arg max(|h_j(x)|)
                        0 ≤ j ≤ bin
   Else
                        j = arg min(|W_+^j - W_-^j|)
                        0 ≤ j ≤ bin
2. W_+^j と W_-^j の差を表す r_j を算出
   r_j = W_+^j - W_-^j
3. 統合対象の探索
   For o = 1, 2, ..., ORIENTATION  ORIENTATION は勾配方向数
   3.1 o 個目の勾配方向のバイナリを * で表現
   3.2 全バイナリパターンから統合対象候補 j' を探索
   3.3 W_+^j と W_-^j の差を表す r_{j'} を算出
       r_{j'} = W_+^{j'} - W_-^{j'}
4. r_j - r_{j'} が最も 0 に近い j' を選択
5. j を j' に統合
    
```

図 3: Real AdaBoost に基づくバイナリパターンの統合

統合されたバイナリパターンは、“*”により表現される。00000000 と 00000001 の場合、値の異なるビット (下一桁) が統合され “*”により表現されるため、0000000* となる。図 2(b) に、提案手法によりバイナリパターンの統合をした結果を示す。識別に重要でないバイナリパターンを統合することで、密な確率密度分布となりメモリ量を削減することが可能となる。

3. 評価実験

評価実験では、従来法である HOG 特徴量、ベクトル量子化 (VQ)、提案手法の識別精度とメモリ使用量の比較を行う。学習用には、ポジティブサンプル 2,054 枚、ネガティブサンプル 6,258 枚、評価用には、ポジティブサンプル 1,000 枚、ネガティブサンプル 1,234 枚を用いる。表 1 に誤検出率 5.0% における検出率と画像 1 枚 (720 × 480 ピクセル) に対する特徴抽出に必要なメモリ量を示す。

表 1: 検出率と画像 1 枚に対し必要なメモリ量の比較

手法	検出率 [%]	メモリ量 [MB]
HOG [1](連続値)	74.3	1,098.63
HOG(64 クラスタに VQ)	63.8	198.75
提案手法	256 パターン	17.17
	128 パターン	
	64 パターン	
	32 パターン	

提案手法は、従来法に比べ必要なメモリを約 2.0% に削減し、かつ同等以上の識別精度を得ることができた。

4. おわりに

HOG 特徴量をバイナリパターン化し、精度を維持したまま特徴量数を削減することができることを確認した。今後は、リアルタイム人検出の実現について検討する予定である。参考文献

[1] N. Dalal and B. Triggs: “Histograms of Oriented Gradients for Human Detection”, CVPR, vol. II, pp. 886-893, 2005.