第3章 Haar-like特徴量の判定器/TLD物体追跡/深層学習で顔検出

画像認識などのデータ処理

森岡 澄夫

写真1　画像認識処理の出力は枠線表示する

IoTデバイスで画像などといった大容量データを処理しようとすると，ラズベリー・パイ・クラスのボードが必要になってきます．ラズベリー・パイ4は，ラズベリー・パイ3と同等の消費電力で2～3倍の速度を出すことができ，画像処理用IoTデバイスとしては最有力候補です．しかし，画像の学習や認識など重い処理をしようとすると実用上ぎりぎりの速度性能であるため，プログラミングでの高速化テクニックはまだ当面は必要です．

実験してみる画像認識処理

画像認識処理の速度評価について述べます．今回行う認識処理は次の3つです．

1，Haar-like特徴量の判定器を用いた顔検出(1)
2，TLD物体追跡（オンライン学習）による顔追跡(2)
3，深層学習結果を用いた顔検出(3)

いずれもOpenCVに処理系やデモ用データが含まれているので，それらを使います．先述の通りOpenCVのバージョンは4.3.0で，ビルド時にTBB（処理並列化ライブラリ）(4)を利用するためのオプションWITH_TBBを指定しています．つまりマルチコアへの対応はOpenCVに全面的に任せることとし，人手で明示的に処理並列化をするような最適化は，今回はしていません．

実行出力の様子はいずれも似ており，**写真1**のように検出した物体に枠線を描いてリアルタイム表示します．

実験1，Haar-like顔検出

● あらまし

これは複数種類の単純な識別器を直列に（カスケードで）適用していくことで対象物を発見するものです．各識別器は，文献(1)に示されている矩形領域内明暗差（Haar-like特徴量）の判定を行います．画像内において対象物はさまざまな大きさで写っています．そこで，「一定サイズの対象物を画像全体に渡ってスキャンした後，少し画像を縮小し，再度スキャンする」という繰り返し動作をします．このため，画像サイズが大きくなると急激に実行時間が増加します．

このHaar-like顔検出は，OpenCVのかなり初期のころから物体検出の代表的デモとして同梱されています．パソコンではともかく組み込み装置にとってはとても重い処理であり，初期のラズベリー・パイ1ではVGA画像での検出処理に4秒以上もかかり，実用性に疑問符がつくものでした（当時はOpenCV2.3.1とNOOBS1.2.1を使用）(5)．今でもESP32やARMマイコンなどの能力ではまだ難しいでしょう．

● プログラム

OpenCVによる処理実装は簡単です．前章リスト2の画面キャプチャ用コードに対し，**リスト1**のコードを追加します（後述の他例でも同様）．cv::CascadeClassifierクラスのインスタンスを生成し，初期化の際にデモ用の識別器データをロードしておきます．検出はdetectMultiScale()メソッドを使って行います．画像の縮小レートや検出物体の最小サイズなど幾つかの設定パラメータがありますが，**リスト1**では検出精度を高めにした数値にしてあります．