

ラズベリー・パイで画像処理

中村 仁昭



写真1 第4部の記事の一部をラズベリー・パイ4で動かした



写真2 第4部の記事の一部をラズベリー・パイ Zero 2 Wで動かした

● トライすること

Windows環境のC++で作成された画像処理ソフトウェアをラズベリー・パイで使うため、Pythonで動作するよう対応しました。具体的には、第4部に掲載された吉田氏の記事、

- ハールライク特徴とカスケード接続による物体検出、顔検出(第6章)
- フレーム間差分+ α で確実に異常を検知(第7章)
- HSV表色系を使って病害検知! 農業見守りへ(Appendix1)
- テンプレート・マッチングを使用した酒瓶の充填量検査(第3章)

のC++プログラムをラズベリー・パイ4およびラズベリー・パイ Zero 2 Wに移植しました(写真1, 写真2)。

実験環境

● OS, カメラ

Raspberry Pi OSは執筆時点で最新のbookwormの64ビット版を選択しました。OpenCVは、現在ではaptでインストールできます。

```
$ sudo apt install libopencv-dev
python3-opencv
```

カメラは手元にあったPiCamera V2を使っています。カメラ・モジュールを使用するためのpicamera2ライブラリはOSインストール時点で入っています。

● 画像処理はNumPyで

純粹にOpenCVとC++の機能のみで実装されていれば、そのままラズベリー・パイでも動作させることは可能ですが、よりラズベリー・パイ的なハードルが低い感じで実装したかったためPythonを選択しました。ただ、OpenCVの範ちゅうを越えて画素にダイレクトにアクセスするにはPythonそのままでは重いため、画像処理はNumPyで行っています。

現在のPythonは動的に解釈を行うためコンパイル言語に比べてループ処理が桁違いに遅くなります。このため画像処理など演算対象が多い場合、可能な限りループを避ける必要があります。

ループを避けるに当たってNumPyの機能を使う必要があります。NumPyは配列同士の演算を効率的に処理できるufuncという機能を提供します。これは通常の演算子を用いることで自動的に呼び出され、ループ処理よりも高速な計算を実現します。また、ブロードキャストにより異なる形状の配列であっても、自動的にサイズを調整して演算を実行できます。これにより、スカラ値と配列の演算も簡単に行えます。配列の比較によりTrue/Falseのマスクを生成して特定の要素を更新することもできます。

C++からPythonに変換するに当たって注意した点などを説明していきます。