

お勧め! Python 画像処理

澤田 英宏

きっかけ

● 画像処理と言えばC++だったけれど

現代は多種多様なプログラミング言語が存在していて、多くの学習時間を必要とせずとも、画像処理や機械学習のプログラミングが可能となってきました。

以前であれば、これらの研究開発は基本的にはCかC++で行い、学校や企業で専門の教育を受けた人だけが仕事に携わっているような状況でした。皆さんも特別な環境で商品開発が行われているというイメージを持っていませんか。

筆者も画像処理系のAIプログラミングを行います。C++系の開発環境を使用することがほとんどです。その内容は、ハードウェア寄りの知識も必要なC言語系の仕様を考えると、プログラミング初心者の方には少しハードルが高いと考えていました。

● 経験が浅い人にも画像処理を体験してほしい

確かに、深い知識をもって開発することは良いことだと思いますが、それが置かれた環境で唯一の方法でない限り、多くの学習時間を費やして開発に臨むのは、仕事であっても時間に追われる現代の開発事情を考えると困難ですし、そもそも楽しくないと筆者も感じます。

今回の特集では、プログラミング経験が少ない方にも、これならなんとかかなりそう、実行すると楽しそう、と感じていただくことを目指しました。

そんな目的で、画像処理からAI・機械学習までサポートしているフレームワークOpenCVと、習得しやすい開発言語Pythonを使って、画像処理の基本から応用まで、すぐに使える100超のプログラムを紹介します。

初心者の方を対象に段階的に理解していただけるように順を追って紹介していますが、興味のあるところだけをピックアップして動作確認していただいても大丈夫です。まずは、楽しいと思えるところから確認してみてください。

Python 画像処理を勧める理由

画像を使用した機械学習関連の情報を検索すると、Pythonで解説されたものが多数ヒットします。

知りたいのは別言語の情報だったりするのですが、あまりにも多いので初めは興味本位でPython情報を確認してみました。するとなんと分かりやすく、サポートしているライブラリの多いことかと感じました。そこで食わず嫌いな筆者が感じた「Pythonを使うことのメリット」を考えてみました。

正直なところ、筆者はPythonを使用した開発に携わることがほとんどありません。執筆開始時点で得意な開発言語と言える状況ではありませんし、しょせんスクリプト言語と少し軽く見ていた状況から、「Pythonいいね!」に転んで、お勧めするようになった理由をつづってみました。

● 1. 世界的にも人気…情報が多い

グーグルの検索ランキングでも、Pythonは常に上位にランキングされるようです。2020年10月時点で人気1位の言語であることは間違いなく、実際にインターネットの検索だけで知りたい情報はほとんど入手できる状況にあります。

ここまで人気の開発言語であれば、周囲にもPythonを趣味や仕事で使われている方も必然的に多くなり、さらに情報入手は容易になる状況になっているようです。

筆者も、過去には今では誰も知らないと思われる開発言語で仕事していたことがあります。当時でもマイナな開発言語で、情報入手に苦労した経験があります。Pythonは世界的に人気があり、多くの人が使っているという事実がどれだけ多くのメリットがあるかを身に染みて理解することになりました。

● 2. 多くのライブラリが提供されている

人気の開発言語ならば当然、多くのライブラリが提供されます。卵か鶏かの議論と同じで、ライブラリが

リスト1 Pythonでカメラ取得映像をHSV変換する例
開発環境はPyCharmを使用

```
import cv2

if __name__ == '__main__':
    cap = cv2.VideoCapture(0)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

    if not cap.isOpened():
        print('Not Opened Video Camera')
        exit()

    while True:
        ret, src = cap.read()
        dst = cv2.cvtColor(
            src, cv2.COLOR_BGR2HSV_FULL)
        cv2.imshow('Window Name', dst)
        if cv2.waitKey(10) > -1:
            break
    cap.release()
    cv2.destroyAllWindows()
```

多くあるから人気なのか、人気だから増えたのかよく分からないほど、多くのライブラリが提供されています。

その中でも、記事で使っているOpenCVが正式にサポートされていることは、画像処理系を扱う上で大きなメリットになっています。

● 3. C++系の開発環境に比べて非常に容易にアプリケーション開発が可能

画像処理関連でよく使用されるC、C++系の開発言語では、開発作業に入るまでの前準備の段階で既に面倒な手順が多く、開発を始めようと思ってもライブラリの導入設定や実装に必要なリンク・ファイルの調査など、多くの準備が必要です。リスト1とリスト2にPythonとC++で、カメラで取り込んだ映像をHSV変換する例を示します。リスト2のC++では、使用する機能ごとにヘッダ・ファイルをインクルードする必要があり、何をインクルードするかは、ドキュメントで調べる必要があります。また、ビルドの際にはリスト2(b)のような設定ファイルを用意する必要があります。

Pythonでは、ライブラリの導入も簡単にできるようになっていて、ほとんどのライブラリはコマンド一発で導入可能です。さらにAnacondaというパッケージを導入すると、さまざまなライブラリが初めから導入された状態でPythonが使えるになります。また、C言語系のようにリンク・ファイルの設定はなく、基本的にはライブラリ導入後はすぐにコード記述に入れます。

スクリプト言語であるため、記述内容がすぐに動作確認できることも大きなメリットですが、メモリ管理

リスト2 C++でカメラ映像をHSV変換する例

```
#include <iostream>
#include <opencv2/core.hpp>
#include <opencv2/videoio.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>

int main()
{
    cv::Mat src, dst;
    cv::VideoCapture cap;

    cap.open(0);
    cap.set(cv::CAP_PROP_FRAME_WIDTH, 640);
    cap.set(cv::CAP_PROP_FRAME_HEIGHT, 480);

    if (!cap.isOpened()) {
        std::cout << "Not Opened Video Camera" <<
            std::endl;
        return -1;
    }
    while (true) {
        cap >> src;
        cv::cvtColor(src, dst,
            cv::COLOR_BGR2HSV_FULL);
        cv::imshow("Window Name", dst);
        if (cv::waitKey(1) >= 0) break;
    }
    cap.release();
    cv::destroyAllWindows();
    return 0;
}
```

(a) メイン…開発環境はQt Creatorを使用

```
TEMPLATE = app
CONFIG += console c++11
CONFIG -= app_bundle
CONFIG -= qt

SOURCES += \
    main.cpp

win32:CONFIG(release, debug|release): LIBS +=
-L$${PWD}/../../OpenCVSourceCode/opencv440/build/x64/
vc15/lib/ -lopencv_world440
else:win32:CONFIG(debug, debug|release): LIBS +=
-L$${PWD}/../../OpenCVSourceCode/opencv440/build/x64/
vc15/lib/ -lopencv_world440d

INCLUDEPATH += $$PWD/../../OpenCVSourceCode/
opencv440/build/include
DEPENDPATH += $$PWD/../../OpenCVSourceCode/
opencv440/build/include
```

(b) ビルドに必要な設定ファイル…Qt Creatorを使用した場合

などハードウェアに深い知識がなくても、機械学習などのプログラミングが誰でも始めることができる仕様になっていることが最大のメリットだと感じています。

さわだ・ひでひろ

広がる

色変換

図形描画

サイズ

ぼかす

エッジ

膨張

加工

回転

検出

便利フザ

車や人