

ProcessingにおけるopenCVの扱い方

米澤ゼミ

21M7114竹村響

1. はじめに

openCVとは、カメラに写っている様々な物体を検出するための技術で、Processingはじめ、java,pythonなどでもライブラリを使用して使うことができる。

仕組みは、機械学習させたデータ(カスケードファイルと呼ばれている、XMLファイル)を用いる。目を学習させたデータは目を、手を学習させたデータは手を検出する。つまり、プログラムで読み込ませたファイルによって、様々なものを検知することができるのである。ゲームのカセットを交換するみたい！！

2. ライブラリの導入

ツール->ツールを追加->ライブラリ でopenCVと検索。

3. 学習データの読み込み

a. 関数を使用(ファイルやフォルダを意識せずに使えて使いやすい)

`opencv.loadCascade("haarcascade_clock.xml");` //目的の部位の検出を学習させたファイルを読み込む(関数呼び出し)

<http://atduskgreg.github.io/opencv-processing/reference/gab/opencv/OpenCV.html>

b. XMLファイルを指定して読み込み

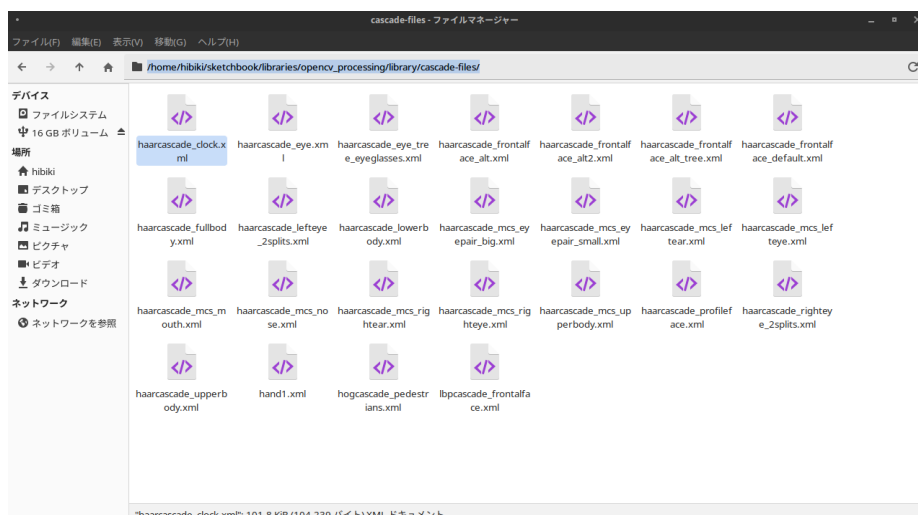
Githubなどに投稿されているXMLファイルを使用する場合はこの手順を使用する必要がある

なお、XMLファイルを保管するフォルダは、

`/home/hibiki/sketchbook/libraries/opencv_processing/library/cascade-files` (ubuntuの場合)

のように、Processingのライブラリが保管されるディレクトリからさらに、`opencv_processing/library/cascade-files`と、下位ディレクトリである。下の画像のように、デフォルトの各ファイルも保管されている。

`opencv.loadCascade("hand1.xml");` //目的の部位の検出を学習させたファイルを読み込む(XML)



4. サンプルプログラム

```
import gab.opencv.*;
import processing.video.*;
import java.awt.*;

boolean onlyfirst = false; //一回目に手を振ったときにtrueになる(初回はtemphandに何も入っていないから、エラーが出るため)
int count = 0; //プログラムを実行する際に、何回手を振る動作を検出したか
int temphand; //1回前に検出した手のx座標を保管する変数

/*座標を保管する*/
Rectangle[] number=new Rectangle[10];

int leng=0; //検出されたデータの要素数

//カメラとopenCvの変数宣言
Capture video;
OpenCV opencv;

//frameをカウントするための変数
int frame = 0;

void setup() {
  size(640, 480);
  video = new Capture(this, width, height, 0);
  openCV_set();
  video.start();
}

void openCV_set(){
  opencv = new OpenCV(this, width, height);
  opencv.loadCascade("hand1.xml"); //目的の部位の検出を学習させたファイルを読み込む(XML)
  //opencv.loadCascade("haarcascade_clock.xml"); //目的の部位の検出を学習させたファイルを読み込む(関数呼び出し)
}

void draw() { //毎秒
  frame++; //フレームが実行される度に加算
```

```

opencv.loadImage(video); //カメラからの入力をopenCVにかける
image(video, 0, 0);
if (frame%10==0)cv(); //何回に一回cv関数を実行するかを決める(ここを調整して何フレームごとに読み取るのかを変更する)

for (int i = 0; i < leng; i++) {
    rect(number[i].x, number[i].y, number[i].width, number[i].height); //hand:座標、
    ここでは、手が検出された所に四角形を描画するが、15フレーム毎に実行とかにすると、
    すぐに上書きされるため、ほとんど見えない
}

if (frame>=60)frame=0; //60回毎にカウントをリセット
}

void captureEvent(Capture c) {
    c.read();
}

void cv() { //openCVの処理を行う関数
    noFill();
    stroke(0, 255, 0);
    strokeWeight(3);

    //opencv.detect()で検出する
    Rectangle[] parts = opencv.detect(); //検出された座標が保管される配列。一つの要素
    に1つの検出された部位の情報が入っている。x座標、y座標、横の長さ、高さが入っている
    はず。いくつも検出されるとその配列も長くなる
    leng = parts.length;
    number = parts; //parts.lengthは映像の中から検出された手の個数(何個も手が検出さ
    れることがある)
}

```

5. 参考サイト

https://r-dimension.xsrv.jp/classes_j/opencv-for-processing/
https://mukai-lab.info/pages/classes/programming_1/chapter11/
https://yoppa.org/sfc_design17/8674.html
<https://qiita.com/hextomino/items/2e8e77d0904c66b02ae9>