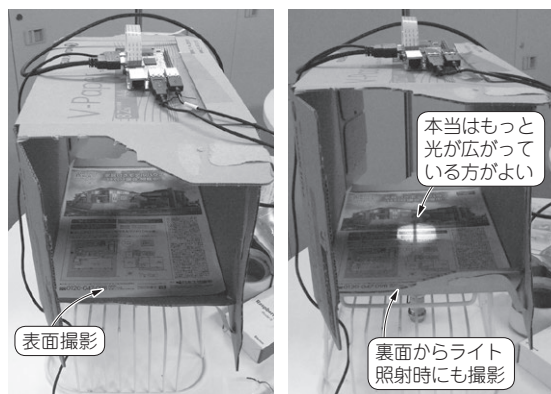


Appendix 3 赤外撮影機能付き

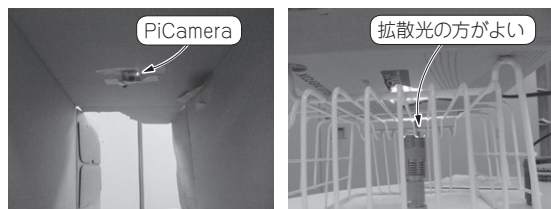
簡易AI判定カメラの自作

ご購入はこちら

佐藤 聖



(a) 全景…チラシの表面を撮影 (b) チラシの光透過具合を撮影



(c) カメラを上部に (d) LEDライトを下部から

写真1 手持ちのラズパイ+専用カメラで作れる簡易的なAI判定カメラ

装置の全体像

私は本特集で、ラズベリー・パイとカメラ、センサーを使って自動的に自動撮影するAI判定カメラを作りました。それだけでなく、自動撮影するほどまでは凝っていない「簡易版の自分用AI判定カメラ」も作りました(写真1, 図1)。学習画像撮影用スタジオとしても使えます。なお、私はラズベリー・パイ専用カメラにRaspberry Pi PiNoir Camera V2(赤外線フィルタ無し版)を、LEDモジュールに赤外線発光タイプを利用しました。可視光を利用するよりも、ハッキリとした特徴量を抽出できると考えたからです。ですが、第4部で紹介している実験は、カメラには通常のPiCameraを、照明には家庭にあるLED電球などを利用できます。

● プログラムの全体像

プログラムのフローは図2の通りです。複数の被写

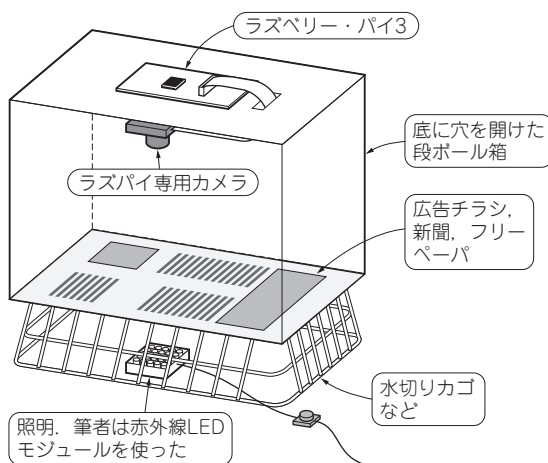


図1 自作のAI判定カメラ

体を撮影する場合には、①～④のプログラムを実行した後、被写体の数だけ⑤～⑥の処理を繰り返し実行します。⑦で⑤～⑥で撮影した画像を確認して⑧で事後処理をします。

この簡易AI判定カメラのPythonプログラムは、筆者提供データ中の体験サンプルDの01 広告チラシの撮影.ipynb, 02 新聞の撮影.ipynb, 03 フリーペーパーの撮影.ipynbになります。いずれも画像撮影処理を行っており体験サンプルAの簡略版です。

01, 02, 03のプログラムの違いは、撮影する画像に付けるファイル名の先頭文字になります。ファイル名は表1に示すルールで作成しており、被写体に対する透過画像と表面画像のそれぞれにA～Fの頭文字を付けて識別できるようにしています。シーケンス番号は被写体ごとに採番します。例えば新聞ならC_0001.jpgがあればD_0001.jpgもpicフォルダ内に作成されます。ここでは01 広告チラシの撮影.ipynb(リスト1)のプログラム内容を解説します。

プログラムの各処理

● 初期設定

▶①ライブラリの読み込み

最初にライブラリを読み込みます。osライブラリ

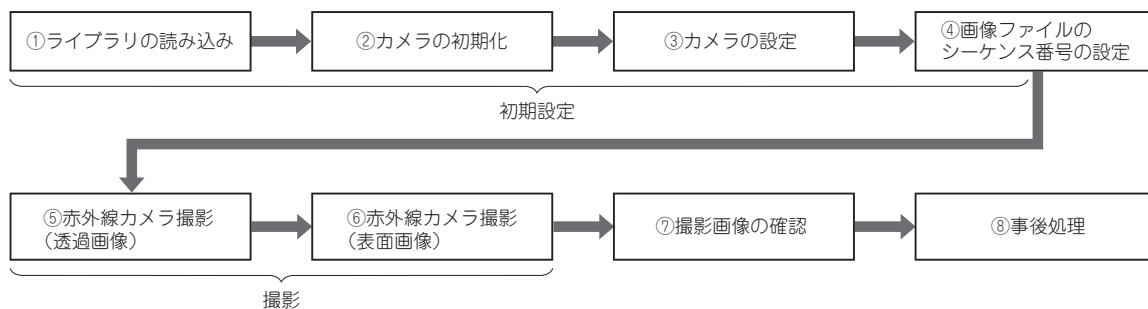


図2 自作した簡易AI判定カメラのプログラム・フロー

とglobライブラリはLinuxファイル・システム上のファイルにアクセスするために利用します。globライブラリはpicフォルダから画像ファイル名リストを取得するのに使います。

ラズベリー・パイのボード上にあるカメラ用コネクタにPiCameraを接続します。PiCameraを制御するためにはpicameraライブラリを使用します。

その他、画像表示用にPIL、Matplotlibと数値演算用にNumPyを読み込みます。最後にPythonライブラリではありませんが、Jupyter Notebook上でインライン表示するために、`%matplotlib inline`を宣言しました。

▶②カメラの初期化

PiCameraは初期化しないと利用できません。あらかじめインスタンス変数cameraを生成して、利用できるようにしておきます。

▶③カメラの設定

PiCameraを設定します。PiCameraのデフォルト画素(1920×1080)です。今回はカメラをセンサとして利用しているので画像の解像度が低くても鮮明でなくても支障がありません。そこで画素が640×640になるようにresolution関数に引き数(640, 640)を設定しました。

brightness関数でデフォルト値と同じ50を設定しています。露出はiso関数に引き数を渡すことで調整でき、デフォルト値は0で露出オートのところを50に固定しました。室内の蛍光灯の明かりでペットボトルが6本入る段ボール箱で試したところ、この値で問題なく撮影できました。もし画像が暗すぎるようでしたら50から値を大きくして調整するとよいでしょう。段ボール箱の形状や容積、利用環境の明るさなどの状態によって最適な設定が異なるかもしれません。

▶④画像ファイルのシーケンス番号の設定

撮影済みの画像ファイル(広告チラシ、新聞、フリーペーパーの透過画像と表面画像)はpicフォルダに格納されます。picフォルダに格納されている画像ファイル名からシーケンス番号を取得し、新たに撮影する際

表1
学習用の画像ファイルは命名規則を決めておく

被写体	ファイル名の先頭文字	
	透過画像	表面画像
広告チラシ	A	B
新聞	C	D
フリーペーパー	E	F

に画像ファイル名に次番号から連番を採番します。

● 撮影

▶⑤赤外線カメラ撮影(透過画像)

被写体を撮影する前に段ボール箱の底に空いた穴から懐中電灯^{注1}などで照らしておきます。ここで一旦、室内の照明を消します。

プログラムはファイル名の設定、カメラ撮影画像の保存、ファイル名の表示です。変数save_picには広告チラシの透過画像に付ける先頭文字Aを設定していますが、02 新聞の撮影.ipynbではC、03 フリーペーパーの撮影.ipynbではEに変えてあります。

▶⑥赤外線カメラ撮影(表面画像)

室内の照明を点けます。部屋の明るさが十分にあれば懐中電灯の明かりを付けたまま表面画像の撮影を行っても問題なく識別できる画像が撮影できるはずですが。

プログラム内容は透過画像と同じで最後にシーケンス番号をカウント・アップしています。変数save_picには広告チラシの透過画像に付ける先頭文字Bですが、02 新聞の撮影.ipynbではD、03 フリーペーパーの撮影.ipynbではFになります。

もし異なる被写体を撮影したい場合には⑤と⑥を繰り返し実行できるので①～④を再実行する必要はありません。

● 後処理

▶⑦撮影画像の確認

撮影済みの画像ファイルをノートブックに表示して確認します。2行目の変数filesではpicフォルダから広告チラシの画像ファイル(表面)の先頭文字Bで始まるファイル・リストを取得します。先ほどと同じように被写体によって02 新聞の撮影.ipynbなら

注1: 筆者はPiCameraのIRフィルタなし版(Raspberry Pi PiNoir CameraModule V2)と赤外線LEDモジュールで撮影しています。