

ステップ3：予測する

ご購入はこちら

佐藤 聖

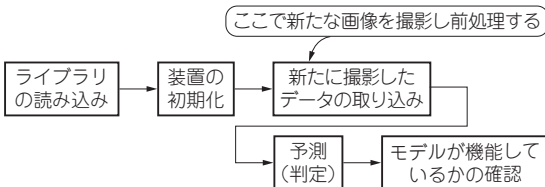


図1 学習済みモデルを使って新たに撮影した対象物を仕分けてみる

ここでは新たに画像を撮影して、学習済みモデルで予測(判定)します。

具体的には、前章で構築した「学習済みモデル」を使って、新しいデータ(新たにポストに投函された郵便物)を予測します。プログラムの処理フローは図1の通りです。新たに投函された郵便物を分類するわけですから、プログラムは01 広告チラシの撮影.ipynbから05 SVMのトレーニングと学習モデル作成.ipynbまでの内容を凝縮したものとなっており、学習済みモデルを構築しないことが大きな違いです。

なお、使っているscikit-learn中のSVM関数の仕様上、2次元配列のテスト・データを用意しないとエラーになってしまい予測が行えません。もし、1つしか被写体を用意したくない場合には、1次元配列をコピーして2次元化するような工夫が必要になります。

準備

● 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(リスト1のln [1])。リレー・シールド、モータ・ドライバ、超音波距離センサの制御用(RPi.GPIO, smbus)、赤外線カメラ用ライブラリ(picamera)、NumPyやPandasはデータ加工用です。画像表示用にPILとMatplotlibを、scikit-learn用にsvmと学習モデルの読み込み用にjoblibを利用します。

● 装置の初期化

続いて装置の初期化を行います。リスト1のln [2]～ln [7]のプログラムは、01 広告チラシの撮影

.ipynb～03 フリーペーパーの撮影.ipynbと同じ内容ですので説明は省略します。

新たに撮影したデータの取り込み

● 画像ファイルのシーケンス番号設定

ここでは画像ファイル名に付けるシーケンス番号を準備します(ln [9])。picフォルダにある画像ファイル・リストから一番大きいシーケンス番号を取得して1を足し、新たに撮影した画像ファイルに使用します。01 広告チラシの撮影.ipynbと同じ処理内容です。

● 画像撮影(透過画像→表面画像)

判定用データとなる画像データを取得します(ln [10])。プログラムは基本的に01 広告チラシの撮影.ipynbと同じ動きをします。ただし、最初に用意した画像と見分けが付くようにファイル名を変えました。被写体(広告チラシ/新聞/フリーペーパー)を撮影する際に表面画像のファイル名はRで始まり、赤外線投光器で赤外線を被写体の裏から透過させて撮影する透過画像はPが先頭文字になります。ここでは被写体の違いによるファイル名の先頭文字が異なることはありません。撮影される画像データはpicフォルダに格納されるのでLinuxのファイル・マネージャなどでもファイルを開けばどんな画像かを確認できます。

15秒以内に被写体を装置に入れ超音波距離センサで検知して表面画像と透過画像を連続撮影できます。もしタイムアウトしてしまったら、このln [10]を再実行すれば連続撮影できます。

ここでは3つの被写体(広告チラシ2枚とフリーペーパー1部)を連続撮影して画像ファイルを作成してpicフォルダに格納しています。被写体とファイル名の関係は表1の通りです。

● 撮影画像の確認と特徴量抽出

ln [11]ではln [10]で撮影した画像を表示して確認します。最初の2組の画像は広告チラシで、最後はフリーペーパーの画像です。フリーペーパーは紙の厚みがあるので透過