

パターン処理工学特論 レポート課題：Skin Detection

出題：酒井智弥，2016/08/05



obama.jpg



obama_pred_label.png^{*1}



obama_pred_skin.png

画素を識別する識別器を機械学習し，肌の領域を検出する．LACS に掲載している skin_detection.zip を展開し，サンプルコード skin_detection.py（仕様は本書の付録を参照）を利用して，**Python コード** と **報告書の PDF ファイル** を提出せよ．ただし，提出物は，それぞれ以下の必要条件を満たすように作成すること．

● Python コード

- ☐ ファイル名は skin_detection_XXXXXXXXX.py（XXXXXXXXX は学生番号）とすること．
- ☐ サンプルコード skin_detection.py が実行できる環境と同じ環境で実行できること．
なお，サンプルコードは numpy, PIL, sklearn, time, os のモジュールを必要とする．
- ☐ サンプルコード skin_detection.py よりも汎化能力が高いこと．
- ☐ サンプルコードと同様に，img_lbl_filenames_test で指定された下記の 4 つの画像ファイルすべての画素を用いて汎化能力を評価し，検出した肌領域を表す画像も保存すること．
"m(01-32)_gr.jpg", "toddler_mollie_aug07_400.jpg", "family4.jpg", "obama.jpg"
- ☐ 学習と評価の合計の実行時間は長くても 3 分程度であること．なお，サンプルコードの実行時間は約 1 分である．
- ☐ img_lbl_filenames_test で指定された画像ファイルに対する F1 score を print で表示すること．
- ☐ 学習と評価には img と lbl フォルダに格納されている画像ファイルのみを使用すること．

ヒント：

- 識別器は何を採用してもよい．自作してもよいが，本課題では，[scikit-learn が提供している識別器](#)の利用を強く推奨する．他のモジュールを使用する場合は報告書に明記せよ．
- 複数の識別器を比較・検討した場合は，その内容を報告書に記載し，最も高い汎化能力を達成したコードを提出する．
- 画素の特徴の表現方法に制限はない．なお，サンプルコードでは，画素を RGB 値の 3 次元特徴で表現している．

^{*1} [W.R. Tan, et al., "A Fusion Approach for Efficient Human Skin Detection", T-II, vol. 8\(1\): pp. 138-147, 2012.](#)

- 報告書

- ☐ ファイル名は `skin_detection_XXXXXXXXX.pdf`（XXXXXXXXX は学生番号）とすること。
- ☐ A4 サイズ 2 ページ相当以上の文量があること。
- ☐ 使用・作成した識別器とその原理を解説すること。
- ☐ サンプルコードからの主な変更点を説明すること。
- ☐ コードを記載する場合は、説明に必須な部分にとどめること。
- ☐ 性能を表す具体的な評価値を示し、サンプルコードよりも汎化能力が高いと言える根拠および汎化能力を高くできた理由を述べること。
- ☐ 誤検出・検出漏れの傾向と原因について考察すること。
- ☐ 止むを得ず必要条件を満たさない場合は、その旨と理由を述べること。
- ☐ 信頼性のある参考文献を引用し、適切に出典を記述すること。

ヒント：

- サンプルコード `skin_detection.py` の内容を報告書の読者が知っていることを前提にしてよい。しかし、提出する **Python** コードを読者が直接見なくても、その内容や動作を理解できるように報告書で解説するべきである。例えば、本書の付録のようにサンプルコードを解説できる。報告書では、特にサンプルコードからの変更点を解説するべきである。

評価・加点項目：

- Python コード

- ☐ 必要条件を適切に満たしていること
- ☐ 汎化能力の高さ
- ☐ 低計算量
- ☐ サンプルコードからの変更内容の単純さ
- ☐ 識別器の性能を十分に引き出す工夫（パラメータの調整、データの使い方等）

- 報告書

- ☐ 必要条件を適切に満たしていること
- ☐ 独自性
- ☐ 機械学習の観点からの考察（学習不足・過学習について等）
- ☐ 複数の識別器の比較・検討
- ☐ 汎化能力以外の性能や識別器の性質に関する考察

提出先：[LACS「パターン処理工学特論」レポート課題](#)

提出期限：9 月 1 日(木) 17:40

問い合わせ先：tsakai@cis.nagasaki-u.ac.jp

付録：サンプルコード `skin_detection.py` の仕様

- 開発環境 [Anaconda](#) でサンプルコードは動作確認済み。Python2.7, Python3.5 の両方可。
- 入力： カラー画像、肌領域を表す 2 値画像。
- 標準出力： 訓練データと評価用データに対する適合率と再現率、および動作状況。
- ファイル出力： 推定した肌領域のカラー画像（ファイル名の末尾は `"pred_skin.png"`），
推定した肌領域の 2 値画像（ファイル名の末尾は `"pred_label.png"`）。

行番号と仕様

- L11-L12： 識別器の学習と評価用に、カラー画像とその肌領域を表す 2 値画像を使用する。これらの画像はそれぞれ `img` と `lbl` に格納されている。
例： カラー画像 `img/obama.jpg` の
肌領域を表す 2 値画像は `lbl/obama.png` である。
- L15-L29： 変数 `img_lbl_filenames_train` は、訓練データとして使用するカラー画像と 2 値画像のファイル名の組（例えば `(obama.jpg, obama.png)`）のリストである。同様に、変数 `img_lbl_filenames_test` は評価用データのファイル名の組のリストである。
- L35-L37： `numpy`, `PIL`, `time` を使用する。
- L39-L45： 関数 `read_image_as_3d_points(f)` は、カラー画像を 3 次元特徴ベクトルの集合として読み込む。ファイル名 `f` のカラー画像の各画素は、画素値 `(R,G,B)` の 3 つの値で特徴表現される。戻り値 `features` は、画素数×3 のサイズの 2 次元配列である。
- L47-L51： 関数 `read_image_as_labels(f)` は、2 値画像をラベルの集合として読み込む。ファイル名 `f` の 2 値画像の各画素は、0（肌領域以外）、1（肌領域）で表される。戻り値 `labels` は、画素数のサイズの 1 次元配列である。
- L53-L62： 関数 `fetch_data(image_label_filenames)` は、
`image_label_filenames` で与えられたすべてのカラー画像と 2 値画像を読み込み、特徴ベクトルの集合を表す 2 次元配列 `features` と、それに対応するラベルの集合を表す 1 次元配列 `labels` を返す。
- L64-L72： 関数 `save_labels_as_image(label_pred, f)` は、推定した肌領域のカラー画像と 2 値画像をファイルに出力する。肌領域の推定結果のラベルの集合 `label_pred` を 2 値画像（黒 0（肌領域以外）、白 255（肌領域））として保存する。この 2 値画像は `f` に `"_pred_label"` を付加したファイル名の PNG 形式であり、保存場所は `lbl` である。画像のサイズはファイル名 `f` のカラー画像と同じとする。また、推定した肌領域を表すカラー画像を保存する。その保存場所は `img` であり、`f` に `"_pred_skin"` を付加したファイル

名の PNG 形式である.

- L78-L79 : 識別器 `clf` は, $\gamma = 10^{-2}$, $C = 1$ の RBF 核関数を用いたサポートベクトルマシンとし, `sklearn` が提供する `svm.SVC` を使用する.
- L85-L87 : 訓練データを `features` と `labels` として読み込む.
- L90-L92 : データ数 (訓練データの全画像の総画素数) が多いので, ランダムに 99% を捨てた訓練データを `features` と `labels` とする.
- L101 : 訓練データ `features` と `labels` で識別器 `clf` を学習する.
- L112-L114 : 訓練誤差を評価し, 適合率と再現率を表示する.
- L116-L117 : 評価用データを `features_test` と `labels_test` として読み込み, 識別器 `clf` で識別する.
- L118 : 汎化誤差を評価し, 適合率と再現率を表示する.
- L128-L136 : 訓練データに対して肌領域を推定し, 結果を画像で保存する.
- L145-L153 : 評価用データに対して肌領域を推定し, 結果を画像で保存する.

ヒント :

- サンプルコードは `img` と `lbl` にある画像ファイルの一部しか訓練データに使用していない. 訓練データを変更したい場合は, L15-L29 の変数 `img_lbl_filenames_train` を書き換える.
- 画素の特徴表現を変更する場合は, L39-L45 の関数 `read_image_as_3d_points` を書き換える.
- `scikit-learn` の識別器を変更したい場合は, L78-L79 を書き換える.
- サンプルコードは読み込んだ画像の一部の画素しか訓練データに使用していない. 使用する画素数を変更したい場合は, L90-L92 で `test_size` の値を変更する. すべての画素を使用したい場合は L90-L92 の処理を省略する.
- 推定された肌領域を表すカラー画像と 2 値画像は, それぞれ `img` と `lbl` に保存される. 性能を測るだけの実験をする場合は, L128-L153 の処理を省略できる.