

マルチメディア情報検索 B1班 中間発表

伊藤広樹 平尾礼央
伊藤光太郎 林田和磨

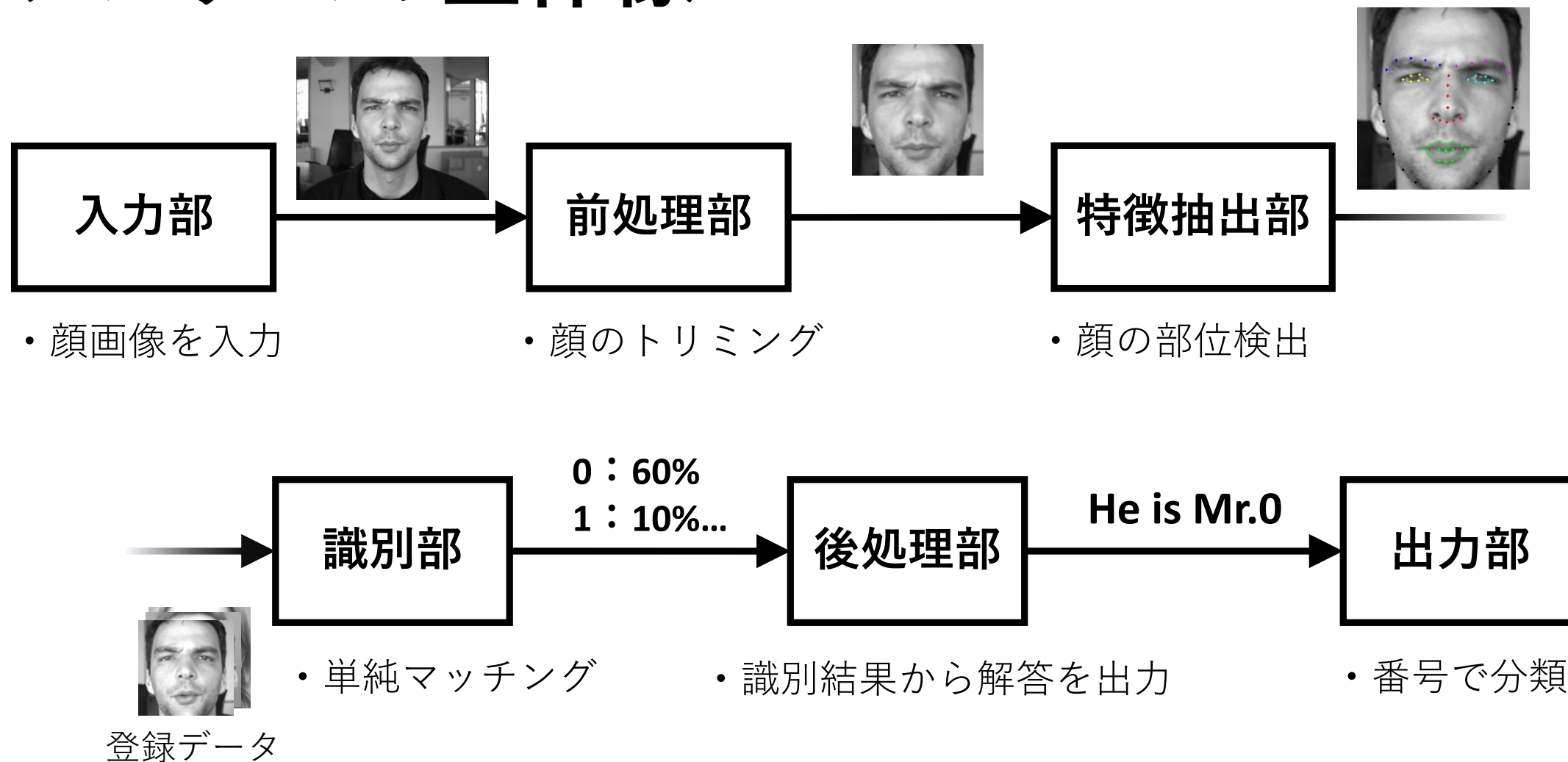
実験目的

「キムタクに一番似ているのは誰か」という疑問や、
「顔をパスワードの代わりとして利用したい」と言った
要求に答えるシステムを開発する。代表的なパターン
識別手法 (K 最近傍法、部分空間法等) を学びながら、
高速で認識率のよいアルゴリズムを作成する。

使用データセット

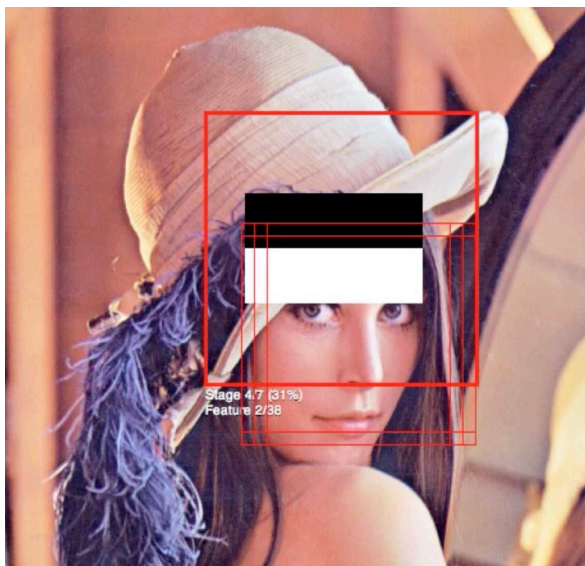
- 与えられたデータは20人の顔写真が10枚ずつ
- 検証用のクエリデータは58枚
 - 人によって枚数は異なる
 - 58枚の内2枚は登録されていない人物
- 認識精度は「 $\text{正解数} \div \text{クエリ数}(58)$ 」で計算

システムの全体像



前処理 顔検出

顔のトリミング



OpenCVで顔を切り出し

- cascade分類器(Haar-Like特徴)
- 検出領域の明暗差により特徴を捉える



Dlibで顔の部位をプロット

- 全体/部位ごとのトリミングも可能
- 点の座標から特徴抽出

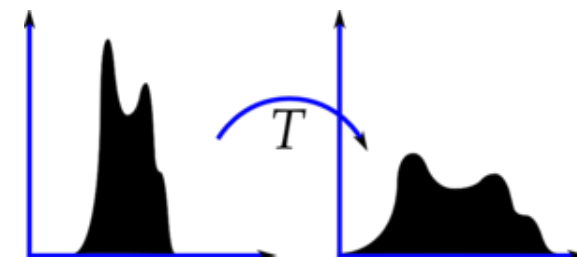
前処理 正規化

画像のサイズを統一



画像を200x200ピクセルへ

画像の輝度を調整



画素値のヒストグラムを平坦化

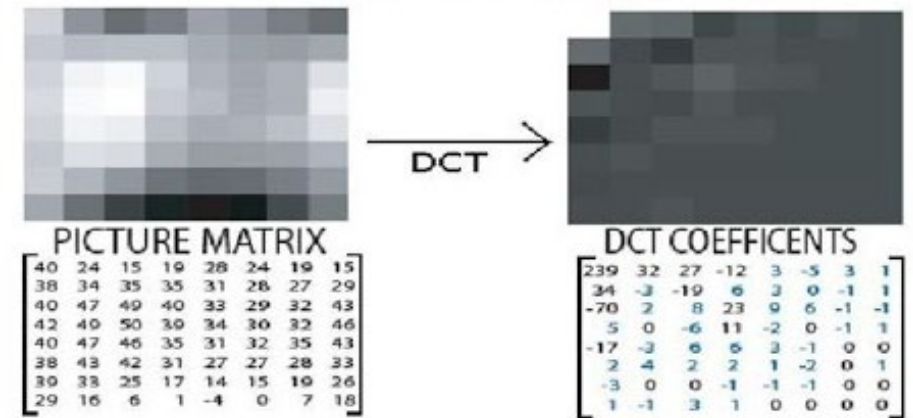
特徴抽出部

Dlibを使った部位情報



口の大きさ、目の大きさ、顔の幅、
各部位間の距離...

2次元離散コサイン変換(DCT)

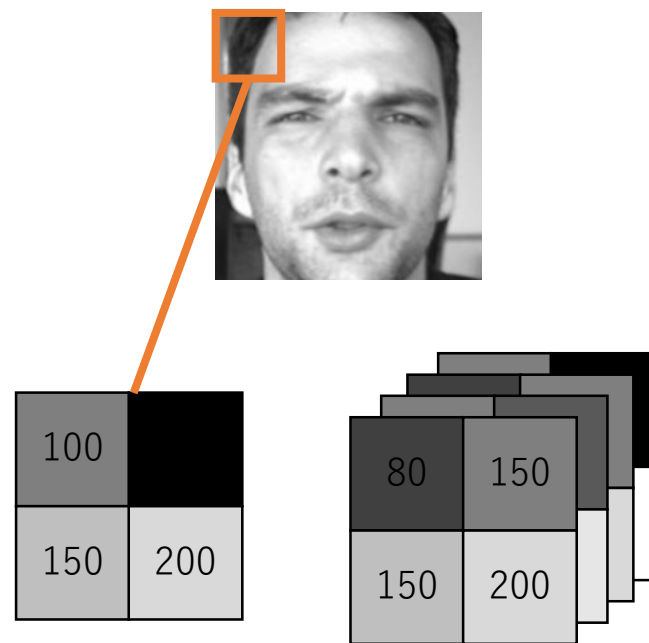


画像を余弦波の周波数と係数に変換

識別部

- 画像そのものを使用
 - **ピクセルマッチング**
 - **NN**
 - **CNN**
- 前処理で出力した特徴量を使用
 - **K最近傍法**
 - **部分空間法**
 - **LightGBM**

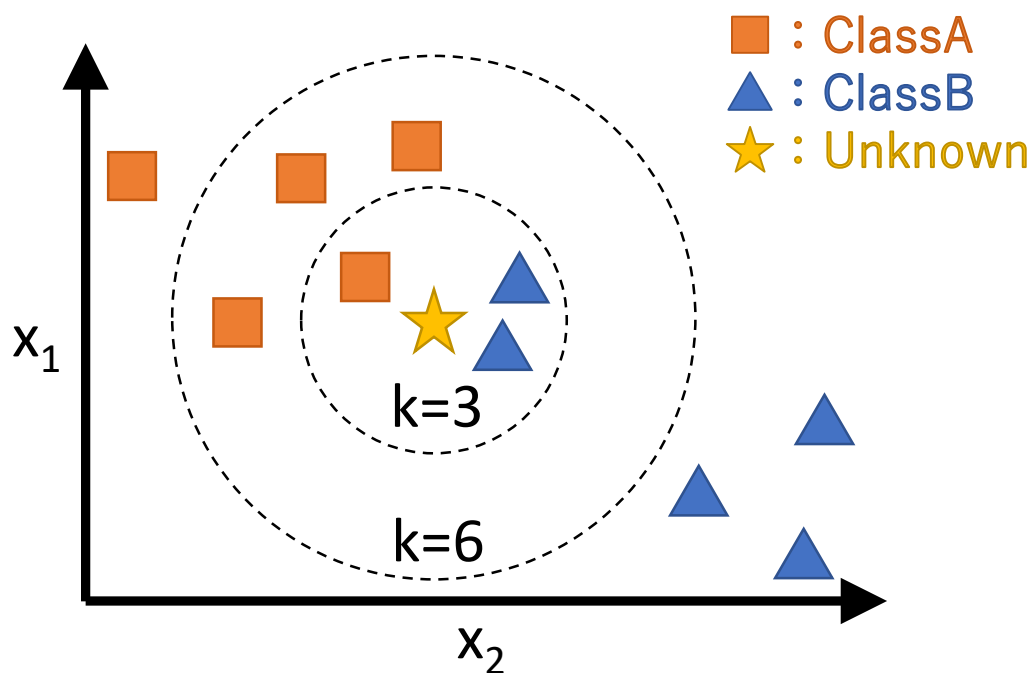
ピクセルマッチング



ピクセルの値を比較
差が最も小さいデータを選ぶ

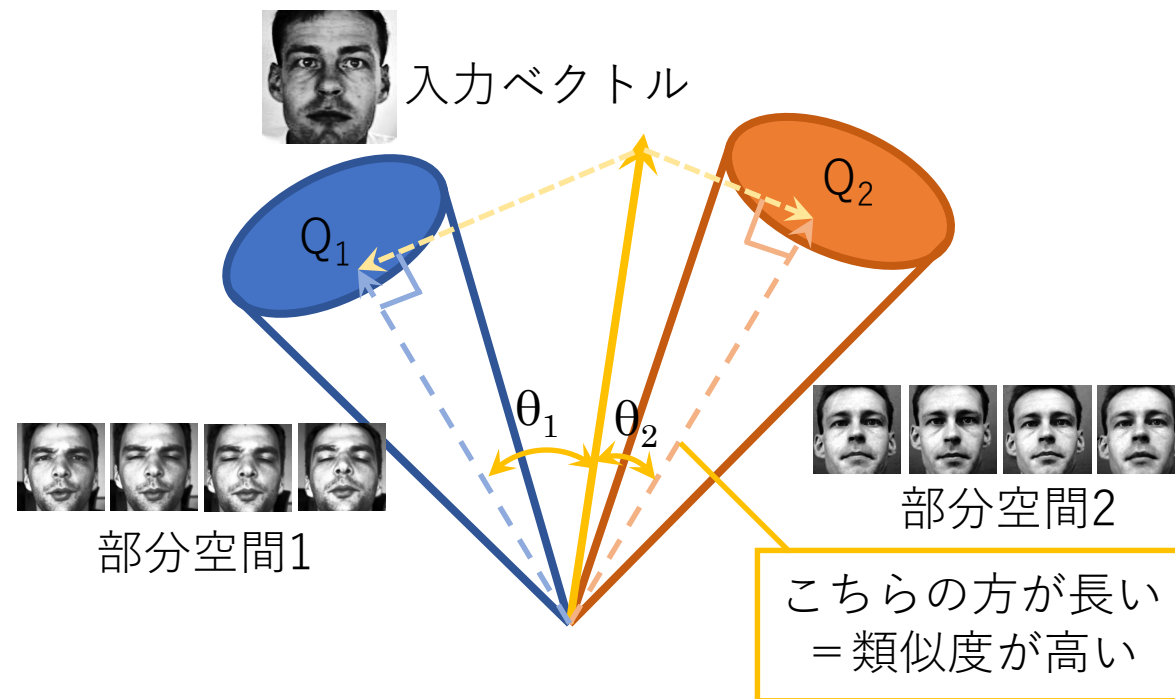
識別部

k最近傍法



特徴量でデータをプロット
特徴が近いk個のデータの多数決

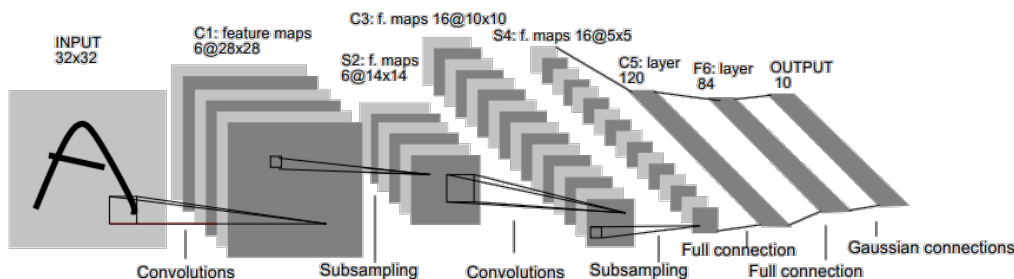
部分空間法



画像をベクトルに変換(KL展開等)
類似度が最も高いクラスに分類

識別部

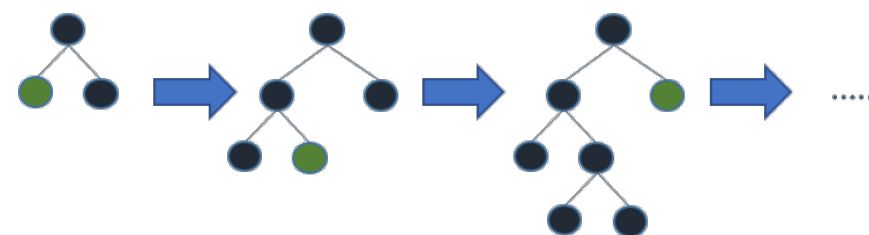
CNN



畳み込みニューラルネットワークの例

- 畳み込み
形が同じものが検出しやすくなる
- プーリング
位置のずれに頑強になる
- 全結合
誤差逆伝播法で重みを学習

LightGBM



Leaf-wise tree growth

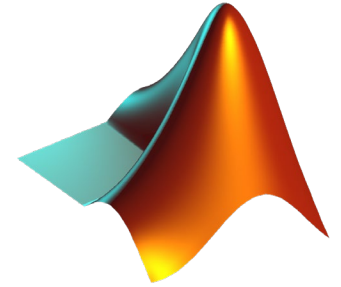
LightGBMの基本アルゴリズム

MicrosoftのOSS

複数の決定木で学習させる手法
特徴量の重要度を出力できる

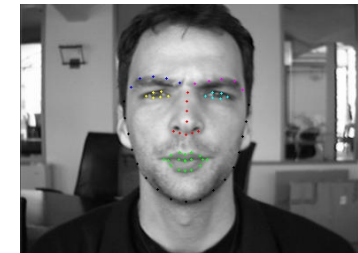
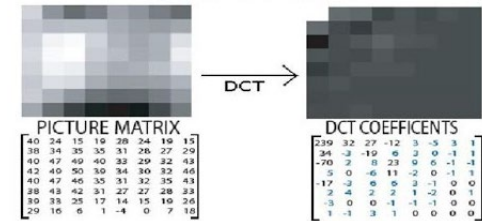
班の方針

- Matlabと画像処理系ライブラリが豊富なPythonを使用
 - Matlab担当(伊藤 光太郎・林田和磨)
 - Python担当(伊藤 広樹・平尾 礼央)
- 基本的に**外部データを使用せず**に精度の向上を目指す
- また、**意図的に検証データに寄せる(Leakage)行為**を禁止
- 最終的にはGUIを作成し、操作できるようにする



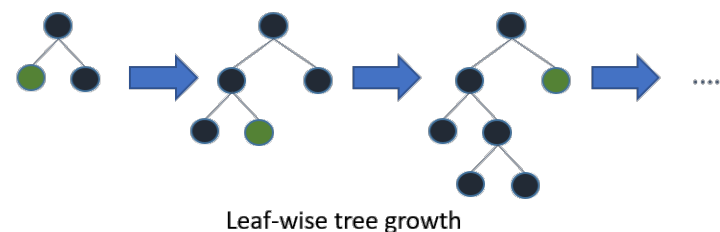
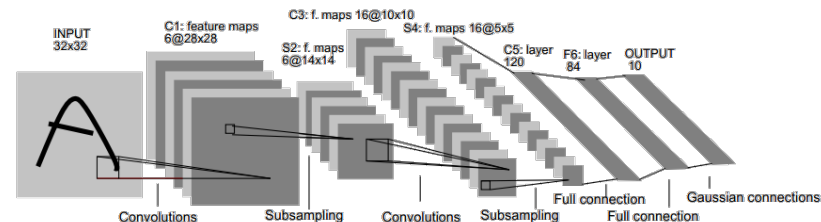
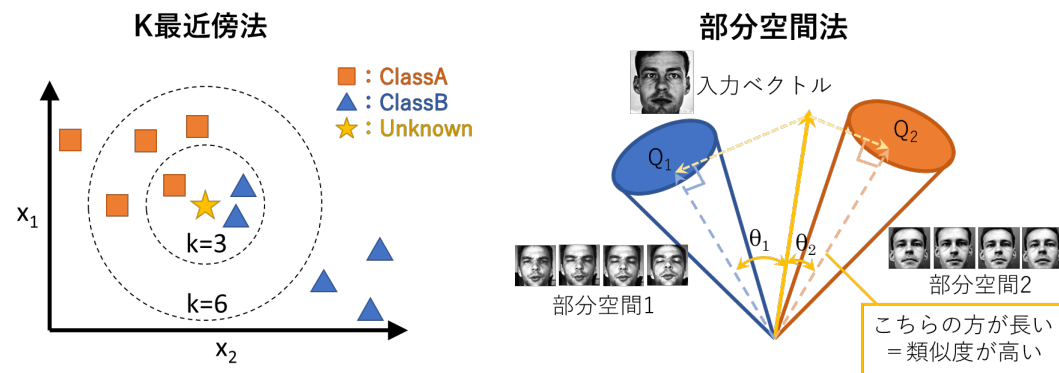
進捗 前処理と特徴抽出

- 顔のトリミング、正規化を実装(matlab,python)
- 学習データの増し(python)
- 離散コサイン変換(DCT)で特徴抽出(matlab,python)
- 画像のヒストグラムを取得(matlab)
- Dlibを用いた顔の部位検出(python)
- 顔の部位から各部位の長さ等の特徴抽出(python)



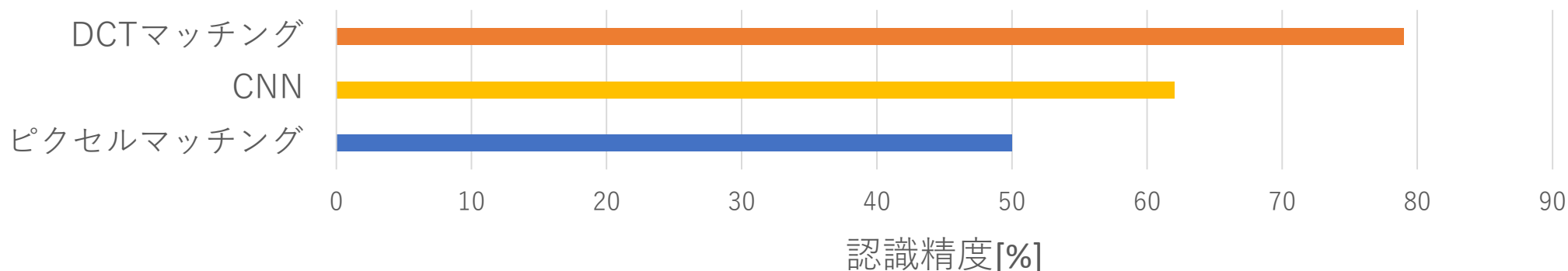
進捗 識別部とGUI等

- 単純マッチング(matlab,python)
- K最近傍法(matlab)
- 部分空間法(matlab)
- CNN(python)
- LightGBM(python)
- GUI作成(python)
- データの分析(python)



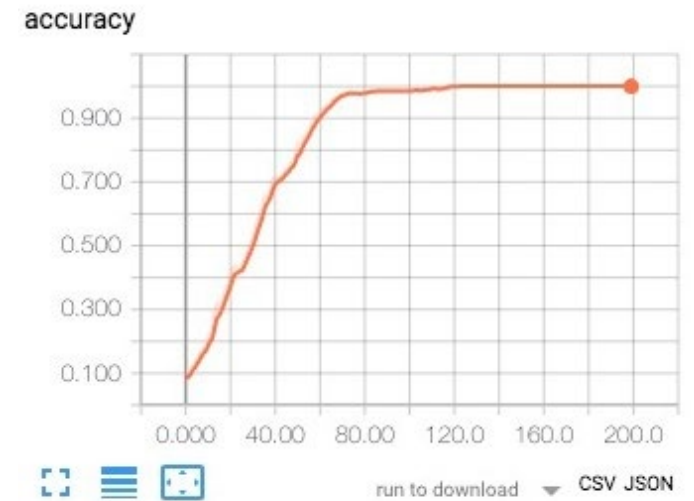
認識精度が高かった手法

- DCTマッチング **79%**
 - エッジ検出後のDCTの特徴量を比較し、類似度が高いものに分類する
- CNN **62%**
 - パラメータを変更した10種類の画像に水増しし、CNNを実行した
- ピクセルマッチング **50%**
 - 正規化した画像のピクセルを比較し、類似度が高いものに分類する

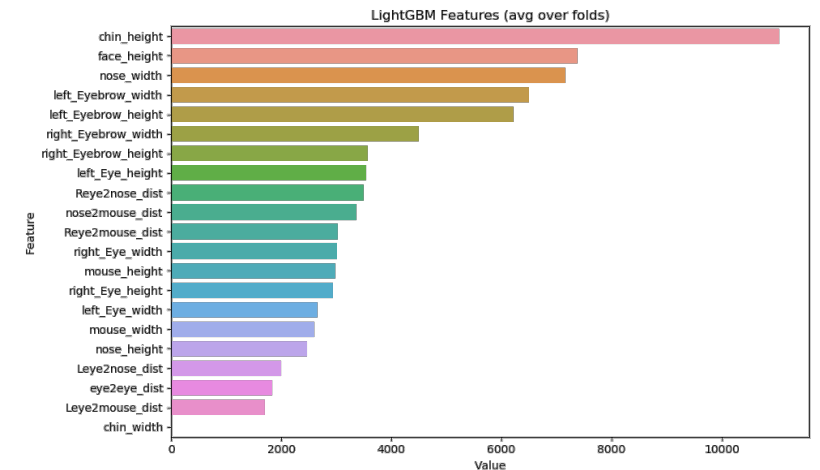


考察

- CNNを実行すると、学習データに最適化はされたが、検証データでは60%程度の精度しか出ない
 - 検証データの40%程度は間違えているので、**過学習**してしまっている
- LightGBMで顔の部位を用いた特徴の重要度を出力した
 - 顔の長さ、鼻の幅、眉毛の幅及び高さのような順で重要度が高いことがわかった



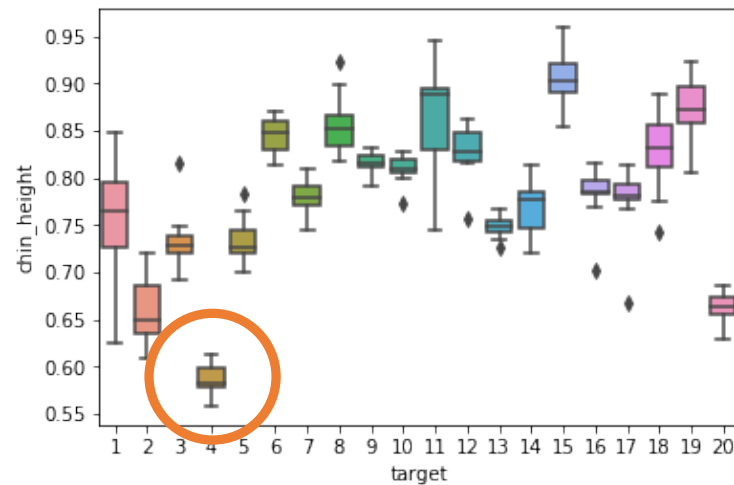
trainデータの精度



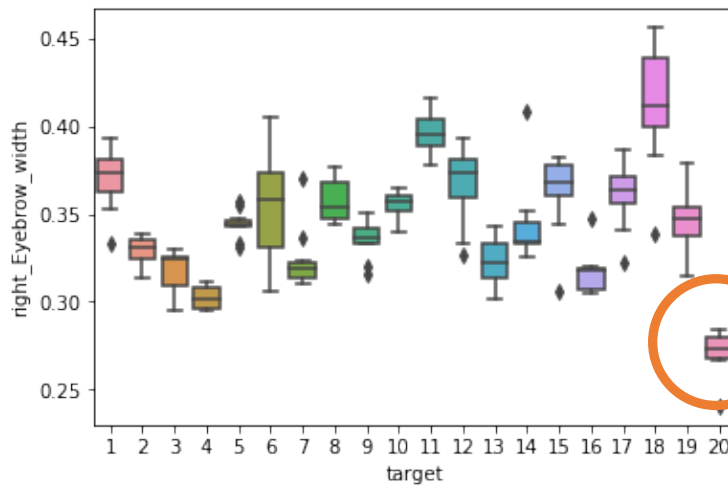
特徴量の重要度

考察

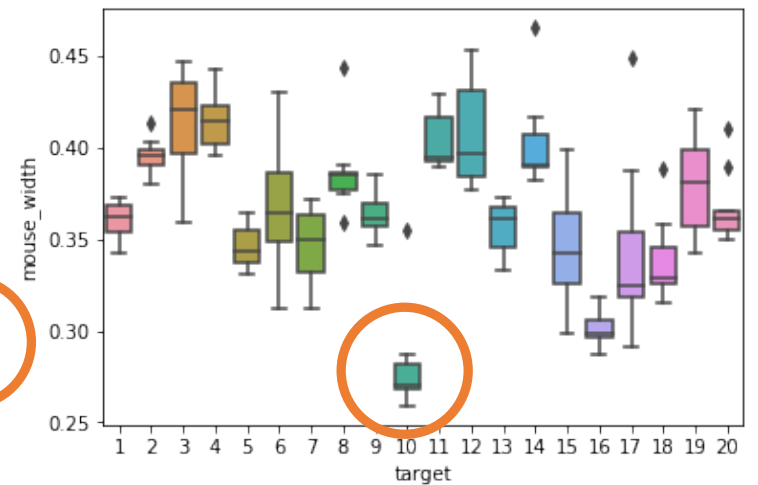
- 顔の各部位の特徴量を箱ひげ図で観察した結果



顔の長さ



右眉の幅

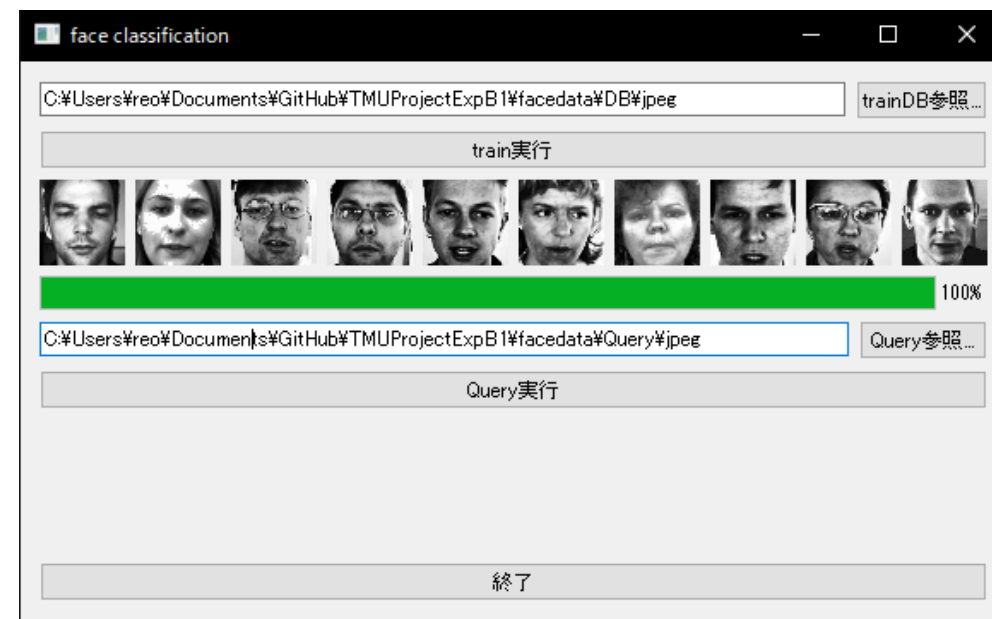


口の幅

- 少なくとも上記の3人はこの項目を見れば分類できると考えられる
- このように固有のデータをもっと特徴量として抽出する必要がある

GUI(PyQt5を使用)

- データベースとクエリをGUIで操作
- 認識精度が最も良いアルゴリズムを適用
- 追加機能
 - カメラで顔登録する機能
 - 顔の読み取りでロック解除など
 - GUI改善



試作GUI

今後のスケジュール

- 有効な特徴量の探索
- 更なる精度の向上(80%以上を目標とする)
- 未登録の場合未登録だと判定する機能
- カメラを利用した顔認識システムの構築
- GUI改良

ご静聴ありがとうございました