

12 CNN による顔の彫りの深さを判定するアプリの作成

新里匠美

指導教員 ソソラ

1. はじめに

日本人は立体的で濃い顔立ちの縄文人と平坦で薄い顔立ちの弥生系渡来人の 2 種類の祖先が存在している。この 2 種類の祖先によって、日本人の顔は大きく 2 種類に分けることができる。顔の彫りが深い縄文顔と平坦な顔立ちの弥生顔である。本研究では、縄文顔か弥生顔か判断するアプリケーションを作成する。



図 1 縄文人(左)・弥生人(右)のイメージ[1]

2. 研究概要

2.1 概要・目的

本研究では、深層学習による画像判定について学習を行い、応用例として縄文顔か弥生顔かを判断するアプリケーションを作成する。

開発環境は以下に示すとおりである。

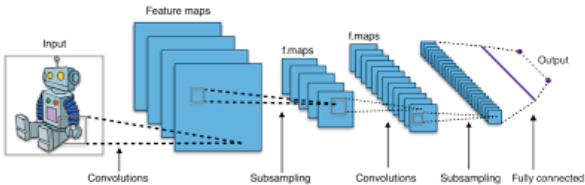
表 1 開発環境

OS	Windows10
使用言語	Python3
使用 IDE	Spyder, JupyterLab
使用ライブラリ	OpenCV, NumPy, Keras, Matplotlib

2.2 畳み込みニューラルネットワーク(CNN)

畳み込みニューラルネットワーク(Convolutional Neural Network :CNN)とは、画像の

分類に適したネットワークである[2]。畳み込み層とプーリング層を交互に重ね、最後に全結合層で特徴を割り出すことで空間的な特徴を抑えることができる。畳み込み層では入力されたデータにフィルタをかけることによって特徴を割り出して特徴マップの作成を行う。プーリング層では特徴マップを特定の領域の代表値を抽出することによって特徴マップの汎化性能を上げる。



1. 図 2 CNN のイメージ[2]

2.3 学習モデル作成の手順

アプリケーションの実装に伴い、教師あり学習によって顔の濃さを判定するモデルの作成を行う。モデル作成の手順は以下の通りである[2.3].

ステップ 1 :

ラベル付き訓練データセットを作成

- ・ Web 上から画像データを入手する
- ・ 画像データの加工を行う

ステップ 2 :

CNN 学習モデルの作成

ステップ 3 :

テストデータで評価する

これらの動作を繰り返すことによってモデルの判定精度を上げていく。

3. ラベル付き訓練データセットの作成

収集した画像から OpenCV によって顔の部分抽出し、データセットを作るプログラムを作成した。このプログラムは以下の手順からなる。

- ① OpenCV で顔検出, 検出部分の切り取り
- ② 切り取り部分のリサイズ
- ③ リサイズした画像を 24bitRGB 形式に変換
- ④ 変換したデータを NumPy 配列方式に変換

4. モデルの作成・評価

参考書[3]の画像判定のプログラムに基づいて, CNN 学習モデルを作成するプログラムを実装した. 使用したコンピュータのスペックは CPU が Core i7, 3.4GHz, 主記憶が 16GB である. データセットとしての 2 種類 (縄文顔, 弥生顔) のデータそれぞれ 50 枚 (64×64), 計 100 枚画像データと, その画像データを回転処理によってデータ拡張を行った 2000 枚の画像データを学習データとテストデータに分け学習を行った. 図 4 に学習回数による精度の変化の様子を示す. ここで, X 軸は学習回数, Y 軸は精度を表している. さらに, 学習データの精度を青線, テストデータの精度をオレンジ色にした. 最善の場合は分類器の精度は学習データに対して 98%, テストデータに対して 76% である.

```
Epoch 1/10
14779/14779 [=====] -
184s 12ms/step - loss: 0.6262 - acc: 0.6885 -
val_loss: 0.6288 - val_acc: 0.7066
Epoch 2/10
14779/14779 [=====] -
184s 12ms/step - loss: 0.4159 - acc: 0.8092 -
val_loss: 0.5358 - val_acc: 0.7894
Epoch 3/10
14779/14779 [=====] -
184s 12ms/step - loss: 0.2797 - acc: 0.8840 -
val_loss: 0.5992 - val_acc: 0.7967
```

図 3 学習の様子

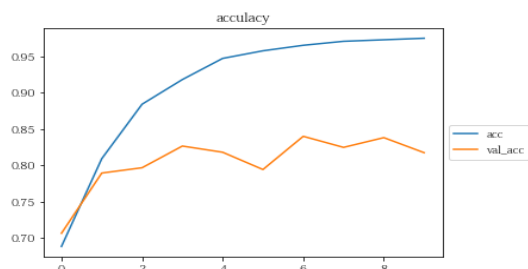


図 4 精度の変動の様子

また, 精度改善のためデータセットに使用する画像の枚数を動作確認時の約 15 倍(元データ各 750 枚 (64×64), 拡張データ 30000 枚)に増加した.

その結果, テストデータに対しての精度は 81% まで改善した. 学習データの増加によって精度が上がることを確認できた.

5. 判定アプリケーションの作成

作成した CNN 学習モデルを使い, コマンドラインで顔写真の画像データを入力すると, 縄文顔か弥生顔かを判断するアプリケーションを作成した. 判定結果は図 5 に示した通りに表示される. このアプリのプログラムは以下の手順からなる.

- ① 入力された画像から顔検出
- ② 検出部分を加工, NumPy 配列に変換
- ③ データを予測
- ④ 予測結果を表示

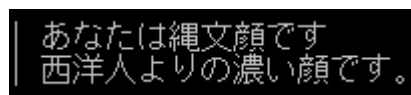


図 5 結果画面

6. まとめ

深層学習の学習しながら, 画像データによる学習モデルを作成し, 縄文顔か弥生顔か判断するアプリケーションを作ることができた. また, 画像データの増量によって精度の改善ができることが確認できた. 本研究で学んだことを今後も活かしていきたい.

今後の課題としては GUI の作成とスクレイピングに関する研究などがあげられる.

7. 参考文献

- [1] 縄文顔と弥生顔-国立科学博物館
<https://www.kahaku.go.jp/special/past/kao-ten/kao/jomon/jomon-f.html>
- [2] Antonio Gulli, Sujit Pal, 直感
 DeepLearning-Python×Keras でアイデアを形にするレシピ, 株式会社オライリー・ジャパン, 2018, p71-76.
- [3] クジラ飛行機, Python によるスクレイピング & 機械学習開発テクニック, ソシム株式会社, 2018, p348.