

はじめに

文字認識とは
文字認識は、紙の書類、PDF、画像に書かれた文字を読み取り、デジタルデータに変換する技術である。

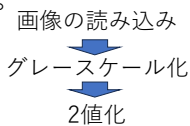
研究目的

Googleをはじめ、多くの企業が文字認識技術を開発しているが、そのほとんどは英語用に作られており、はっきりしていない日本語の文字は正しく認識されないことがある。日本語用の文字認識を作ること、精度の向上を目指す。また個人個人の字にあった文字認識を作成することができないかを実験する。

研究手法

PythonのOpenCVというライブラリを主に使う。

処理の流れ

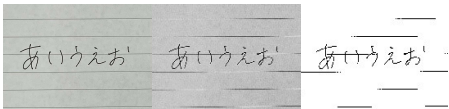


実験

①文字を読み取りやすくする

目的：画像から罫線を取り除き、読み取りやすくする

結果：左から、元画像、グレースケール画像、結果



考察

閾値が適切でないために罫線も残ってしまっている
→閾値を変更

```

import cv2
1 img=cv2.imread('meji.jpg')
2 gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
3 ret,thresh=cv2.threshold(gray,127,255,cv2.THRESH_BINARY)
4 cv2.imshow('meji.jpg',img)
5 cv2.waitKey(0)
6 cv2.destroyAllWindows()
7
8 cv2.imshow('meji.jpg',gray)
9 cv2.waitKey(0)
10 cv2.destroyAllWindows()
11
12 cv2.imshow('meji.jpg',thresh)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
15
16 cv2.imshow('meji.jpg',img)
17 cv2.waitKey(0)
18 cv2.destroyAllWindows()
  
```

127から60に変更

あいうえお ←数値を変更した結果、きれいに罫線が消えた

考察

文字認識において、プログラムが読み取りやすい形にするには文字や罫線の濃さによって数値を変更する必要がある。

②公開されている分類器（顔認識用）を使う

目的

OpenCVをインストールしたときに一緒にインストールされる顔の分類器(harcascade_frontalface_alt.xml)を使って認識の練習をする。

結果 ↓実験②の結果左から、元画像、グレースケール画像、二値画像、結果



実験②の結果 左から、元画像を斜めにした画像、結果↓

→顔を認識することはできたが傾いている場合はどうなるのか

実験②”
下の画像を使ってどの顔が認識されるか実験する



考察

正面の顔を学習しているため、顔が傾いていると認識できない。
→文字認識についても同様のことが起こるのではないかと

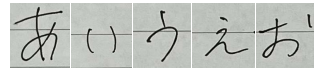
この研究で使用している画像はすべて著者（横川勇気）が撮影しています

③実際に文字認識を行うプログラムの作成

ニューラルネットワークによって、トレーニングモデルを作成するプログラム(1)と生成されたモデルを使って実際に文字認識を行うプログラム(2)の二つを作成し、あいうえおの5字を読み取る。

トレーニングモデルにはETL4を利用する。

使用した画像→



ETLとは

「ETL文字データベース」は手書きまたは印刷の英数字、記号、ひらがな、カタカナ、教育漢字、JIS第1水準漢字など、約120万の文字画像データを収集しています。1973年から1984年にかけて電子技術総合研究所(現 独立行政法人産業技術総合研究所)において、日本電子工業振興協会(現 電子情報技術産業協会)・大学・民間の研究機関の協力のもとに文字認識研究用に収集され、ETL-1からETL-9の9種類のデータにまとめられています。

<http://etlcnb.db.aist.go.jp/?lang=ja> より引用

↓トレーニングモデルを作成するためのプログラム(1)

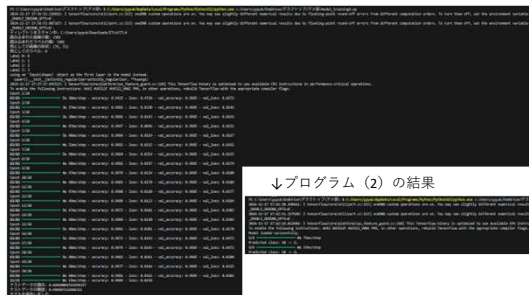
作成したトレーニングモデルを利用するためのプログラム(2)



結果

(1) データの損失率が非常に高く、精度が低いことが分かる
(2) すべての文字が「あ」と認識されてしまい失敗した。

↓プログラム(1)の結果

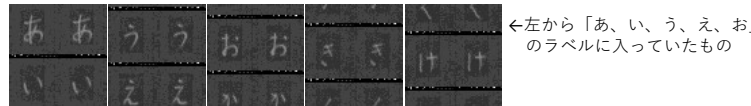


↓プログラム(2)の結果

考察

データセットの読み込み方(画像の切り分け方やラベルの設定の仕方)が間違っているのではないかと

③'読み込んだデータを画像化して表示する



←左から「あ、い、う、え、お」のラベルに入っていたもの

結果

データセットの読み込みに問題があった

今後の展望

データセットの読み込み方を見直し、プログラムが正しく動作するようにする。

データセットの量を増やし、精度の向上を目指す。

参考文献

勝山裕(2015-01-20)“統計的性質に基づく文字の高精度認識に関する研究”.
<https://tohoku.repo.nii.ac.jp/records/71910>

手書きひらがなの認識で99.78%の精度をディープラーニングで
<https://qiita.com/yukoba/items/7a687e44395783eb32b1>

Tensorflowを2ヶ月触ったので“手書きひらがな”の識別95.04%で畳み込みニューラルネットワークをわかりやすく解説
<https://qiita.com/tawago/items/931bea2ff6d56e32d693>