

RaspberryPi を用いたくずし字認識組み込みシステムの開発 Development of Embedded System for Recognizing *Kuzushiji* by Deep Learning

竹内 正広[†] 早坂 太一[†] 大野 亙[†] 加藤 弓枝[‡] 山本 和明[§]
Takeuchi Masahiro Hayasaka Taichi Ohno Wataru Kato Yumie Yamamoto Kazuaki

1. はじめに

近年、国文学研究資料館により歴史的曲籍画像がデータ化されるなど、歴史的資料に対する動きが活発である。期待されることとして、画像データ化による文化財の保護や、このなかには、さまざまな分野の書物が含まれており、地震や噴火などの過去の記録をもとにした自然科学の研究の発展など、異分野を融合させた研究の展開がある。これらの資料には、くずし字とよばれる、江戸時代以前に使われていた変体仮名と漢字をくずして書いた文字が使われている。これを読むには、専門の知識が必要となる。そこで、先行研究で、近年注目を集めるディープラーニングと呼ばれる機械学習の手法を用いたくずし字判別モデルが構築され、精度は、変体仮名が 95.0%、漢字などが 67.7%の正解率 91.9%[1]で判別可能である。しかし、くずし字の判別には、構築されたモデルをパソコンで使う方法やスマートフォンなどから WWW アプリケーションで行う方法などがあるが、パソコンのスペースやネットワーク環境などの制約があり、例えば小中学校の授業中では行うことが難しい。

本研究では、小型で比較的安価であり、教育機関での活用が期待されているシングルボードコンピュータ RaspberryPi を用いて、構築されたモデルによる、くずし字の認識組み込みシステムの開発を行う。

2. 開発したシステムの概要

システム構成図を図 1 に示す。使用する言語は、Python3 を用いて、RaspberryPi 用のカメラモジュール PicameraV2 から対象のくずし字を撮影する。本研究で用いる、Deep Learning のモデルは、くずし字 1 文字に対して認識を行うため、OpenCV を用いて、認識する 1 文字の選択、画像の変換などを行う。この画像に対して、Tensorflow[2]と呼ばれる、PC などに加えて、RaspberryPi にも提供している、機械学習用ライブラリを用いて認識を行う。Tensorflow では、構築された認識モデルを読み込み、認識し、結果を 7 インチディスプレイに表示する。

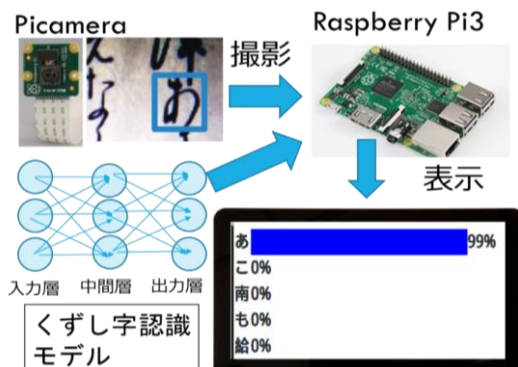


図 1 システムの概略図

[†] 豊田工業高等専門学校 [‡] 鶴見大学
[§] 国文学研究資料館

3. システムの詳細

3.1 Opencv による画像処理

認識を行う画像に対する、前処理を OpenCV によって行う。文字の選択は、図 2 に示すようにタッチスクリーンに指を触れてから離すまでの x-y 座標を取得し、図 2(a)に示すように選択領域を囲った後、図 2(b)に示すようにトリミングを行う。グラフを構築する際の学習に用いた画像が、62×62 ピクセルの 1 チャンネルであるため、切り出した画像を 62×62 にリサイズする。その後、グレースケールを行い、2 値化を行ってチャンネル数を 1 にして、テストデータと同じ形式にする。図 2(c)に画像処理が完了した画像を示す。

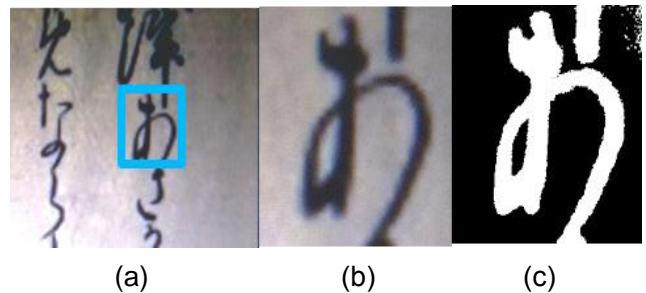


図 2 OpenCV による画像処理の例

3.2 認識に用いるモデル

本研究で用いる、くずし字を認識するモデルは、先行研究[1]を元にしたものを用いる。モデルは、畳み込みニューラルネットワーク(CNN)と呼ばれる手法を用いたものである。ネットワークの構造は、5 つの畳み込み層と 2 つの全結合層によるもので、この CNN モデルに対して 200,000 回学習を行った。学習には、3,985 種類の 418,831 文字 (平均 105 文字/種類) のデータを用いた。そのうち、変体仮名(48 文字+濁点・半濁点)は、75 種類の 262,547 文字 (平均 3,501 文字/種類) であり、その他の漢字、カタカナ、記号などは、3,910 種類の 156,284 文字 (平均 40 文字/種類) である。

精度は、テストデータの源氏物語絵巻「桐壺」11,286 文字 (変体仮名 10,026 文字、その他 1,260 文字) に対して正解率 91.9%(変体仮名 95.0%,その他 67.7%)である[1]。

3.3 構築されたモデルの処理

本研究で使用する構築済みモデルは Caffe と呼ばれるライブラリを用いて作成されたが、RaspberryPi で実行させるために Tensorflow を用いる。Caffe のモデルはネットワーク構造を示したファイル(prototxt)と重みファイル(caffemodel)に分かれている。このままでは、Tensorflow では実行できないので、Caffe models in Tensorflow[3]で、prototxt を .py に、caffemodel を .npy 形式のファイルに変換を行い、Tensorflow

で実行できる形にする。その後、ProtocolBuffer 形式にグラフを変換して、保存を行い、次回以降は、このグラフを読み込むだけで、くずし字の認識を行うようにする。これによって、Tensorflow を用いて、RaspberryPi 上で実行することができる。

4. 開発したシステムの成果

本研究で開発したアプリケーションによる認識結果の表示画面を図 3 および図 4 に示す。まず、タッチパネルの操作で RaspberryPi を動かすため、シェルスクリプトをディスプレイに対して 2 回タッチを行い、python で作成した本プログラムを実行する。その後、wxPython を用いた GUI アプリケーションが起動する。そこから、撮影、切り取り、認識、履歴のそれぞれのボタンをクリックすることで処理を行う。



図 3 認識結果の表示例

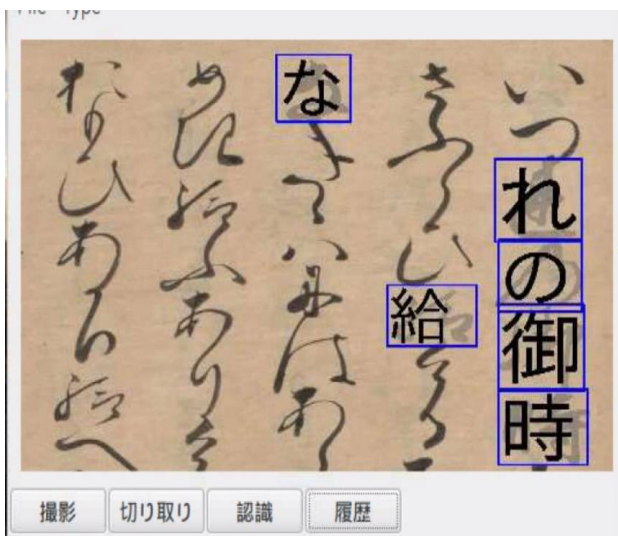


図 4 複数回認識を行った結果

撮影は、ディスプレイの背面についている、PicameraV2 (固定フォーカス約 30cm) により 3 秒間プレビューを行った後撮影し、画像を表示する。

切り取りは、認識したい 1 文字に対して、ディスプレイをタッチ、ドラッグして選択を行う。

認識は、1 文字を約 3 秒で認識を行い、図 3 に示すような、結果の表示の処理を行う。保存をすることで結果の保存と元の画像に対して認識したもじの置き換えを行う。図 4 が複数回文字を認識した結果である。

履歴は、保存されている座標の領域を認識対象の画像に対して、図 4 に示すように、矩形で囲って表示する。囲った領域内をタッチすることで、過去の結果を表示する。

5. おわりに

RaspberryPi によって、カメラで撮影した、1 文字に対するくずし字の認識を行えた。1 文字を認識するのにかかる時間は約 3 秒であり、WWW アプリケーションによる、サーバー側での認識時間約 0.4 秒[1]と比べると、高性能なパソコンを用いなくても、RaspberryPi のような小型ボードでも問題なく、くずし字の認識ができると考えられる。

しかし、認識は約 3 秒で行えるが、アプリケーションを立ち上げるのにかかる時間が約 90 秒かかるのが問題である。これは、TensorFlow を Python で実行する場合、構築済みモデルの読み込みにかかる時間に加え、実行を行う初回の数回は、読み込んだモデルに対する調整に時間がかかるため[4]である。そのため、起動前の処理に、モデルの読み込み(約 15 秒)と、あらかじめダミーのデータで処理を行い、モデルに対する調整(約 65 秒)をし、使用者が認識を行うときには約 3 秒で認識を行えるようにするためである。他にも、使用するモジュールのインポートに数秒かかる。

対策として、Python は、インタプリタ方式であり、静的な型宣言を行わない言語で、常に動的にオブジェクトの演算を行うため、認識を行う処理を C++ で静的にコンパイルして実行することによる高速化の方法[5]が考えられる。

今後の課題は、上記の方法による起動の高速化、エラー処理、使いやすさの向上が考えられる。

謝辞

本研究は、JSPS 科研費 JP16K02433 の助成、平成 28 年度内藤科学技術振興財団研究助成、および TRC-ADEAC 株式会社からの助成を受けたものです。

参考文献

- [1] 早坂太一, 大野互, 加藤弓枝, 山本和明, “深層学習による変体仮名翻刻アプリケーション開発の試み,” 人工知能学会全国大会論文集, 3Q1-2in1, 2017/5
- [2] 小池 誠, “キュウリ自動選別コンピュータ,” インターフェース, 第 43 巻, 第 3 号, pp.23-51, 2017/3
- [3] Caffe models in TensorFlow, <https://github.com/ethereon/caffe-tensorflow>, 2017 年 6 月 21 日閲覧
- [4] TensorFlow for Raspberry Pi, <https://github.com/samjabrahams/tensorflow-on-raspberry-pi>, 2017 年 12 月 2 日閲覧
- [5] Python で学習したネットワークを C++ で実行する http://memo.saitodev.com/home/tensorflow/use_graph_in_cxx, 2018 年 1 月 10 日閲覧