くずし字認識の進化とサービス化の展開

カラーヌワット・タリン、北本朝展

（ROIS-DS 人文学オープンデータ共同利用センター（CODH）、国立情報学研究所）

人文学オープンデータ共同利用センター（CODH）と国文学研究資料館が 2016 年に 日本古典籍くずし字データセットを公開した後、くずし字認識研究は大きく進展した。そして 2019 年の Kaggle くずし字認識コンペティションを経て、CODH は KuroNet くずし字認識サービスを公開した。本論文はこうした流れを振り返り、くずし字認識と物体検出アルゴリズムの関係、KuroNet の進化、くずし字認識のサービス化、Kaggle コンペの教訓、くずし字認識スマホアプリの開発、くずし字認識研究の課題とデータセットの拡大など、くずし字研究の多岐にわたる展開をまとめる。

The Evolution of Kuzushiji Recognition Research and Services

Tarin Clanuwat / Asanobu Kitamoto

(ROIS-DS Center for Open Data in the Humanities, National Institute of Informatics)

Kuzushiji recognition research has been progressing recently after the release of Kuzushiji dataset from the Center for Open Data in the Humanities (CODH) in 2016. In 2019, CODH also released KuroNet Kuzushiji Recognition service after Kaggle Kuzushiji Recognition competition. In this paper, we talk about the benefit of using object detection algorithm for Kuzushiji recognition. We also explain about the development of KuroNet, Kuzushiji recognition system and the online Kuzushiji recognition service. We also talk about what we learned from Kaggle competition and discuss about the future work for Kuzushiji recognition research including on-device Kuzushiji recognition application with smartphone and how to expand the Kuzushiji dataset for machine learning.

１．はじめに

日本は大量の歴史的資料がよく保存されている国である。その規模は 1 億点とも 10 億点ともいわれ、これらの資料を読み解ければ、これまで知られていなかった多くの事実がどれだけ見えてくることだろうか。ところが、そこに立ちはだかるのが「くずし字」である。くずし字をちゃんと読める人は人口のたった 0.01%だけなので、専門家以外、くずし字資料を利用することができない。

この問題を解決するために、人文学オープンデータ共同利用センターではくずし字認識研究をスタートした。そして最初のくずし字認識の論文を、2018 年の人文科学とコンピュータシンポジウム（じんもんこん）で発表した[1]。この論文を投稿した当時、認識できるくずし字はわずか 10 文字しかなかったため、人間の目でも解読しにくい文字として「ゝ」「し」「く」などを選択した。このように性能に限界があるモデルだったものの、その成果からは、「物体検出」によるくずし字認識の可能性を感じることができた。その後、稿者らはくずし字認識モデル「KuroNet」[2]を提案し、これを段階的に改善することで、現在は数千字種のくずし字が認識でき、条件が良ければ 85～90%の精度が出るまでに至った。

本論文はこのような物体検出によるくずし字認識の研究の展開を中心に、2 年前のじんもんこんの発表当時からの研究の進展を述べる。その主な項目は、Kaggle くずし字コンペ、KuroNet のサービス化、くずし字認識スマホアプリなどである。こうした項目について考察した後、最後にくずし字認識研究の今後の課題についても考察する。

２．物体検出によるくずし字認識

2.1 くずし字認識に特有の問題

機械によるくずし字認識の研究は決して新しい研究課題ではない。しかし、従来の文字認識アルゴリズムは、認識精度が 50%以下にとどまるなど、実用性の高いレベルには至らなかった。それはなぜなのだろうか。その理由は、歴史的資料に書かれている文字（くずし字）と言語を、現代の言語（現代日本語）の感覚で解読しようとしたからではないかと考えている。

第一に、文字の問題がある。現代日本語では、文法や読み方、読み順などは標準化されている。一方、くずし字には、「変体仮名」など現代日本語に存在しないさまざまなルールが多くある。また、くずし字には、そもそも、漢字、平仮名、片仮名のどれなのかを判断しづらい文字もある。例えば、「見」という文字が漢字に見えたとしても、実際にはその文字は平仮名として使われていることが多い。また「二」、「三」、「八」の場合は、漢字、平仮名、片仮名のどの文字として使わ

れているかは、単独の文字だけでは判断できず、文脈を見ないと判断することはできない。さらに「屋」の場合も、漢字なのか平仮名なのかを判断することは文章を読まないと難しい。最新の KuroNet もこの問題に影響を受けている。くずし字を現代文字に変換することを評価基準にすると、たとえモデルが正しく認識しても、評価データのラベルが異なれば不正解ということになってしまい、精度が上がらないことになる。最悪の場合、モデルの開発者はモデルに問題があると誤解し、モデルを悪い方向に調整して本来の精度が悪化することもあり得る。このように、データの問題とモデルの問題を区別するためには、くずし字の専門知識が必要となる。

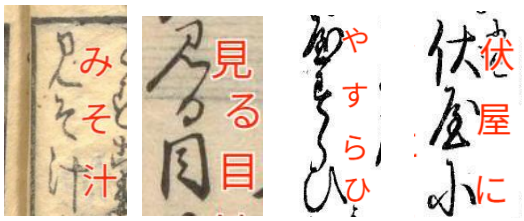


図 1 くずし字データセットで見られるデータラベル。漢字か平仮名か文脈で判断する「み」、「見」と「や」、「屋」の例

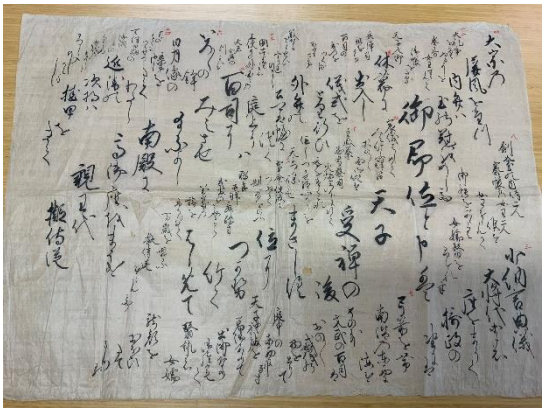


図 2 人間にでも読み順を判定しにくい資料（稿者私物）

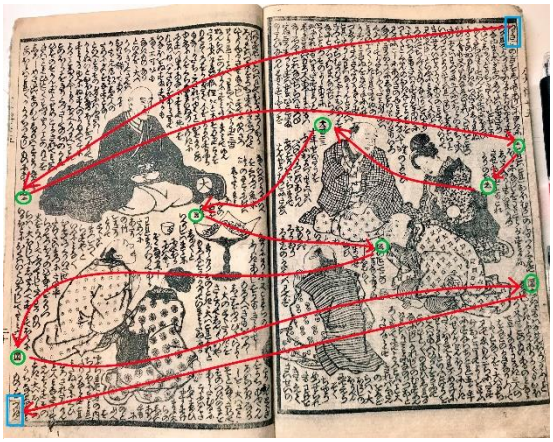


図 3 複雑なテキストレイアウトの資料（稿者私物）

第二にレイアウトの問題がある。図 2 のように人間にとっても認識が難しいレイアウトもある。一般的な文字認識では、モデルが画像を読み込み、レイアウト解析を行い、一行一行の文字を分割し、文字認識を行うという流れで処理が進んでいく。しかし、くずし字資料の場合はレイアウトが自由であり、行というものを定義することも簡単ではない。現代日本語であれば、横書きは左から右に順番に読むというルールが決まっているが、くずし字の場合はルールに多様性がある。横書きでは右から左の順で読むし、縦書きでは右から左まで、そして上から下までという順番が一般的だが、散らし書きはもっと複雑な読み方となる。図 2 のように文字の大きさと墨の濃淡で読み順を決めることも珍しくなく、一行も縦横まっすぐにはなっていない。また版本の場合、一枚の板に文字や絵を合わせて彫るため、図 3 のように読み順を示すための印を使うなど、当時の読者にも難度が高いレイアウトになっている。このようにレイアウト解析は、くずし字認識においてもっとも難しい課題だと考えている。

くずし字の場合にレイアウト解析が難しいのには、他の理由もある。

1. 近世の版本では漢字にルビが振ってあることが多いが、くずし字データセットではルビにラベルが付いていないことが多く、画像中にはラベルのない文字が存在することになる。
2. 漢文の句読点も、文字は存在するがラベルが存在しない点は、ルビと同様である。
3. 版本は一枚の板から版木が作成され、絵入だと文字は絵の周りに書かれることが多いため、字は絵の余白に押し込まれ、時には一行に一文字しかない場合も出てくる。

このように、くずし字に特有の問題が存在することを認識しないまま、現代の言語に対して開発された文字認識をそのままくずし字に対して拡張しようとすれば、失敗するリスクが高まると言える。

2.2 処理順の逆転というアイデア

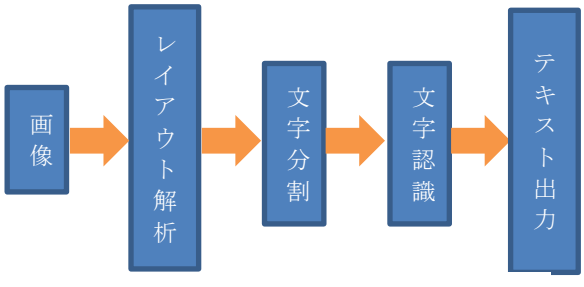


図 4 一般的な文字認識の手法

図 4 は、典型的な OCR（光学的文字認識）ソフトウェアの処理手法を示す。この手法では、画像を入力した後の最初の処理がレイアウト解析となる。しかし最初のレイアウト解析で失敗して

しまうと、その後の文字認識も失敗してしまい、全体として OCR の精度が向上しない。つまり最も難しいレイアウト解析を最初に行うという処理順になっていることが、くずし字 OCR の精度が向上しない最も大きな原因ではないかと考える。

これに対して稿者らは、難易度が高いレイアウト解析を後に回して文字認識を先に行うという、処理順を逆転させた全くくずし字認識モデルの研究を進めた。そして文字認識の部分には、画像中のどこに何があるかを画像中から直接探し出す物体検出（object detection）技術を適用することで、レイアウト解析をしなくても文字認識ができるようにした。

図 5 Google Cloud Vision API Text Detection[3]

このようなアプローチをくずし字認識に適用したのは稿者らのチームが最初であろうが、風景画像から文字を取り出すという問題は、物体検出の研究分野で以前から進められてきた。図 5 のように、カメラで撮影された画像からテキストを認識する Scene-Text Detection も実用化されている。このような手法を適用する際に、一文字ごとの認識に意味がある言語とあまり意味がない言語がある。例えばアルファベット文字の言語では、一文字ごとの認識にはあまり意味がなく、ワードごとに認識したほうが後処理で修正しやすい。一方くずし字認識の場合は、一文字ごとに認識した方が、後処理などの次のステップを進めやすくなる。

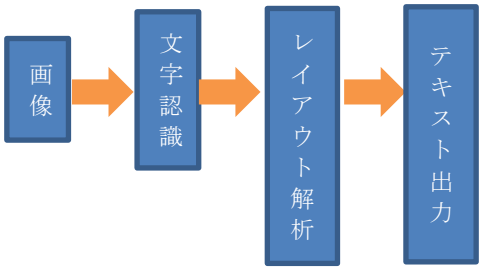


図 6 物体検出によるくずし字認識の手

物体検出アルゴリズムを使ったくずし認識の手法は図 6 の流れとなり、難易度の高いレイアウト解析が後回しにされている点が大きな特徴である。しかし物体検出はコンピュータビジョン分野の中でも応用が多く研究が盛んなため、アルゴリズムも多数開発されている。それぞれのアルゴリズムには長所と短所があり、どのアルゴリズムがくずし字認識に適切かは実験してみないとわからない面がある。

2.3 KuroNet の進化

KuroNet の最初のバージョンには U-Net というアルゴリズムを活用した。しかしこの時点ではまだアルゴリズムは単純で以下の問題を抱えていた。まずモデルに入力する古典籍 1 ページの画像サイズは 512x512 pixels にとどまり、認識できる文字種も最大で 409 字種に限られていた。画像サイズに 512 pixels を選択した理由は、1 ページの画像がこのサイズであれば、人間の目でまだ文字が読めたからである。しかしこの選択はよくなかった。また、GPU メモリーの制限の問題もあり、多くの文字を認識することができなかった。今になってみると、KuroNet の問題は、稿者たちが End-to-End にこだわりすぎた点にあることがわかる。1 ページの画像を 512x512 pixels に縮小すると、書かれている文字は非常に小さくなり、モデルが認識するには小さすぎるサイズとなってしまった。

2019 年 9 月に発表した KuroNet は、こうした問題を解決するためにモデルを大幅に改善したバージョンである。まず U-Net より安定した Residual U-Net（特に FusionNet [4]という Residual U-Net のバリエーション）を採用した。次に GPU メモリーの問題を解決するために、Teacher Forcing [5]というテクニックを取り入れ、画素ごとに文字があるかを判断し、文字のある確率が高い画素だけ文字認識を行う手法とした。さらに精度を向上させるために、Mixup Regularization [6]も採用した。Mixup Regularization は学習画像の Opacity を 70%、無関係に選んだランダム画像の Opacity を 30%にして、学習画像の上にランダム画像をノイズとして重ねて学習する方法である。あえて学習しにくくすることによって、モデルをより鍛えることを狙う。このような Data Augmentation を取り入れることで、KuroNet の精度は 10%ほど高くなっただけでなく、GPU メモリーを節約するテクニックにより、入力画像のサイズも 976x976 pixels にまで拡大することができた。

そして、2020 年 2 月に発表した改善版の KuroNet [7]では、学習の際に画像をランダムに切り取る Random Cropping を採用した結果、認識精度はさらに向上しただけでなく、さまざまな文字サイズへの対応も改善し、テストデータの平均精度は 85%～90%に達した。これは後述する Kaggle コンペの勝者の精度 95%よりは低いが、KuroNet はコンペに優勝することが目的ではないため、勝つための最適化は行っていない。この KuroNet モデルは、現在 CODH が IIIF Curation Viewer を

用いて公開する KuroNet くずし字認識サービスで実際に使われている。

KuroNet は画素ごとに文字種を推定するため、複雑なレイアウトの影響を受けず、くずし字のように続けて書かれる文字（連綿体）の認識も苦にしないという特徴がある。

３．Kaggle くずし字認識コンペ

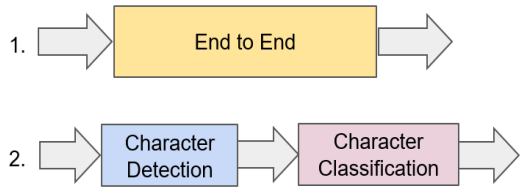


図 7 Kaggle Kuzushiji コンペの上位の手法

CODH はくずし字データセットを活用した国際的な規模のコンペティション「Kaggle くずし字認識」を開催した。このコンペについては、すでにいくつかの論文[8][9]で経緯を紹介しているため、本論文ではその詳細は割愛する。

Kaggle くずし字認識コンペティションの参加者の何人かは、オープンソースでモデルを公開した。そこからは多くのアイディアを得ることができ、稿者にとっても勉強になった。各モデルには強みと弱みがあり、テストデータで精度が一番高いモデルがベストとは必ずしも言えない。参加者のモデルは、そのまま実サービスに投入できるというより、どのアルゴリズムがくずし字認識のどの問題を解決できるか、というアイディアの集合体のようなものと考えればよい。

例えば、上位参加者の手法を大ざっぱに分けてみると、2 種類に分けられる。まず KuroNet のような End-to-End 手法、そして文字検出してから文字認識を行うという 2 段階の手法である。それぞれの手法には長所と短所があり、どちらの手法の方がよいのかは一概に判断できない。しかし 2 段階の手法は文字検出（Character Detection）モデルの精度が 98%以上もあることがわかった。これは物体検出アルゴリズムの Region Proposal Networks (RPN)と似たようなアイディアである。RPN は入力画像中から物体候補領域を抽出するためのネットワークである。くずし字は続けて書かれている文字のため、RPN で文字分割して、単独の文字を見るだけでは、どの文字なのかを認識しづらいという問題はある。しかし、文字検出から得られる Bounding Box は KuroNet では出せない情報である。KuroNet は画素ごとに画像を見るため、Bounding Box をそこから計算することはできなかった。この文字検出のアイディアは、KuroNet の改善に利用するというよりも、今後のくずし字研究にとってもっとも重要な課題、すなわちくずし字データセットの構築をスピードアップするという目的に使えることが期待できる。このデータセット構築の課題については、6．くずし字認識研究の課題でも触れる。

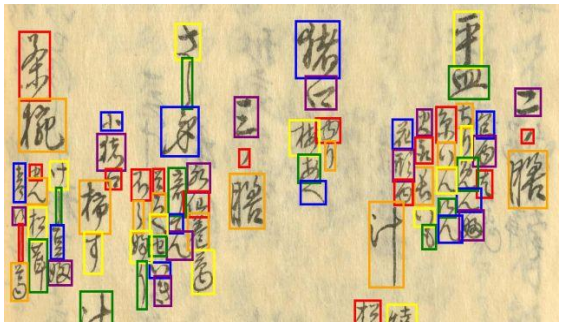


図 8 Kaggle 優勝者[10]の文字検出モデルで認識した Bounding Box

４．くずし字認識のサービス化

2019 年 11 月に公開した KuroNet くずし字認識サービスは、国内外の多くの美術館、図書館が利用する画像公開方式 IIIF に基づき、世界中で公開されている資料にプログラミングなしでくずし字認識を適用できるサービス[11]である。KuroNet サービスの精度と認識スピードは研究者にとっては有用であるが、IIIF で公開されていない資料には対応しておらず、手持ちの資料をすぐに調査できないという問題がある。しかし、KuroNet サービスが現段階でこのような方式になっているのには以下の理由がある。

まず、IIIF で公開されているくずし字資料は全世界に数多く存在しており、これらの資料を認識できるサービスを立ち上げることが最優先の課題だった。例えば、国文学研究資料館が新日本古典籍総合データベースで公開している資料を翻刻し、検索できるようにすれば、今後の人文学研究に大きな貢献になることが期待できる。2020 年、新型コロナウイルス感染症（COVID-19）の影響は多くの研究者が受けており、現場で資料調査を行うことが難しくなっている。そのため、オンラインで資料調査できることの重要性はますます大きくなってきている。

次に、KuroNet も Kaggle のくずし字認識モデルも、撮影環境や画質など、画像の状態に認識精度は大きく影響されるという問題がある。図書館や美術館から IIIF で一般に公開されている画像は、同じ角度、同じ大きさで、できるだけ画質が高くなるようにきれいに撮影されたのがほとんどである。こうした画像は、くずし字認識モデルにとってベストの状態となっている。このような理由から、KuroNet くずし字認識サービスは、当面の間は IIIF 画像のみを対象としたサービスとして運用する計画である。

KuroNet サービスのもう一つの課題は、文字認識を行った後のテキスト出力である。上述したようにくずし字資料はレイアウト解析が難しいため、KuroNet はレイアウト解析を後回しにしている。しかしテキスト出力するには、レイアウト解

析を行わなければならない。ゆえにテキスト出力までを考えると、レイアウト解析は必要な技術である。しかし一般のレイアウト解析と異なるのは、すでに文字コードと座標が得られているという点である。つまり KuroNet の後のレイアウト解析は、このような文字コードと座標を入力データとしたレイアウト解析ということになる。

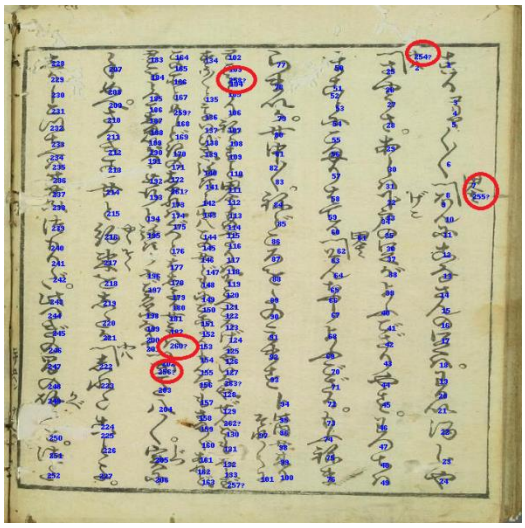


図 9 ディープラーニングによる読み順（文字シークエンス）推定結果。赤い丸は推定が間違っている箇所である。

レイアウト解析を行うため、稿者のチームはまず、シークエンスを扱えるディープラーニングモデルによる読み順推定を試みた。これはうまくいくときは自然な読み順を推定する。しかし間違えるときはランダムに間違えるため、たとえ精度が99%であっても残り 1%の間違いを探すには大変な労力がかかることがわかった。ディープラーニングはブラックボックスであり、ページごとに原因も異なるため、なぜそのようなミスが発生するかを説明可能とすることは困難である。

間違いを探すために人間が最初から最後まで読まないといけないのならば、自動推定は人間の手間を減らすことに寄与しない。そこで稿者らは、ディープラーニングで読み順推定する手法の研究を中断し、簡単なレイアウトに対してシンプルなルールベースのアルゴリズムを適用することとした。こちらの場合はルールを目で見ることができるため、どこで間違えているかを少なくとも説明することができる。現在の KuroNet くずし字認識サービスでは、こちらのシンプルなルールベースの読み順推定アルゴリズムを利用している。

ところが、IIIF 形式で公開されていない資料や、個人蔵の手持ちの資料は少なくない。KuroNet サービスを公開した後、手持ちの資料を認識したいという問い合わせが多数あった。この需要に応えるためには、任意の画像をアップロードできるサービスを公開するという可能性もある。しかしそうしたシステムを公開する前に、検討すべき課題がいくつかある。

1. ユーザが自分の資料をサーバにアップロードする方法では、歴史的資料であっても個人の手紙や日記の場合、その内容が個人のプライバシーに関わる可能性がある。
2. システムが使いやすくなれば、ユーザが増えてサーバの負荷も高まるため、サーバ管理やハードウェアのコストが増大する。
3. インターネットに接続できる環境で利用する必要があるため、データ通信にかかるコストが増大する。

3 番目の問題は、KuroNet をオフライン認識できるような形で提供できれば解決できる。しかし、オフライン版の KuroNet を使うには、高性能のパソコンと GPU を用意し、Ubuntu、PyTorch、CUDA などの各種ソフトウェアをインストールし、Python スクリプトを実行しなければならい。くずし字認識の大半のユーザである一般の人、高齢者、文系研究者、図書館、博物館の学芸員には、こうした要求はハードルが高すぎると言わざるを得ない。

稿者が考える理想的なくずし字認識サービスとは、手持ちの資料の写真を撮影し簡単に認識することができることである。これはスマホ上でくずし字認識ができれば実現できるはずである。そこで、誰でも使いやすいくずし字認識のスマホアプリ開発に着手した。

５．くずし字認識スマホアプリ

図 10 稿者の Flutter フレームワークで開発したくずし字認識スマホアプリ。

スマホアプリでくずし字認識を行うという方法は、複数の資料画像を大量処理するという用途には向かないが、手軽にくずし字認識を行えるという利点があるため、このアプリを実現することは重要な研究課題と考えている。くずし字認識をスマホアプリで実現するには、3 つの方法が考えられる。

A) スマホカメラで資料を撮影した画像をサーバに送信し、サーバ側でくずし字認識を行う、サーバ認識の方法。
B) スマホカメラで資料を撮影した画像を、ス

マホ内でくずし字認識を行う、オンデバイス認識の方法。

C) スマホカメラで資料を撮影するのではなく、カメラを資料の上にかざすだけでリアルタイムにくずし字認識結果を表示する、リアルタイムオンデバイスくずし字認識の方法。

サーバ側の高性能 GPU が使えるため、A の精度は当然ながら最も高くなる。しかも現在の KuroNet サービスの API を活用すればすぐに実現でき、オンデバイス用の新しいモデルを開発する必要はない。そこで KuroNet くずし字認識サービスを活用し、Flutter[12]フレームワークでインターフェースを構築することで、スマホアプリによるくずし字認識のデモを作成した。

スマホアプリの開発では、iOS と Android という 2 大 OS への同時対応が大きな課題である。しかし Flutter を使えば、iOS と Android しかも Web やデスクトップに使えるアプリを一度に生成できるため、2 つの OS に別々に対応する必要がない。その点で Flutter は便利なフレームワークである。

さてスマホアプリとして実現する A、B、C の方法について、認識精度と開発の難度から見れば A が最も精度が高く、開発もしやすい。C はおそらく最も精度が低く、開発難度も高い。とするならば、B や C は必要だろうか。A は確かに開発の工数としては少なくなるだろうが、サーバ側処理には、プライバシーの問題や通信費用の問題だけでなく、一枚の画像を認識するのに 5−6 秒かかるという問題もあり、大量の利用者をさばききれない可能性がある。一方、スマホ側の性能の進化も早く、スマホ上でできることもどんどん増えている。もしサーバに頼らずスマホ側で処理できることが増えれば、それだけサーバ側の負担は減少する。

B と C の方法はスマホのハードウェアの性能にも大きく依存する。最新の Google Pixel phone (Pixel Neural Core)、iPhone や iPad (Apple Bionic Chip Neural Engine)は、オンデバイス機械学習向けに開発されたため、スピードが速く、電源消費も少ない。これらのハードウェアを活用したくずし字認識アプリが開発できれば、ユーザの環境設定や、サーバの負荷に関する問題を減らせる可能性も十分にある。

以上のことから、精度は多少低くなるかもしれないが、認識スピードが速く、プライバシー保護も可能で、通信費用もかからないという、オンデバイスのくずし字認識は、今後に向けて重要な研究課題だと考えている。

６．くずし字認識研究のこれからの課題

くずし字認識システムを改善するには、データセットの規模を拡大することがもっとも重要な課題であると考えている。KuroNet と Kaggle コンペで使用したデータセットは、国文学研究資料館が作成したくずし字データセットである。このデータセットは、近世の古典籍 44 点から作成されており、古典籍の画像は約 6000 枚、文字数は 100 万文字以上に達する。このデータセットのポイントは、一丁全体の画像と、Unicode ごとにまとめられた文字ごとの画像、そして文字が一丁の画像のどこにあるかを示す座標データを含んでいる点にある。この座標データが物体検出アルゴリズムには不可欠となるが、座標データの作成は基本的に人力で行っているため、時間もかかる大変な作業となっている。そこでこれからの重要な課題は、モデルの改善のためのアルゴリズム研究より、むしろデータセットの拡大なのではないかと考えている。そのためには、データセット作成の作業に要するコストをできるだけ減らす必要がある。

古典籍の翻刻データは、基本的に画像とテキストしかない。しかし物体検出アルゴリズムを適用するには、座標データが不可欠である。ということは、座標データが作成できるシステムがあれば、翻刻テキストからくずし字認識用のデータセットを構築するためのコストを大きく削減できることになる。そこで Kaggle 優勝者の文字検出モデルを利用し、Precision の高い Bounding Box を自動的に作成できるようにした。そして KuroNet によるくずし字認識結果と、認識した Bounding Box とをマッチングし、KuroNet が間違って認識した文字のラベルを人間が翻刻したテキストで修正できれば、大量に存在する翻刻テキストデータをデータセット化することが可能になる。そして、最終的に人間による確認を加えていけば、さらにデータセットの品質を向上させていくことも期待できる。

データセットを大規模化すればそれに応じて KuroNet の精度も向上するという単純な話ではないものの、データセットが重要であることは間違いない。ディープラーニングは大抵の場合、データが多ければ多いほど、より精度の高いモデルを作成することができるからである。しかしくずし字認識モデルの場合、データが多くなるとそれだけ認識すべき文字の種類も多くなる。そして、コンピュータの計算能力もそれだけ多く必要となる。この計算能力は個人用のパソコンでは足りないことが多いため、ソフトウェア開発ができても実行環境の整備に困難が生じる。また文字種類を増やせば増やすほど、認識スピードも遅くなる。

このようにくずし字認識には精度、スピード、計算力のトレードオフの問題がある。精度が高くなると、スピードが遅くなり、計算力も必要になる。スピードが早いと精度が落ちる。最もサービス化に適したモデルとは、単純に精度が最も高いモデルではなく、精度、スピード、計算力のバランスのよいモデルである。こうした問題を考えながら、くずし字認識モデルを今後も改良していく

必要がある。

最後にモデルの評価に関連して、Kaggle コンペのデータセットを利用してモデルのパフォーマンスを比較する手法にも問題があることを指摘しておきたい。Kaggle コンペでは優勝者の精度が 95%に達したが、この数字は適切なものと言えない面がある。Kaggle コンペを開催する際に、非公開のデータしかテストデータに使えないという縛りがあったため、テストデータの分布が偏ることになってしまった。コンペ開催当時、非公開のデータはほとんどが物語ジャンルのものだったため、テストデータは平仮名が多く、漢字が少ないデータセットとなってしまった。ゆえに Kaggle のテストデータは、学習データよりもはるかに簡単なデータセットとなってしまったのである。例えば頻度の高い最初の文字 1200 字種だけを完璧に認識できれば、精度はそれだけで 85%以上に達する。しかしこれは理想的な評価基準とは大きく異なる。くずし字認識モデルに期待するのは、人間には解読しにくい文字の認識であり、人間にも読める簡単な文字の認識ではない。以上の問題を踏まえると、Kaggle テストデータで精度を競うのは適切ではなく、Kaggle データセットで新しいモデルを比較することも積極的にはお勧めしない。

くずし字データセットの公開者としてやるべきことは、データセットの文字種の分布に配慮したうえで、学習データとテストデータを注意深く分割することである。データセットの適切な分割は、今後くずし字データセットに新しいデータを追加する際に配慮すべきことである。機械学習研究の面では、他のモデルと比較しやすいデータセットが好まれるが、くずし字研究の面ではより正当なパフォーマンス評価に近づくように、データセットの分割を継続的に更新していくことが望ましい。そしてその分割は決してランダム戸はせず、くずし字に関する専門知識を有する人が行うべきである。

7．おわりに

くずし字認識は、文学や歴史の研究に大いに貢献すると期待できるが、研究課題はまだ多くある。これらの課題を解決していけば、モデルの精度も向上し、誰でもくずし字資料を利用できるようになる未来はそう遠くないと考えている。

参考文献

[1] Tarin Clanuwat, Alex Lamb, Asanobu Kitamoto “End-to-End Pre-Modern Japanese Character (Kuzushiji) Spotting with Deep Learning”. 人文科学とコンピュータシンポジウム、じんもんこん 2018 論文集, (参照 2018-12)

[2] Tarin Clanuwat, Alex Lamb, Asanobu Kitamoto “KuroNet: Pre-Modern Japanese Kuzushiji Character Recognition with Deep Learning”. The International Conference on Document Analysis and Recognition (ICDAR) https://arxiv.org/abs/1910.09433, (参照 2019-09).

[3] Google Cloud Vision API Detect text in images. https://cloud.google.com/vision/docs/ocr (参照 2020-11)

[4] Quan TM, Hildebrand DGC, Jeong W. “Fusionnet: a deep fully residual convolutional neural network for image segmentationin connectomics”. CoRR http://arxiv.org/abs/1612.05360 (参照 2016).

[5] Lamb AM, GOYAL AGAP, Zhang Y, Zhang S, Courville AC, Bengio Y. “Professor forcing: a new algorithm for training recurrent networks”. In Advances in neural information processing systems. (参照 2016). p. 4601–4609.

[6] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. “mixup: Beyond empirical risk minimization”. arXiv preprint arXiv :1710.09412 (参照 2017).

[7] Alex Lamb, Tarin Clanuwat, Asanobu Kitamoto “KuroNet: Regularized Residual U-Nets for End-to-End Kuzushiji Character Recognition”. Springer Nature Special Issue on Document Analysis and Recognition, https://link.springer.com/article/10.1007/s42979-020-00186-z (参照 2020-05).

[8]北本 朝展, カラーヌワット タリン, Alex LAMB, Mikel BOBER-IRIZAR, "くずし字認識のための Kaggle 機械学習コンペティションの経過と成果", 人文科学とコンピュータシンポジウム じんもんこん 2019 論文集, pp. 223-230, 2019

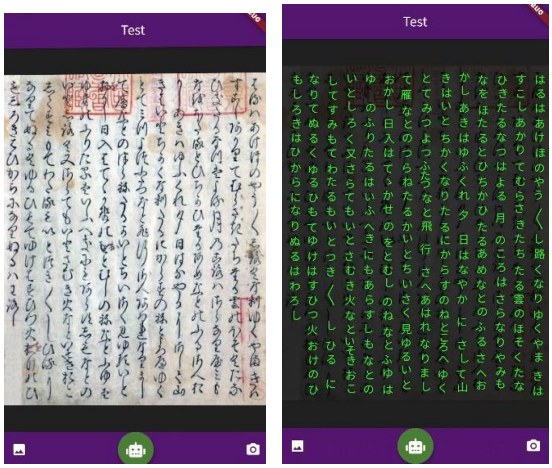
[9]北本 朝展, カラーヌワット タリン, ボーバー・イリザー ミケル, "Kaggle くずし字認識—世界規模の人文系コンペ開催への挑戦—", 人工知能学会誌, Vol. 35, No. 3, pp. 366-376, 2020

[10]Kaggle Kuzushiji Recognition コンペの優勝者 Tascj チームモデル。https://github.com/tascj/kaggle-kuzushiji-recognition （参照 2020-11）

[11]北本 朝展, カラーヌワット タリン, "AI によるくずし字認識と歴史的資料全文検索への道", 専門図書館, No. 300, pp. 26-32, 2020

[12]Google の開発した Flutter Framework, https://flutter.dev/ (参照 2020-11)

付録



くずし字認識スマホアプリ（iOS 版）の認識画面

IIIF Curation Viewer 上の KuroNet くずし字認識サービス