

両面印刷に対応した点字学習支援アプリケーションの開発

大貫 佑真[†] 秋岡 明香[‡]

明治大学 総合数理学部

1. はじめに

点字の自学は難しく、日常的な情報入手手段として点字を利用する人は、視覚に障がいを持つ人の1割未満である[1]。また、視覚に障がいを持つ人の53.1%がスマートフォンを利用している[2]。スマートフォン利用率は今後増加すると考えられる。そのため、点字の自学を支援する目的で、点字をスマートフォン等で撮影することで語彙を認識するアプリケーションを開発してきた[3]。しかし、点字が両面印刷されている場合を想定しておらず、表面と裏面を区別していないという問題があった。先行研究として両面印刷された点字に対し、凸点と凹点の明暗の特徴から点字を認識する研究がある[4]。しかし、1ページ全体の認識では実用性に乏しい結果であると述べている。本稿では、物体認識アルゴリズムの手法である YOLOv5s[5]を用いて、両面印刷に対応したアプリケーションを開発した。また、以前のアプリケーションを iOS で実行可能にし、処理を軽くすることで汎用性を高めた。

2. 提案手法

2.1 全体の流れ

本研究ではこれまでに、スマートフォン等で撮影した画像を入力とし、入力画像の2値化、確率的ハフ変換による傾き検出、画像の回転、指で指し示した行の選択、テンプレートマッチングによる点検出、点字の認識という順序で点字を認識するアプリケーションを開発してきた。本研究では、入力画像の2値化、テンプレートマッチングによる点検出、点字の認識の箇所を改良した。凸点と凹点を検出するため、データを作成し YOLOv5s のモデルを学習させた。2値化の前処理と点の検出には、学習させたモデルを使用した。また、点字の認識の箇所を改良し、以前より認識速度を向上させた。それに加え、以前のシステムを iOS で実行可能なように書き換えることで、スマートフォン上での実行を可能にした。

2.2 YOLOv5s の学習

スマートフォンでフラッシュを使い屋内で撮影し

Development of learning braille support application that support duplex printing.

[†] Yuma Onuki, Meiji University

[‡] Sayaka Akioka, Meiji University

た画像を 640×640 の大きさに切り抜き、凹点と凸点のラベル付けを行う。図1にラベル付けの例を示す。総枚数 1500 枚のうち、1200 枚を学習用、300 枚を検証用とし、YOLOv5s のモデルで学習を行う。学習済みモデルを iOS で実行するために、onnx 形式を経由し、CoreML 形式に変更する。

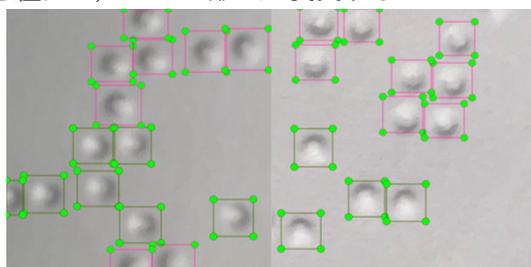


図1. ラベル付け画像(凸点:赤,凹点:緑)

2.3 凹凸の2値化

2.2 での学習済みモデルで凹点を検出する。2値化の際、検出した点の箇所を白塗りすることで裏面に印刷された点を考慮する必要がなくなる。本研究で用いた入力画像のサイズは 3024×4032 である。モデルの入力サイズが 640×640 であるため、入力画像を 640×640 に切り抜きを行う。切り抜いた画像の端の点は全体が写らず精度が落ちる可能性があるため、横方向に対し70、縦方向に対し75のゆとりを持たせ切り抜く。切り抜いた画像をモデルの入力画像とし、各結果をIoU値0.45としてNon-Maximum Suppression(NMS)で重複削除する。図2に結果を示す。

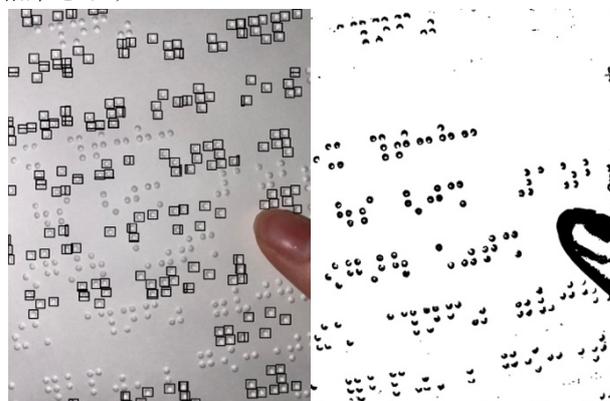


図2. 凹点検出(左)と凹点を削除した2値化(右)

2.4 点の認識

指で示した行の選択を行なったあと、学習済みモデルを使い凸点の認識を行う。テンプレートマッチングによる点の認識と比較し、指の影などが点と認識される可能性がなくなるため、精度が向上する。画像の切り抜きは2.3と同じ方法とする。切り抜いた画像はゆとりを持たせているため、2つの画像に同じ点が写ることがある。点を重複して認識する可能性があるため、NMSを各画像に行うのではなく、切り抜き前の全体画像に対し行うことで重複を避ける。ここではNMSのIoU値は0.05とした。凸点の検出結果を図3に示す。

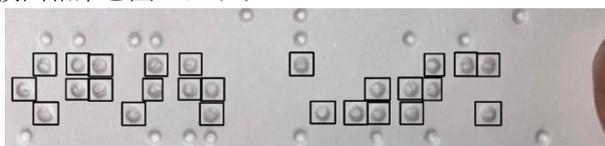


図3. 凸点の検出例

2.5 点字の認識

以前のアプリケーションでは、認識した点を用意した点字のテンプレートに合うよう位置を変更していた。しかし、テンプレートマッチングによる認識は計算量が多く時間がかかる。そこで、左上から右下までの点の位置を「111000」や「001011」のようにし、点字の辞書を作成した。文字ごとに点の位置を保存し、辞書と一致する文字を検索することで認識した。例を図4に示す。

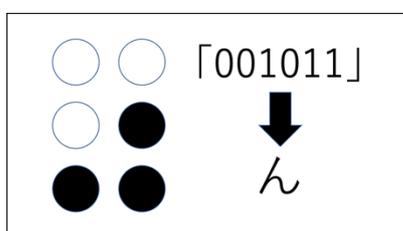


図4. 文字の認識例

3. 実験と評価

スマートフォンでフラッシュを使い屋内で撮影した3024×4032のサイズの画像を用意する。汚れがなく、物体が置かれていない白い紙に両面印刷されているものを用いる。本研究ではiPhone11 Pro Maxを用いる。2値化から回転までの精度、回転後の凸点の認識率、文字の認識率、入力画像から文字認識までの実行時間の測定を、50枚の画像で行う。また、屋外や暗闇などの環境においても試す。

実験結果は、2値化から回転までの精度は98%、回転後の凸点の認識率は96%、文字の認識率は約84%、入力画像から文字認識までの実行時間は平均して23秒程度であった。日光が当たっている状態では凹点が上手く認識できず、回転を行うことができなかった。暗闇では問題なく認識することができた。

4. 考察

2値化から回転までは、画像中の点が少なくハフ変換で正しく直線を検出できないことが原因だと考える。直線を検出できないものは画像の傾きを出すことができないため、直線検出以外の方法を検討する必要がある。回転後に凸点が認識できていないものは、回転前の画像を学習させているため、回転により光の当たり方が変化したことが原因だと考える。文字の認識率はあまり高くなく、照明条件による環境の変化にも弱い。この問題は、フラッシュよりも強い光が当たることで、用意していたデータと大きく異なる入力画像になってしまうことが原因だと考える。また、暗闇では問題なく実行できていることから、フラッシュより明るい光がある時に認識できないと考えられる。そのため、限られた環境だけでなく、様々な照明条件を考慮した環境のデータを学習に使用することで改善できるのではないかと考える。

5. おわりに

本研究では、凹凸に対応したアプリケーションを開発した。屋内の一定環境下では凹凸検出を精度良く行うことができた。しかし、環境の変化に弱いため、今後は環境の変化にも対応できるよう改善する必要がある。また、実際に視覚に障がいを持つ人にアプリケーションを使用していただくことで、意見を取り入れて実用性を高めたいと考える。

参考文献

- [1] 厚生労働省：平成28年生活のしづらさなどに関する調査（全国在宅障害児・者等実態調査），https://www.mhlw.go.jp/toukei/list/dl/seikatsu_chou_sa_b_h28.pdf 2018.
- [2] 渡辺 哲也，加賀 大嗣，小林 真，南谷 和範：視覚障害者のスマートフォン・タブレット利用状況調査2017（福祉情報工学），電子情報通信学会技術研究報告117(251)，69-74，2017.
- [3] 大貫 佑真，秋岡明香：画像処理を用いた点字学習アプリケーション，第82回全国大会講演論文集，397-398，2020.
- [4] 小泉 圭巧，井出 陽，本多 高：両面印刷された点字の自動認識に関する研究，日本シミュレーション学会大会発表論文集：シミュレーション・テクノロジー・コンファレンス・計算電気・電子工学シンポジウム24，285-288，2005.
- [5] Glenn Jocher, yolov5, GitHub repository, <https://github.com/ultralytics/yolov5>, 2020.