

平成30年度 修士論文

ニューラルネットワークを用いた
デジタルペン筆記の改行検出

平成31年2月12日

17350913

岸 兼也

指導教員 三浦 元喜 准教授

九州工業大学大学院 工学府 先端機能システム工学専攻

概要

学習者の筆記データを収集する手段としてデジタルペンがあり、得られたデータを用いて学習者の支援を行う研究が行われている。様々な研究があるが、その中でも筆記を一行ごとに分割し、それを利用することで学習支援を行う研究がある。筆記の分割は手動と自動の2通りで行われると考えられるが、手動で行う場合は学習者に余計な負担を与えてしまうため、自動で分割できるようにする。自動で筆記を分割する手法として、我々はLSTMを用いた手法を提案する。収集した筆記データから各ペンストロークの座標データを学習させることで、筆記の改行検出を行った。我々が手動で改行を判断したものと比較したところ、F値は0.81であった。検出の正確さを向上させるためにデータ拡張を行い、学習データを増やすことで、F値を0.84まで向上させることができた。この結果を関連研究と比較するためにt検定を行ったところ、有意水準5%で有意に高いという結果が得られた。

目次

第1章 序論	3
1.1 背景	3
1.2 筆記の改行	4
1.3 目的	5
第2章 関連研究	8
2.1 筆記の分割に関する研究	8
2.2 改行の検出に関する研究	9
第3章 提案手法	10
3.1 機械学習について	10
3.1.1 機械学習	10
3.1.2 再帰ニューラルネットワーク	10
3.2 LSTMの実装	11
第4章 実験	13
4.1 筆記データの収集	13
4.2 モデルの学習と評価	14
4.3 結果	14
4.4 考察	15
第5章 改善	19
5.1 閾値の変更	19
5.2 特徴の追加	19
5.3 データ拡張	21
5.4 考察	22

5.4.1	改善後の結果について	22
5.4.2	関連研究との比較	25
第6章	結論	27
6.1	まとめ	27
6.2	今後の展望	27
	謝辞	29
	参考文献	30

第1章 序論

本論文では、デジタルペンで収集した筆記に対して、機械学習を行うことによって筆記から改行を検出する方法とその評価について論ずる。本章では、教育と情報技術の関係と、改行の検出を行う理由、そして、本研究の目的について説明する。

1.1 背景

近年、情報技術の発展により、教育のなかで様々なデジタル機器が用いられている。文部科学省は「教育の情報化ビジョン」[1]として、2020年までにデジタル機器の1人1台体制を実現させ、ICT (Information and Communication Technology = 情報通信技術)教育の本格化を目指しており、すでに、いくつかの学校では電子黒板やタブレットを用いた授業が行われている。また、ビジネスマンから学生まで幅広いサービスが存在するオンライン学習であるeラーニングを使用する人も多い。このように情報技術が教育の中で用いられている一方で、「2020年代に向けた教育の情報化に関する懇談会」最終まとめ[2]では、デジタル機器や無線LAN等のネットワーク、システムなどの構築にコストがかかることや、専門知識が必要となることで教員や子供が使いやすい状況になっていないなどの課題が挙げられた。

そこで、学習者にかかる負荷が少なく、より直感的に使用できるものとしてデジタルペンがある(図1.1)[3, 4]。アノト方式のデジタルペン(図1.2)では、特殊なドットパターンが埋め込まれた紙を使用し、ペン先にあるカメラでそれを読み取って筆記を電子化する。そのため、通常の授業で用いられているように紙とペンによってデータを収集できる。このようにして得られた筆記データを用いて、学習の支援ができると考えられている[5]。

デジタルペン筆記を扱う学習支援の研究として、以下のような研究がある。我々は以前の研究で、ペンストロークの時間差を用いた学習者の弱点検出を行った[6]。前のス

トロークからそのストロークを書くまでの時間をそのストロークの時間差として、問題の解答時間と目標解答時間から閾値を設定し、閾値よりも時間差が大きいストロークを弱点のストロークとした。学習者に弱点を提示する際に、弱点のストロークを表示するだけでは使いづらいと考え、弱点のストロークを含む一行を弱点として表示するシステムを作成した。また、前田は多人数学習者筆記の同時閲覧インタフェースの検討を行っている [7]。改行の検出を行い、それをを用いた表示閲覧インタフェースは、数式を理解しやすいという意見を得た。また、飯山らは解答停滞箇所の検出を行っている [8]。筆記から文字列を抽出するために文字認識を行ったが、その最初のステップとして筆記を一行ごとに分割した。その後、抽出した文字列同士を DP マッチングによって対応付けし、筆記の時間データから解答停滞箇所を検出した。また、浅井らは数学手書き答案の解答戦略分類を行っている [9]。始めに、解答ステップグルーピングとして改行の検出を行っている。解答ステップグルーピングを行った後に、解答ステップ距離、解答戦略距離を算出し、答案を解答戦略ごとに分類した。この分類は、教師の採点や分析の負担を減らし、さらなる教育の質向上へとつなげることが可能であると考えられる。これらの研究のように、デジタルペンの筆記データに基づく学習支援を行うための処理として、筆記の改行を検出することはよく行われている。

1.2 筆記の改行

筆記の改行について、通常の筆記では図 1.3 のように筆記が上下に移動した場合に改行とする。特別な筆記として、図や表は 1 つの行として扱う (図 1.4)。また、上下に移動がほとんどない場合でも、右側に細かい計算をする場合のように意味の異なる内容を書く場合は改行とした。その他にも、答えに下線を引いている部分や、「{」内に式が複数並ぶ場合も 1 つの行として扱う。また、我々は筆記の時系列を考慮し、1 つの行を書いて改行した後に、もう一度その行内に戻って書いた場合は別の行として扱う。これは、リアルタイムで改行検出を行う際に、入力されたデータに対して時系列に処理を行うことを想定しているためである。

1.3 目的

筆記を一行ごとに分割する方法として、手動と自動の2通りが考えられる。手動で分割を行う場合、筆記の途中に行う場合と筆記を終えた後に行う場合がある。筆記の途中で行う場合は、学習者が意図的に改行を行ったと判断する必要があるため、学習活動を阻害する可能性がある。また、筆記を終えた後に行う場合も、本来の学習とは関係のない筆記の分割のためのステップが介入するため、学習活動の阻害につながると考えられる。それだけでなく、手動の場合では筆記のデータ量に応じて作業量が変化するので、多くのデータに対して手動で分割するのは負担が大きいと考えられる。

本論文では、筆記を自動で分割するために改行を検出する手法について提案し、その評価を行う。

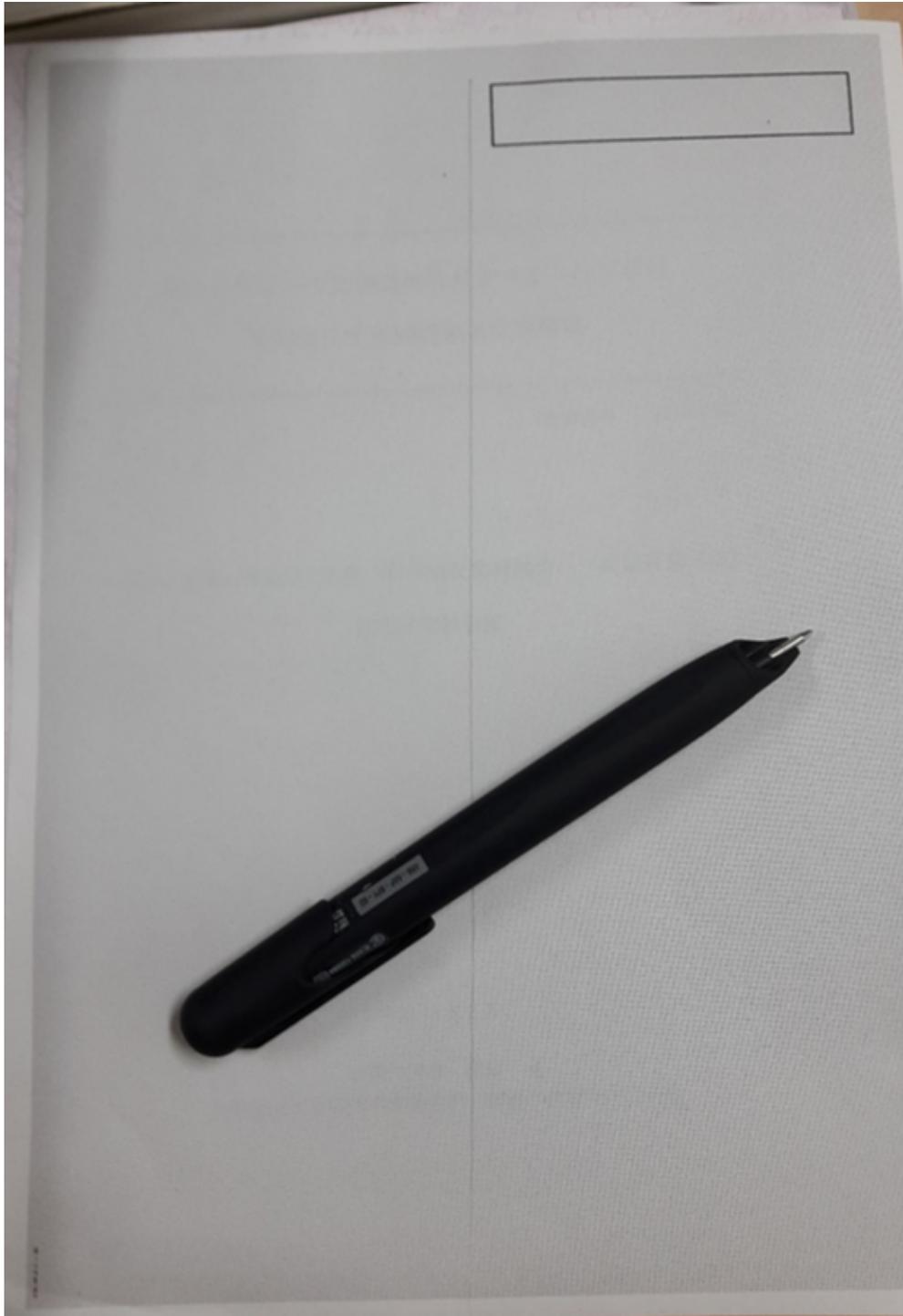
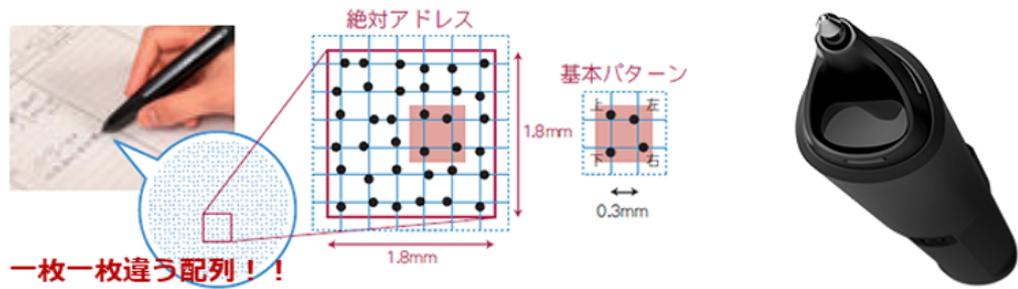


図 1.1: デジタルペンと用紙



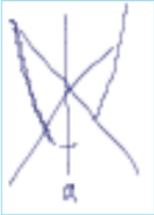
出典 <http://www.ctc-g.co.jp/~ctcsp/products/anoto/index.html>

図 1.2: アノトペンの仕組み

$$f(a) = 3a^2 - 6a - 36 \rightarrow -27 \quad \text{改行}$$

$$\rightarrow 3a^2 - 6a - 9 = 0$$

図 1.3: 改行の例

$$p = 3a^2 - 6a - 36 \rightarrow \text{改行}$$


1つの図中で改行はしない

図 1.4: 特殊な改行の例

第2章 関連研究

本章では、関連研究として筆記の分割や、行の検出を行っている研究について紹介する。

2.1 筆記の分割に関する研究

大西らは講義映像から板書のブロック分割を行った [10]。板書のブロックは小さな矩形の集合であると考え、一定時間書き込みのない場合、その箇所を板書矩形として抽出した。そして、講義者の板書のレイアウトから生成したファジールールに従って、それらの矩形の統合・分割を行うことによって、板書をブロックに分割した。

工藤らはデジタルノートの再構成支援を行うにあたって、ある一定の区切りによって区切られた文章や図形などを一つの「群」として認識する群認識を行った [11]。これはデジタルペンで文章を記入した後に「送信」ボタンをタップすることで、「送信」から「送信」までに記入した文章を一つの「群」として認識する手法で、手動で筆記の分割を行っている。

待井らと Truyen らは手書き文書に対して、文字と図形の分離、分類を行った [12, 13]。待井らは文字パターンと図形パターンのストロークの性質の違いを利用することで文字と図形の分離を行った。また、Truyen らは単一ストロークから得られる様々な特徴を用いることで各ストロークが文字か図形かの分類を行った。

Dieu らは文書画像から表領域の識別を行った [14]。文書画像から関心領域 (ROI) の検出と拡大を行い、その後、すべてのテキストブロックの配置と閾値から ROI が実際の表であるかの判定を行った。

我々の研究では、これらの研究のように大きい範囲の検出や分離ではなく、筆記を 1 行ごとに分割するために改行の検出を行う。

2.2 改行の検出に関する研究

後藤は手書きの答案用紙に対して改行の検出を行った [15]. まず, 答案用紙から図やグラフを除去し, その後, 細線化を用いた手法, 細線化とハフ変換を用いた手法, 処理領域を分割した手法の3つの精度と処理時間を比較した. その結果, 処理領域を分割した手法が検出率 91.1%, 処理時間 4.8 秒となり, 他の手法よりも 5% 近く精度が向上し, 処理時間は 10 分の 1 にすることができた.

前田は多人数学習者筆記の同時閲覧インタフェースの検討を行う際に, 改行の検出を行った [7]. 書いている線と 1 つ前の線の座標データを用い, 20mm 以上離れたときを認識条件としたとき, 改行の検出率は最も高い平均 63.92% であった.

飯山らは解答停滞箇所の検出を行う際に, 改行の検出を行った [8]. まずペンストロークデータから文字の大きさを算出し, 文字の大きさと 1 つのストロークの x 座標と y 座標からそのストロークで改行されているかを判断した.

浅井らは数学手書き答案の解答戦略分類を行う際に, 解答ステップグルーピングとして改行の検出を行った [9]. 時系列で隣り合う 2 ストローク間の重み付けマンハッタン距離が閾値を超えた場合, 改行と判断した.

Yi らは手書き文書の改行の検出を行った [16]. テキスト行の構造を強化するためにガウシアン窓によって二値画像をグレースケールに変換し, レベルセット法を用いることで改行を検出した. また, この手法で, 英語, 中国語, 韓国語などのいくつかの異なる言語で書かれた文書でテストすることで, 言語に依存しないことを定性的に確認した.

我々の研究では, 改行の検出に機械学習を用いることが異なる.

第3章 提案手法

本章では、筆記の改行を検出する手法について提案を行う。

3.1 機械学習について

筆記の改行を検出するために、我々は機械学習の一つである LSTM を用いる。ここでは、その説明を行う。

3.1.1 機械学習

機械学習は機械にデータを学習させることで、予測や分類、認識、データ処理などを行うための手法である。学習の方法として、大きく分けると「教師あり学習」と「教師なし学習」がある。教師あり学習は、データに対してラベルと呼ばれる正解を付けた正解データを学習させ、データを入力をしたときにその正解を出力するために学習を行う。主に予測や分類を行うために用いられる。教師なし学習は、正解データがなく、入力されたデータの構造や特徴などの本質的なものを見つけ出すための手法である。主にデータのグループ分けや圧縮のために用いられる。また、学習の方法とは別に、機械学習には決定木、ニューラルネットワーク、サポートベクターマシン、クラスタリングなど、様々な手法がある。

3.1.2 再帰ニューラルネットワーク

ニューラルネットワークの一つとして、再帰ニューラルネットワーク (RNN) があり、その派生に Long short-term memory (LSTM) がある [17]。ここでは、それらについての説明を行う。RNN はニューラルネットワークに再帰構造を追加したもので、直前のデータだけでなく、それよりも前のデータを考慮して結果を出力することができる。つまり、RNN

は時系列データを扱うことができる。RNN は文章の予測、手書き文字認識、音声認識などで主に用いられている。また、RNN の派生として LSTM がある。LSTM は RNN にメモリと 3 つのゲートを組み込むことで、RNN では難しいとされている長期的な時系列データの学習が可能になっている。本研究では、改行の検出をするために学習者の筆記データを用いる。これは学習者の筆記が時系列データであるため、LSTM を用いることが適していると考えられる。

3.2 LSTM の実装

LSTM のモデルの作成はニューラルネットワークライブラリである keras を用いる [18]。筆記の改行では、同じ行内での筆記と比較して、ストロークの x 座標と y 座標が大きく変化するため、LSTM の特徴としてストロークの x 座標と y 座標のデータを用いる。今回は、ストロークの始点をそのストロークの座標とした。LSTM に入力する際に、データの値が大きすぎると学習に時間がかかるため、データの平均を 0、標準偏差が 1 となるように正規化する標準化を行った。

LSTM の入力として、2 ストローク分の座標のデータを用いる。2 ストローク分のデータを入れ、そのストローク間で改行が行われているかを判断する。全てのストローク間で判断させるため、入力するストロークデータは 1 ストロークずつずらして入力する (図 3.1)。教師データは、筆記の座標データを用いる入力側のデータだけでなく、改行を判定する出力側のデータも必要となる。我々は、上述した改行の判定を基に、筆記データに対して改行の判定を行った。ストローク間で改行していた場合を「1」、改行していない場合を「0」として改行についての正解データを作成した (図 3.1)。

LSTM のパラメータについて、隠れ層の数を 50、バッチサイズを 256、学習のエポック数を 1500 とした。これらの値は、実装後に簡単なテストを行って決定した。その際、それぞれのパラメータを 1 つずつ変更させた場合は、結果に大きな変化は見られなかった。また、オプティマイザは Adam を使用した。keras には SGD, RMSprop, Adagrad など様々な種類のオプティマイザが用意されているが、汎用性が高いと言われる Adam を使用する。Adam 内のパラメータは keras のデフォルト値を使用した。

今回実装を行った LSTM は予測値として 0 から 1 の値を出力する。1 に近いほどそのストローク間で改行している可能性が高いことを示す。本研究では、改行をしているか

第4章 実験

4.1 筆記データの収集

我々は九州工業大学の学生9人に対してセンター試験の数学の大問を1つ解いてもらい、その時の筆記データを収集した。合計で6754個のストロークデータを収集できた(図4.1)。被験者をそれぞれNo.1からNo.9とする。筆記データの例を図4.2に示す。

ストロークNo.	No1,x座標	No1,y座標	No2,x座標	No2,y座標	No3,x座標	No3,y座標	No4,x座標	No4,y座標	No5,x座標	No5,y座標
0	66.25	107.4609	44.5	104.5	57.24958	104.4996	377.5	138.2496	618.9996	320.4293
1	34	126.2498	37	106.75	64.75	132.9998	391	126.9998	620.4996	348.9608
2	47.49958	126.2498	50.43828	108.25	44.5	114.2497	393.25	130.7109	45.25	75.99979
3	55.74958	120.9998	39.99958	135.2498	51.24958	116.4998	36.24958	51.99979	52.74958	78.24979
4	72.99958	123.9998	57.99958	144.9511	57.24958	114.2497	36.99958	44.49979	57.99958	76.74979
5	98.5	129.9996	60.21086	149.4997	84.24958	137.4996	31	46.75	71.49958	72.99979
6	98.49958	134.5	68.49958	143.5	84.99958	140.5	46.67928	42.99979	82.74958	78.99958
7	111.2099	129.25	76.74958	141.2498	93.25	137.4511	39.99958	71.49958	83.49958	80.5
8	117.25	129.2497	78.24958	136.75	99.24958	134.4998	37.75	69.99969	93.99958	74.5
9	125.5	120.25	90.24958	148.7496	103	127.75	49.74958	74.49979	94.70115	77.49972
10	128.4996	132.9511	94	145.75	114.25	135.25	48.99958	79.75	100.7496	71.49979
11	131.4996	129.9998	102.9996	145	116.4996	133.7498	63.25	74.5	105.2496	77.5
12	142	127.7065	112.75	148	126.25	129.25	66.96577	72.99979	107.5	75.24979
13	154.75	128.4565	121.7496	147.25	136	132.25	72.96086	68.49979	115.7109	73.75
14	166.7496	128.5	124	147.2497	145.7496	133	78.99958	75.25	121	72.96086
15	170.5	128.4996	136.7496	150.25	149.5	129.9997	82	73.74979	129.2496	71.5
16	180.9996	131.5	140.5	147.2497	162.9996	133	88.75	72.25	131.4609	70.74969
17	183.9996	126.9998	146.4996	144.25	165.9996	130.7498	97	70.74979	138.9996	72.24958
18	192.2251	124	157.75	148.7498	175.7109	126.2498	105.2099	70	144.2496	69.99979
19	202.75	127.75	163.7496	139	186.25	129.2497	110.5	66.99964	151.7496	66.99979
20	215.4609	122.4511	176.4609	152.4997	192.25	121.7011	121.7496	70.74958	163	69.99979

図 4.1: 座標データの一部

4.2 モデルの学習と評価

モデルの学習と評価について、leave-one-out 交差検証を行う。これは、データのうち1つを抜き出してテスト用データとし、残りを学習用データとする。これを全てのデータで1回ずつテストを行うように繰り返し行う方法である。今回、No.1 から No.9 まで9つのデータがあるので、9回繰り返し、それぞれの評価の平均を出すことで学習したモデルの評価を行う。

LSTM で学習したモデルがどれだけの性能で改行を検出できるかを評価するため、モデルが検出した改行と上述した改行の定義を基に我々が判断した正しい改行の比較を行う。9つの筆記データそれぞれに対して、精度と再現率を算出し、F 値を求める。それぞれの値を求める式は以下のようなになる。

$$\text{精度} = \frac{TP}{TP + FP}$$

$$\text{再現率} = \frac{TP}{TP + FN}$$

$$F = \frac{2 \times \text{精度} \times \text{再現率}}{\text{精度} + \text{再現率}}$$

それぞれ、TP はモデルが正しい改行を検出した数、FP はモデルが間違えて改行を検出した数、FN はモデルが正しい改行を検出できなかった数である。我々はモデルの正確さを、精度と再現率の調和平均である F 値で評価する。

4.3 結果

結果を表 4.1 に示す。表 4.1 より、F 値の平均は 0.811 なので、モデルは約 8 割の正確さで改行を検出することができた。

表 4.1: モデルの評価

No.	1	2	3	4	5	6	7	8	9	平均
TP	34	53	50	51	54	44	39	29	33	43
FP	6	15	1	3	5	8	4	0	7	5.44
FN	14	12	15	24	27	7	15	4	20	15.33
精度	0.850	0.779	0.980	0.944	0.915	0.846	0.907	1.000	0.825	0.894
再現率	0.708	0.815	0.769	0.680	0.667	0.863	0.722	0.879	0.623	0.747
F 値	0.773	0.797	0.862	0.791	0.771	0.854	0.804	0.935	0.710	0.811

4.4 考察

どのような箇所で誤検出をしているかを確認するために、筆記データに対して図 4.3 のように可視化を行った。その結果、次のことがわかった。

- 大きく左に移動するときに間違えて改行を検出すること (FP) が多い。
- 左下以外の方向への改行が検出できていないこと (FN) が多い。
- 図や表は不規則な動きをするため誤検出 (FP と FN) が多い。

誤検出の詳細を知るために、誤検出時の筆記の移動方向とそのときの平均予測値を表 4.2 にまとめた。その他の誤検出として、以下のものがあつた。FP について、大きく右へ移動、左上へ移動している場合、改行を検出してしまった。また、FN について、小さく左へ移動、大きく右上へ移動している場合、改行を検出できていなかった。

表 4.2 より、誤検出の個数について考える。FP に関して、左方向への移動のときの誤検出が多い。図を書く場合だけでなく、通常の数式を書いている場合においても、左に大きく戻ることがあり、そこで改行しているため多くなっている。FN に関して、下、左下方向への移動のときの誤検出が多い。これらは、移動量が比較的小さい場合に改行をうまく検出できないことが多い。

次に、予測値について考える。FN に関して、左方向 (左上と左下も含む) への移動のときに予測値が大きい。また、上と右方向については予測値が小さい。このことから、左方向を含まない場合、改行を検出しにくいと考えられる。

表 4.2: 誤検出時の移動方向と平均予測値 (上 3 つが FP, 下 7 つが FN を示す)

方向	個数	平均予測値
左	34	0.83
左下	4	0.87
下	5	0.80
左上	12	0.15
上	17	0.04
右上	19	0.10
右	3	0.03
右下	18	0.05
下	37	0.09
左下	30	0.17

$$(1) \begin{aligned} y &= x^2 + 2ax + 3a^2 - 6a - 36 \\ &= (x+a)^2 + 2a^2 - 6a - 36 \\ &\quad (-a, 2a^2 - 6a - 36) \end{aligned}$$

$$(2) \begin{aligned} 2a^2 - 6a - 36 &= \\ f(a) = 3a^2 - 6a - 36 &= -27 \\ 3a^2 - 6a - 9 &= 0 \\ a^2 - 2a - 3 &= 0 \\ (a+1)(a-3) &= 0 \\ a &= -1, 3 \end{aligned}$$

$$(3) \begin{aligned} A: y &= x^2 + 6x - 27 \\ B: y &= x^2 - 2x - 27 \\ A \text{ について} \\ y &= (x+3)^2 - 36 \\ B \text{ について} \\ y &= (x-1)^2 - 28 \\ \text{x軸方向} & \quad 4 \\ \text{y軸方向} & \quad -8 \end{aligned}$$

$$(4) \begin{aligned} D/4 &= a^2 - (3a^2 - 6a - 36) \\ &= -2a^2 + 6a + 36 \\ D/4 \geq 0 \text{ である} & \text{ 条件をいふと} \\ -2a^2 + 6a + 36 &\geq 0 \\ a^2 - 3a - 18 &\leq 0 \\ (a+3)(a-6) &\leq 0 \\ \therefore & \quad -3 \leq a \leq 6 \end{aligned}$$

$$(5) \begin{aligned} p &= 3a^2 - 6a - 36 \\ &= 3(a-1)^2 - 39 \\ -3 &\leq a \leq 6 \text{ かつ} \\ \text{最小値} & \quad -39 \quad (a=1) \\ \text{最大値} & \quad 36 \quad (a=6) \end{aligned}$$

$$(6) \begin{cases} D/4 \geq 0 & \text{--- ①} \\ \text{軸} x = -a > -1 & \text{--- ②} \\ f(-1) > 0 & \text{--- ③} \end{cases}$$

① について
 $(a)^2 \geq 1 \Rightarrow -3 \leq a \leq 6$

② について
 $-a > -1 \Rightarrow a < 1$

③ $f(-1) = 1 - 2a + 3a^2 - 6a - 36 > 0$
 $= 3a^2 - 8a - 35 > 0$
 $(3a+7)(a-5) > 0$
 $a < -\frac{7}{3}, 5 < a$

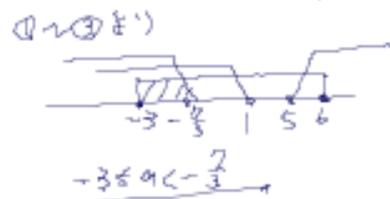


図 4.2: 筆記データの例

$$(1) y = x^2 + 2ax + 3a^2 - 6a - 36$$

$$= (x+a)^2 + 2a^2 - 6a - 36$$

$$(-a, 2a^2 - 6a - 36)$$

$$(2) \cancel{2a^2 - 6a - 36 =}$$

$$f(0) = 3a^2 - 6a - 36 = -27$$

$$3a^2 - 6a - 9 = 0$$

$$a^2 - 2a - 3 = 0$$

$$(a+1)(a-3) = 0$$

$$a = -1, 3$$

$$(3) A: y = x^2 + 6x - 27$$

$$B: y = x^2 - 2x - 27$$

Aに7117
 $y = (x+3)^2 - 36$

Bに7117
 $y = (x-1)^2 - 28$

x軸方向 4
y軸方向 -8

$$(4) D/4 = a^2 - (3a^2 - 6a - 36)$$

$$= -2a^2 + 6a + 36$$

$$D/4 \geq 0 \text{ である条件は } 7117$$

$$-2a^2 + 6a + 36 \geq 0$$

$$a^2 - 3a - 18 \leq 0$$

$$(a+3)(a-6) \leq 0$$

$$\therefore -3 \leq a \leq 6$$

$$(5) p = 3a^2 - 6a - 36$$

$$= 3(a-1)^2 - 39$$

$$-3 \leq a \leq 6 \text{ かつ } 1$$

最小値 -39 (a=1)
最大値 36 (a=6)

$$(6) \begin{cases} D/4 \geq 0 & \text{--- ①} \\ \text{軸 } x = -a > -1 & \text{--- ②} \\ f(-1) > 0 & \text{--- ③} \end{cases}$$

①に7117
(4) かつ $-3 \leq a \leq 6$

②に7117
 $-a > -1$
 $a < 1$

③ $f(-1) = 1 - 2a + 3a^2 - 6a - 36$
 $= 3a^2 - 8a - 35 > 0$
 $(3a+7)(a-5) > 0$
 $a < -\frac{7}{3}, 5 < a$

①~③ かつ

 $-3 \leq a < -\frac{7}{3}$

図 4.3: 改行ミスの可視化 (赤は FP, 青は FN を示す)

第5章 改善

筆記の改行検出について、実験結果や考察を元に、モデルのF値向上を試みる。

5.1 閾値の変更

予測値の多くは0や1に近い値となっているが、中には0.4や0.6などの予測値もある。これらのものに関しては、閾値を変更することで正しく検出することができ、F値が向上すると考えられる。また、閾値を変更したときの基本的な変化は、閾値を大きくするとFNの個数が多くなり、FPの個数が少なくなる。逆に、閾値を小さくするとFPの個数が多くなり、FNの個数が少なくなる。表4.1より、FNの個数がFPの個数と比較して多いので、閾値を小さくすることで、FNの数が少なくなって、F値が向上すると考えられる。ここでは、閾値を0.1から0.9まで0.1ごとに変更して各モデルのF値の変化を確認する。

結果は表5.1のようになった。表5.1より、半数以上のモデルは閾値を小さくすることでF値を大きくすることができた。また、それぞれの閾値の平均値を出すと、閾値が0.5のときにF値が最も大きくなった。

5.2 特徴の追加

改行時の筆記の移動距離が小さい場合、改行の検出がうまくできていない。筆記の移動距離は文字のサイズが大きい場合に大きくなり、文字のサイズが小さい場合に小さくなる傾向にあると考えられる。よって、文字のサイズをモデルの学習時の特徴として追加することで、モデルのF値をより向上させることができると考えられる。ここでは、文字のサイズを各ストロークのバウンディングボックスの幅と高さの2つに分けて追加した。バウンディングボックスとは、図5.1のように1つのストロークを囲むように作られ

表 5.1: 閾値変更による F 値の変化

No.	1	2	3	4	5	6	7	8	9	平均
0.1	0.804	0.772	0.903	0.784	0.768	0.796	0.804	0.882	0.656	0.797
0.2	0.784	0.791	0.885	0.794	0.792	0.833	0.788	0.937	0.686	0.810
0.3	0.769	0.791	0.883	0.785	0.803	0.830	0.792	0.935	0.680	0.808
0.4	0.778	0.779	0.864	0.782	0.775	0.838	0.808	0.935	0.702	0.807
0.5	0.773	0.797	0.862	0.791	0.771	0.854	0.804	0.935	0.710	0.811
0.6	0.744	0.794	0.852	0.781	0.759	0.851	0.753	0.935	0.652	0.791
0.7	0.756	0.756	0.821	0.752	0.750	0.804	0.769	0.935	0.636	0.776
0.8	0.759	0.732	0.800	0.733	0.727	0.813	0.727	0.881	0.578	0.750
0.9	0.720	0.710	0.687	0.606	0.586	0.747	0.675	0.881	0.423	0.671

た長方形の枠のことである。特徴を追加して、モデルの F 値を比較した。LSTM への入力は 1 つのストロークデータの中に [x 座標, y 座標, 幅, 高さ] の 4 個の特徴が入る。



図 5.1: ストロークの幅と高さ

結果は表 5.2 のようになった。予想に反し、多くのモデルで F 値が小さくなった。

表 5.2: 特徴の追加による F 値の変化

No.	1	2	3	4	5	6	7	8	9	平均
TP	34	48	43	48	43	41	36	24	32	38.78
FP	3	11	1	4	5	4	5	1	11	5
FN	14	17	22	27	38	10	18	9	21	19.56
精度	0.919	0.814	0.977	0.923	0.896	0.911	0.878	0.960	0.744	0.891
再現率	0.708	0.738	0.662	0.640	0.531	0.804	0.667	0.727	0.604	0.676
F 値	0.800	0.774	0.789	0.756	0.667	0.854	0.758	0.828	0.667	0.766

5.3 データ拡張

左方向以外への移動での改行検出がうまくいかない場合が多い。それを改善するために、データ拡張を行う。データ拡張とは、主に画像処理を行う際に用いられている、集めた学習用データを様々な方法で変換して新たなデータを増やす方法である。変換方法の例として、ノイズを加える、拡大縮小、反転、回転などの方法がある。

今回、データ拡張として、データの上下反転、左右反転、上下左右反転を行い、3種類のデータを新たに作成する。収集した筆記データについて、x座標の範囲が0から700(右方向が正)、y座標の範囲が0から990(下方向が正)なので、左右反転の場合、y座標は元のままでx座標を $700 - (\text{元の } x \text{ 座標})$ とし、上下反転の場合、x座標は元のままでy座標を $990 - (\text{元の } y \text{ 座標})$ とした。上下反転の場合はx座標を $700 - (\text{元の } x \text{ 座標})$ 、y座標を $990 - (\text{元の } y \text{ 座標})$ とした。元のデータに新しいデータを追加して、モデルの性能を比較する。3つの新しいデータをまずは1つずつ追加し、次に2つずつ追加し、最後に3つ全て追加する。

結果は表 5.3～表 5.9 のようになった。平均値を見ると、1つ追加したときは上下反転データを追加したときが良く、2つ追加したときは上下反転データと左右反転データの2つを追加したときが良かった。全体で見ると3つ追加したものが最も良かった。

表 5.3: 左右反転データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	35	54	52	53	61	44	45	30	28	44.66
FP	6	18	1	6	5	6	6	1	10	6.55
FN	13	11	13	22	20	7	9	3	25	13.66
精度	0.854	0.750	0.981	0.898	0.924	0.880	0.882	0.968	0.737	0.875
再現率	0.729	0.831	0.800	0.707	0.753	0.863	0.833	0.909	0.528	0.773
F 値	0.787	0.788	0.881	0.791	0.830	0.871	0.857	0.938	0.615	0.818

表 5.4: 上下反転データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	37	53	58	52	54	42	46	30	35	45.22
FP	8	13	2	7	11	7	9	0	10	7.44
FN	11	12	7	23	27	9	8	3	18	13.11
精度	0.822	0.803	0.967	0.881	0.831	0.857	0.836	1.000	0.778	0.864
再現率	0.771	0.815	0.892	0.693	0.667	0.824	0.852	0.909	0.660	0.787
F 値	0.796	0.809	0.928	0.776	0.740	0.840	0.844	0.952	0.714	0.822

5.4 考察

5.4.1 改善後の結果について

閾値を変更した場合について、予想していたように、閾値を小さくすることで半数以上のモデルで F 値を向上させることができたが、大きな変化はなく、それぞれの閾値における平均値は閾値が 0.5 のときが最も大きくなった。これは、閾値が 0.1 や 0.2 などのときに F 値が向上するものがある一方で、閾値が小さくなりすぎているため、他の筆記に対して正確な改行検出ができないからだと考えられる。

特徴を追加した場合について、結果は予想に反し、F 値が小さくなった。特徴を追加する前と比較すると、大きいストロークを中心に変化があった。良くなった箇所は主に、答えに下線を引いてる箇所とグラフや表などの大きいストロークを含む箇所である。答

表 5.5: 上下左右反転データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	35	53	57	53	60	43	40	29	29	44.33
FP	8	17	3	3	4	7	10	0	8	6.66
FN	13	12	8	22	21	8	14	4	24	14.00
精度	0.814	0.757	0.950	0.946	0.938	0.860	0.800	1.000	0.784	0.872
再現率	0.729	0.815	0.877	0.707	0.741	0.843	0.741	0.879	0.547	0.764
F 値	0.769	0.785	0.912	0.809	0.828	0.851	0.769	0.935	0.644	0.812

表 5.6: 左右反転データと上下反転データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	37	54	56	53	61	45	45	31	41	47.00
FP	7	17	2	8	7	7	6	0	16	7.77
FN	11	11	9	22	20	6	9	2	12	11.33
精度	0.841	0.761	0.966	0.869	0.897	0.865	0.882	1.000	0.719	0.867
再現率	0.771	0.831	0.862	0.707	0.753	0.882	0.833	0.939	0.774	0.817
F 値	0.804	0.794	0.911	0.779	0.819	0.874	0.857	0.969	0.745	0.839

えの下線部分 FP が 9 個から 3 個まで減った。残りの 3 個も予測値を見ると小さくなっていた。また、グラフや表などの FP の個数も減っていた。しかし、逆に悪くなった箇所もあり、FN が多くなっている。間違えた式を消去するために書く消去線や、通常の筆記の中でも若干長いストロークなどで改行を検出できないことがあった。特徴として文字サイズを追加することで、大きいストロークを中心に変化が見られたが、今回の結果より、それぞれのストロークの幅と高さ自体は改行を判断するための特徴にはなりにくいと考えられる。筆記の移動距離に直接関係するような特徴を設定することが必要となるだろう。

データ拡張として、左右反転データ、上下反転データ、上下左右反転データをそれぞれ追加した場合について、改行検出にどのような変化があるのか、データ追加前の結果と比較して確認する。左右反転データを追加した場合、良くなった点として、右方向へ筆

表 5.7: 左右反転データと上下左右反転データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	34	53	53	52	64	47	44	29	37	45.89
FP	8	20	1	6	3	7	6	2	16	7.67
FN	14	12	12	23	17	4	10	4	16	12.44
精度	0.810	0.726	0.981	0.897	0.955	0.870	0.880	0.935	0.698	0.861
再現率	0.708	0.815	0.815	0.693	0.790	0.922	0.815	0.879	0.698	0.793
F 値	0.756	0.768	0.891	0.782	0.865	0.895	0.846	0.906	0.698	0.823

表 5.8: 上下反転データと上下左右反転データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	38	56	54	58	58	43	43	30	33	45.89
FP	8	19	1	6	7	7	5	0	11	7.11
FN	10	9	11	17	23	8	11	3	20	12.44
精度	0.826	0.747	0.982	0.906	0.892	0.860	0.896	1.000	0.750	0.873
再現率	0.792	0.862	0.831	0.773	0.716	0.843	0.796	0.909	0.623	0.794
F 値	0.809	0.800	0.900	0.835	0.795	0.851	0.843	0.952	0.680	0.829

記が移動したとき改行を検出できなかった箇所が正しく検出できることが増え、左方向へ筆記が移動したとき余計な箇所での改行を検出することが減った。逆に、悪くなった点として、左方向へ筆記が移動したとき正しい改行を検出できないことが増え、右方向へ筆記が移動したときに余計な改行の検出が増えた。つまり、追加前は左方向へ移動したときに改行だと判断することが多かったが、左右反転データを追加することで右方向へ移動した場合にも改行だと判断するようになった。同様に、上下反転データを追加した場合は、上方向、主に左上方向へ移動したときの改行の検出精度が高まり、上下左右反転データを追加した場合は、右上方向へ移動したときの改行の検出精度が高まった。3つの拡張データを全て追加した場合、それぞれ1つずつ追加したときの結果を合わせたような結果となった。1つずつ追加したときは変わらず、3つ追加したときのみ変化した箇所として、左下に筆記が移動したときに改行を検出できないことが増えた。これは、左

表 5.9: 3 つの拡張データを追加

No.	1	2	3	4	5	6	7	8	9	平均
TP	38	48	58	54	65	46	44	30	36	46.55
FP	8	14	1	6	8	6	3	0	13	6.55
FN	10	17	7	21	16	5	10	3	17	11.77
精度	0.826	0.774	0.983	0.900	0.890	0.885	0.936	1.000	0.735	0.881
再現率	0.792	0.738	0.892	0.720	0.802	0.902	0.815	0.909	0.679	0.806
F 値	0.809	0.756	0.935	0.800	0.844	0.893	0.871	0.952	0.706	0.841

右反転データが右下方向，上下反転データが左上方向，上下左右反転データが右上方向への移動に対して改行をより検出するようになったため，3 つすべて追加したときは，1 つずつ追加したときよりも左下方向への移動による改行の検出ができなくなっていると考えられる。

5.4.2 関連研究との比較

図やグラフの除去を行わず，2 ストローク間の移動距離を用いて横一行の認識を行っているため，手法が類似していると考えられる前田の研究 [7] との比較を行う。前田は横一行の認識率を評価指標としているため，我々の実験結果から横一行の認識率を以下の式より算出する。

$$\text{認識率} = \frac{TP}{TP + FP + FN}$$

我々の実験で最も F 値の大きい表 5.9 のデータを用いると表 5.10 のようになった。また，結果を比較するために我々の収集した筆記データに対して前田の手法を用いて横一行の認識率を算出した。前田の研究では 2 ストローク間の移動距離が 20mm 以上のときに改行と判断する場合は最も認識率が良かったため，我々も座標の移動距離が 20 以上で横一行の認識をしたところ，認識率が 40% 程度で明らかに精度が悪かった。これは，我々の座標における 20 の距離が 20mm ではないと考えられるため，移動距離の閾値を 10 ずつ変えて平均認識率が最大となる値を探した。閾値を変えていった結果，表 5.10 のようになり，閾値が 50 のときに平均認識率が最も高い 67.9% となった。

表 5.10: 横一行の認識率

No.	1	2	3	4	5	6	7	8	9	平均
提案手法	67.86	60.76	87.88	66.67	73.03	80.70	77.19	90.91	54.55	73.28
前田の手法	57.58	57.14	83.10	71.76	73.47	67.61	67.12	81.08	52.27	67.90

表 5.10 の結果より，有意水準 5% で t 検定を行ったところ， $t(16) = 2.44$ ， $p = 0.026$ となり，有意差が認められた．つまり，LSTM を用いることで，筆記の移動距離と閾値を用いる手法と比べて認識精度を向上させることを確認した．

第6章 結論

本章では、これまでの結果をまとめ、今後の課題を述べる。

6.1 まとめ

本論文では、筆記を一行ずつに自動で分割するために筆記の改行を検出する手法を提案し、その評価を行った。LSTM で学習したモデルが検出した改行を、実際に人が改行と判断したものと比較したところ、F 値は 0.81 であった。また、データ拡張を行い、学習データを増やすことで、F 値を 0.84 まで向上させることができた。この結果を関連研究と比較するために t 検定を行ったところ、有意水準 5% で有意に高いという結果が得られた。

6.2 今後の展望

ここでは、改行検出の精度を向上させるための方法や、改行検出の利用方法について考える。

改行検出の精度を向上させるために、他の特徴の追加や、データクレンジング、グリッドサーチなどが考えられる。特徴として文字の大きさを追加したとき、F 値を見ると小さくなっていたため、全体としては良くなかったが、データ拡張を行ったときには正確に検出できなかった箇所を検出することができていた。今回の場合は特徴が局所的に有効だったと考えられるため、汎用性の高い特徴を追加すれば精度は向上すると考えられる。また、データクレンジングは学習データの前処理のことで、今回の場合だと、筆記の図や表、細かい計算などは除去してから検出を行った方が正しい改行検出を行うことができたと考えられる。グリッドサーチについて、今回の LSTM のパラメータは、1 つずつのパラメータを変化させて結果の変化を見る簡単なテストしか行っていない。複数

のパラメータを同時に変更させて、適切なパラメータを発見するグリッドサーチを行うことで、より適切なパラメータを設定することができたと考えられる。

改行検出の利用について、筆記を一行ずつに分割することで、文字認識や一行ごとのデータの収集など、その後の筆記の処理が容易になると考えられる。また、今回は時系列を意識したオンライン筆記に対して改行検出を行った。オフライン筆記の改行検出では考慮することが難しい、学習者が解答時に戻った箇所を判断できるため、学習者の迷いなどを判断することができる可能性がある。

謝辞

本論文を完成するにあたり、ご指導ご教授くださりました三浦准教授に御礼申し上げます。また、サーベイ輪講や中間発表においてご指導ご教授くださりました情報セクションの先生方、学生の皆様に御礼申し上げます。加えて、本論文のデータ収集や評価実験において、被験者としてご参加頂きました九州工業大学の学生の皆様、研究のみならず様々なことでアドバイスをしてくださった三浦研究室の皆様に感謝いたします。最後に、大学生活を送るにあたって経済面や生活面において、ご支援をして頂いた家族に心から感謝申し上げます。

参考文献

- [1] 文部科学省. 教育の情報化ビジョン～21世紀にふさわしい学びと学校の創造を目指して～. http://www.mext.go.jp/b_menu/houdou/23/04/_icsFiles/afieldfile/2011/04/28/1305484_01_1.pdf, 2011.
- [2] 2020年代に向けた教育の情報化に関する懇談会. 「2020年代に向けた教育の情報化に関する懇談会」最終まとめ. http://www.mext.go.jp/b_menu/houdou/28/07/_icsFiles/afieldfile/2016/07/29/1375100_01_1_1.pdf, 2016.
- [3] 三浦元喜, 杉原太郎, 國藤進. 一般教室での日常的利用を考慮したデジタルペン授業システムの改良. 日本教育工学会論文誌 34(3), pp. 279–287, 2010.
- [4] 内田誠一, Marcus Liwicki, 岩村雅一, 大町真一郎, 黄瀬浩一. デジタルペン. 映像情報メディア学会誌 Vol.64, No.3, pp. 293–298, 2010.
- [5] 浅井洋樹. デジタル手書きの情報処理 —目に見えない手書きデータに眠る新たな可能性を切り開く—. 情報処理, Vol.57, No.8, pp. 744–747, 2016.
- [6] 岸兼也. ペンストロークの時間差を用いた学習者の弱点検出. 九州工業大学工学部電気電子工学科システムエレクトロニクスコース卒業論文, 2017.
- [7] 前田祐樹. 多人数学習者筆記の同時閲覧手法に関する研究. 九州工業大学大学院工学府先端機能システム工学専攻修士論文, 2018.
- [8] 飯山将晃, 中塚智尋, 森村吉貴, 橋本敦史, 村上正行, 美濃導彦. ペンストロークの時間間隔を用いた解答停滞箇所の検出. 教育システム情報学会誌, Vol. 34, No. 2, pp. 166–171, 2017.

- [9] 浅井洋樹, 山名早人. 電子ペンを利用した数学手書き答案の戦略分類手法～多項式展開問題を題材として～. 研究報告コンピュータと教育 (CE), Vol. 2016, No. 6, pp. 1–8, 2016.
- [10] 大西正輝, 泉正夫, 福永邦雄. 講義映像における板書領域のブロック分割とその応用. 電子情報通信学会論文誌 D, Vol. 83, No. 11, pp. 1187–1195, 2000.
- [11] 工藤良, 三浦元喜. デジタルペンの筆記特性を利用したデジタルノート再構成支援. 情報処理学会インタラクシオン 2012 シンポジウム, pp. 229–234, 2012.
- [12] 待井君吉, 中川正樹. オンライン手書き紙面パターンにおける文字, 図形分離の一手法. 情報処理学会論文誌, Vol. 37, No. 4, pp. 490–499, 1996.
- [13] Truyen Van Phan and Masaki Nakagawa. Text/non-text classification in online handwritten documents with recurrent neural networks. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pp. 23–28. IEEE, 2014.
- [14] Dieu Ni Tran, Tuan Anh Tran, Aran Oh, Soo Hyung Kim, and In Seop Na. Table detection from document image using vertical arrangement of text blocks. *International Journal of Contents*, Vol. 11, No. 4, pp. 77–85, 2015.
- [15] 後藤克成. 手書き答案用紙画像からの行切り出し手法の高速化. 三重大学大学院地域イノベーション学研究所博士前期課程地域イノベーション学専攻修士論文, 2014.
- [16] Yi Li, Yefeng Zheng, and David Doermann. Detecting text lines in handwritten documents. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, Vol. 2, pp. 1030–1033. IEEE, 2006.
- [17] 斎藤康毅. ゼロから作る Deep Learning2 —自然言語処理編. オライリージャパン, 2018.
- [18] Keras: Python の深層学習ライブラリ. <https://keras.io/ja/>.

对外発表リスト

- Kazuya Kishi, Motoki Miura: Detecting Learners' Weak Points Utilizing a Digital Pen, 6th International Conference on Learning Technologies and Learning Environments, Hamamatsu, pp. 762-767, July 2017.
- Kazuya Kishi, Motoki Miura: Detecting Learners' Weak Points Utilizing Time Intervals of Pen Strokes, International Journal of Learning Technologies and Learning Environments, International Institute of Applied Informatics, Vol. 1, No. 1, pp. 61-77, March 2018.