領域分割済み画像化された文書の構造保持型情報抽出に関する研究

高峯茉優[1]　伊東桂佑[2]　鶴田直之[1]　乙武北斗[1]

概要：自治体が公開する会議資料を自然言語処理によって構造化し，データベース化する研究が進められている．本研究では，片田江ら(2026)の手法によって抽出された文書の領域分割情報を手掛かりとして画像化された文書から文章および表内容の抽出法を提案する．文章領域からは PaddleOCR を用いて文字検出・認識を行い，縦書き・横書き情報をもとに読み順ソートにより自然な読み順の文章を抽出する． 表領域に対しては，PP-StructureV3 を用いて構造解析を行い，罫線の欠落など構造解析に失敗した場合には，OCR で取得した文字位置情報を基に表の構造復元を補助する．実験では文章 97.3%，表 85.9%の F 値を得た．

キーワード：文書画像処理，構造保持型情報抽出，自治体公開文書

Research on Structure-Preserving Information Extraction from Segmented Image-Based Documents

MAYU TAKAMINE[†1], KEISUKE ITO[†2], NAOYUKI TSURUTA[†1], HOKUTO OTOTAKE [†1]

Abstract: Research is underway to structure and build databases of meeting materials published by local governments using natural language processing. This paper proposes a method for extracting text and table content from imaged documents, using the region segmentation information extracted by the method of Katada et al. (2026) as a guide. For text regions, character detection and recognition are performed using PaddleOCR. Based on vertical/horizontal writing information, text is extracted in a natural reading order through reading-order sorting. For table regions, structural analysis is performed using PP-StructureV3. When structural analysis fails due to missing borders, the method assists in restoring the table structure based on character position information obtained via OCR. Experiments yielded an F-score of 97.3% for text and 85.9% for tables.

Keywords: Document image processing, structure-preserving information extraction, local government public documents

1. はじめに

近年，地方分権の進展にもかかわらず，住民による地方自治への参加は十分に進んでいない．この課題を解決するためには，住民の地方自治への関心を高め，参加を促進する取り組みが求められている．その一環として，地方自治体が公開する会議資料を自然言語処理によって構造化し，データベース化する研究が進められている[1][2]. 多くの自治体は議会活動に関する資料を PDF 形式で公開しているが，PDF 文書はレイアウトやフォーマットの自由度が高いため，文字や語の順序を保持した正確なテキストデータの抽出は容易ではない．また，議会資料には表形式の情報が多く含まれており，セルの位置関係や階層構造を維持した抽出は一層困難である．

本研究では，先行研究[3]の手法（以下，前処理）により検出された領域分割の結果を活用し，PaddleOCR[4]および PP-StructureV3[5]を用いた文書構造を維持したテキストおよび表の抽出手法を提案する．なお，本研究が対象とする文書はページ全体が画像として保存されている PDF ファイルである．

2. システムの構成

本研究で構築した提案システムの全体像を図 1 に示し，各構成要素について以下で述べる．

(1) 前処理結果による処理の振り分け

本システムは，入力として PDF から抽出した画像データと，前処理により検出された領域分割結果（カテゴリ ID およびポリゴン座標を含む JSON ファイル）を受け取る．ここで，カテゴリは以下のように階層的に定義されている．なお，H (Horizontal)と V (Vertical) はそれぞれ横書き，縦書きを示す．

Page（ページ）

- PTitle（ページ内に一つだけ存在する大見出し）
- PSegment（タイトル、文章、リード文を持つかたま

1 福岡大学
Fukuoka University
2 福岡大学大学院
Fukuoka University Graduate School

り）

- TitleV（PSegment 内に 1 つだけのタイトル、見出し文）
- TitleH
- LeadV（Title,Paragraph を補足する文、リード文）
- LeadH
- ParagraphV（文章、本文、箇条書きの文
- ParagraphH

• FSegment（図や表とその説明文のかたまり）

- Figure（イラスト、挿絵、写真）
- Table（表、罫線で囲まれた文字や数字 ※箇条書きは含まない）
- CaptionV（図や表を簡潔に説明する 1～2 行の文）
- CaptionH

本研究では, FSegment 内の Figure 以外の全ての文字領域を対象とする．領域分割結果の入力後，まずカテゴリ ID に基づいて処理を文章領域の抽出処理と表領域の抽出処理の二つに分岐する. 最終的に, 表領域からは表データ（.xlsx），テキスト領域からはテキストファイル（.txt）が生成される．本システムにより, 複雑なレイアウト構造を持つ文書から，テキストと表を適切に分離し，それぞれの形式に適した形で高精度に抽出することが可能となる．

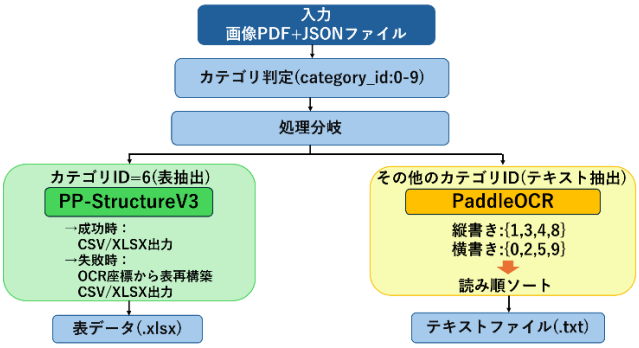

図 1　提案システムの処理フロー

Figure1 Process Flow of the Proposal System

(2) 文章領域の抽出処理

文章領域に対しては，PaddleOCR によって文字検出・認識を行い，縦書き・横書きのカテゴリに応じて読み順ソートを適用する．これにより縦書き・横書きが混在する議会だよりにおいても，文章の自然な読み順を保持したテキストデータ（.txt）として出力が可能となる．

(3) 表領域の抽出処理

表領域に対しては，PP-StructureV3 を用いることでセル境界の抽出および階層構造の解析を行い，表の内容を構造化データ（CSV/XLSX）として出力する．また，罫線の欠落などによって構造解析が失敗した場合には，OCR で取得した文字位置情報を基にセルの再構築を行うことで，表の構造復元を補助する．

3. 文章抽出アルゴリズム

3.1 マスク処理による関心領域（ROI）外の白背景化

文書領域の分割結果を使用するため，対象のポリゴン領域のみを抽出し，範囲外を白背景化する処理を実装した．この処理により，隣接する他の段落や図解が OCR の認識範囲に混入することを防止し，OCR エンジンが対象領域のみに注視できる環境を構築する．

具体的には，

1. ポリゴン座標からマスク画像を生成し，ポリゴン内を白，範囲外を黒として二値化する．
2. その後，元画像とマスクを合成することで，ポリゴン外を白背景とした画像を生成する．

この処理により，後述の OCR および表抽出処理の精度向上を図る．

図 2　マスク処理の例

Figure2 An example of mask processing

3.2 文章領域の抽出処理

(1) 縦・横書きの判別

議会だよりは日本語特有の縦書き・横書きが混在する文書であり，OCR の読み順が乱れやすい．本システムでは，前処理の結果を用いて，適切な並べ替え処理を切り替える設計とした．この判定により，後述する座標ベースのソート処理を最適化する．

(2) 正規化処理

本システムでは，OCR によって抽出された文字列を正確に評価するため，文字列に対して正規化処理を施した．OCR 出力には，全角・半角の混在や，不要なスペースの挿入といったノイズが多く含まれる．これらは文字そのものの認識性能とは無関係の要因であり，そのまま評価に用いると実際の性能を不当に低く見積もる可能性がある．この問題を防止するため, 本システムでは以下の 2 種類の正規化処理を行った．

1. Unicode 正規化（NFKC）

まず，Unicode 正規化（NFKC: Normalization Form Compatibility Composition）を適用し，字体や互換文字の揺れを統一した．NFKC は, 外見がほぼ同一であるにもかかわらず別コードとして扱われる文字（例：全角数字「１２３」と半角数字「123」, 濁点の合成形「が」と分解形「か゛」など）を一貫した表現へ変換する．

これにより，データ上は異なるコードポイントで表されている文字であっても，同一文字として扱うことが可能となり，OCR の認識性能とは関係のない差異を排除できる．

2. 不要スペースの削除

次に，OCR により意図せず挿入される全角・半角スペースをすべて除去する．文書レイアウト（段組，縦書き，改行位置など）に依存して生じるスペースは，評価対象とするテキストそのものの情報とは無関係である．これらの空白をそのまま評価に含めると，単語境界の誤認識などとは無関係な理由でエラー数が増加し，性能評価を過小に見積もってしまう．

(3) 読み順ソート処理

PaddleOCR は，文字列の認識結果を返す．しかし，この認識された文字列は検出された順序のまま列挙されており，必ずしも文書の自然な読み順と一致しない．そのため，本システムでは文字列の認識結果とともに返される，検出した文字列の位置座標（バウンディングボックス）を基に行や列のまとまりを判断し，適切な読み順に再構成するアルゴリズムを実装した．

1. 縦書きの場合

縦書き文書では，右から左の順に読むのが自然である．各文字列の X 座標（最大 X 座標）を基準に，X 座標の降順でソートすることで，縦書き特有の右から左への読み順を再現する．

2. 横書きの場合

横書き文書では，上から下の順に読むのが自然である．各文字列の Y 座標（最小 Y 座標）を基準に上から下へソートすることで，文書の自然な読み順を実現する．

以上の処理により，OCR が本来の文書構造とは異なる順序で文字列を出力した場合であっても，文書の自然な読み順に合致したテキスト抽出を実現する．

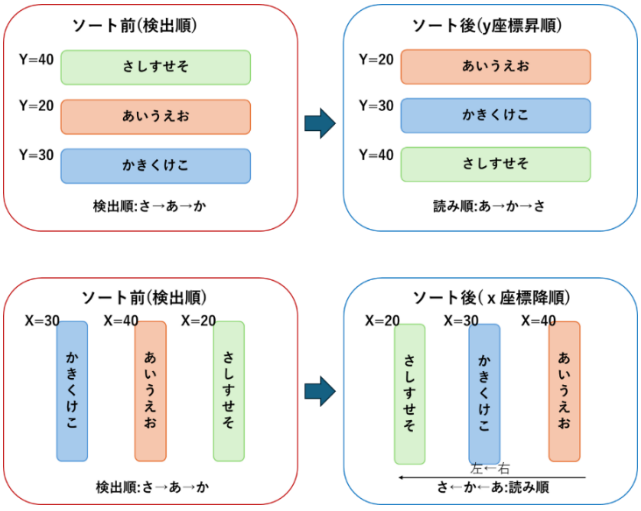

図 3　ソート処理の概念

Figure3 Concept of Sorting

4. 表領域の抽出処理

4.1 ROI(関心領域)切り出し

テキスト領域については画像全体に対して OCR を適用するが，表領域については以下の理由から ROI（Region of Interest: 関心領域）切り出しを行う．

表内のセルに記載される文字は，通常の本文テキストと比較してフォントサイズが小さく，文字間隔も狭い．このような小さな文字を含む表を，余白を含む画像全体に対して OCR すると，相対的に文字サイズがさらに小さくなり，認識精度が低下する可能性がある．そこで，表領域のみを切り出した ROI 画像を生成し，表部分を拡大した状態で OCR を適用することで，文字認識の精度を向上させる．

具体的には，マスク処理された画像から，ポリゴンの最小外接矩形を計算し，10 ピクセルのパディングを追加して ROI 画像を切り出す．このパディングは，表の境界が画像端に密接している場合に PP-StructureV3 が境界線を誤検出することを防ぐために必要である．

切り出された ROI 画像に対して PP-StructureV3 を適用することで，セルの結合状態を含んだ高度な構造化データを HTML 形式で出力し，これを CSV/XLSX 形式に変換する．

4.2 抽出失敗時の表再構築

罫線のない表等，構造認識が困難な表の場合では，PP-StructureV3 が失敗するケースがある．このような場合に対処するため，本システムでは OCR 座標から表構造を推定する再構築アルゴリズムを実装した．

再構築処理では，まず PaddleOCR が返す認識済み文字の位置座標を利用する．バウンディングボックスの高さ（max_y - min_y）の中央値を算出し，その 0.6 倍を行分割の閾値として設定する．この閾値を用いて，Y 座標の差に基づいて文字を行単位でクラスタリングする． 次に，各行内の文字を X 座標でソートし，列方向に整形する．最後に，行と列の情報を基に表形式のデータフレームを構築し，CSV/XLSX 形式で出力する．

この 2 段階のアプローチにより，構造解析に成功すれば構造を活かし，失敗しても座標から再構築する，表抽出システムを実現した．

5. 実験

提案システムが，複雑なレイアウトにおいてどれほど正確に文字抽出と読み順整合，表構造を抽出可能か検証する．なお，前処理の性能は適合率 84.33%，再現率 77.35%，F 値 80.54%である．本実験での性能評価では，前処理が正しく行われている領域のみを対象とした．

5.1 実験環境

- 開発環境: Visual Studio Code
- Python 3.12.3
- PaddleOCR: 3.2.0
- PP-StructureV3（PaddleOCR 3.2.0 に付属）
- Pillow (PIL): 10.4.0
- NumPy: 1.26.4
- pandas: 2.2.3
- BeautifulSoup4: 4.13.4 (HTML 解析用)
- openpyxl: 3.1.5 (Excel 出力用)

5.2 実験対象

実験には，北九州市，みやま市，うきは市，筑後市，築上町，大野城市の議会だより 30 枚を使用した．対象文書は画像化された PDF 文書であり，縦書き，横書き，図表が混在する複雑なレイアウトを持つ．評価方法は，文章領域と表領域で異なる．

まず，文章領域の評価指標は，文字列の不一致を詳細に分析するため，レーベンシュタイン距離（編集距離）[6]に基づき，以下の 3 つの操作数を計数する．

- 置換 (Substitution, S): 正解と異なる文字が抽出された数
- 挿入 (Insertion, I): 正解にない余分な文字が抽出された数
- 欠落 (Deletion, D): 正解にある文字が抽出されなかった数

レーベンシュタイン距離（編集距離）とは，二つの文字列がどの程度異なっているかを定量化する指標であり，一つの文字列をもう一つの文字列に変換するために必要な最小の編集操作回数を指す．本研究では，正解データ（GT）と OCR 抽出結果の乖離を詳細に分析するために採用している．例えば（図 4 参照），正解が「コミュニティ」で，OCR 結果が「コミュニテ」（「ィ」が欠落）となった場合，編集距離は 1（欠落 1 回）となる．

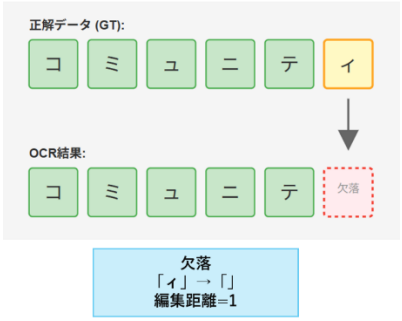

図 4 レーベンシュタイン距離の例
Figure4 An example of the Levenshtein distance

前述の操作数に基づき，以下の 4 つの指標を定義する．ここで，N を正解文（GT）の総文字数，M を OCR による抽出結果の総文字数とする．

- CER (Character Error Rate: 文字エラー率)：文字単位のエラーの割合を示す．

$$CER = \frac{S + I + D}{N}$$

- Recall (再現率)：未検出の少なさを表わす．

$$Recall = \frac{N - (S + D)}{N}$$

- Precision (適合率)：誤検出の少なさを表わす．

$$Precision = \frac{M - (S + I)}{M}$$

- F-score (F 値)：Recall と Precision の調和平均．誤検出，未検出両方を考慮．

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

次に，表領域の評価では，表を構成するセル構造の正しさと，各セルに含まれる文字認識の正確さの両方を評価する必要がある．表は単なる文字列の集まりではなく，行・列のレイアウトやセルの分割構造を含むため，通常の文字認識指標（CER など）だけでは評価が不十分である．そのため本研究では，以下の 3 種類を用いて表領域の性能を総合的に評価する．

- セル完全一致率：セル構造が正しく復元され（構造一致），かつセル内の文字列が GT と完全一致しているセルの割合

$$\text{セル完全一致率} = \frac{\text{構造一致かつ文字が完全一致したセル数}}{\text{GT の総セル数}}$$

- セル内文字の評価：セル内の文字を評価するために CER，Recall，Precision，F-score を用いる．これらはテキスト抽出の評価と同様に，レーベンシュタイン距離に基づく 置換（S）・挿入（I）・欠落（D）の 3 種類の誤りから算出する．ここで，N は正解文字数， M は抽出文字数である．

$$CER = \frac{S + I + D}{N}$$

$$Recall = \frac{N - (S + D)}{N}$$

$$Precision = \frac{M - (S + I)}{M}$$

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- セル構造の評価：セル構造が正しく検出されているかどうかを評価するため，Structure Recall（構造の再現率）と Structure Precision（構造の適合率），Structure F-score（調和平均）の 3 つの指標を用いる．

$$Recall = \frac{\text{正しく予測したセル数}}{\text{実際に予測すべきセル数}}$$

$$Precision = \frac{\text{正しく予測したセル数}}{\text{予測したセル数}}$$

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

5.3 実験 1:テキスト抽出性能検証

実験 1 では，テキスト領域のみを対象とし，PaddleOCR による文字認識精度を検証する． 縦書き・横書きに応じた読み順ソートを行った OCR 出力と正解データ（GT）を比較する．テキスト抽出の評価結果を表 1 に示す．

文字エラー率は 4.44%，F 値は 97.28%となり，高精度な

テキスト抽出が可能であることを確認した．特に適合率が99.01%と高く，誤検出が少ないことが示された．一方で，再現率（R）が 95.61% とやや低いのは，欠落（D）が 762 文字と比較的多く，全体のエラー数の約 79%であったことに起因している．特に，以下の種類の文字で欠落が集中する傾向が確認された．

- 句読点（「,」「.」）
- 括弧類（「(」「)」「＜」「＞」など）
- 記号（「※」「●」「・」「−」等）
- 拗音・促音などの小さい文字（「っ」「ッ」「や」「ゅ」「ょ」等）
- 長音符（「ー」）

これらは画素数が少なく，背景ノイズの影響を受けやすいことから，OCR が検出に失敗しやすいと考えられる．
しかしながら，欠落文字は「意味上の補助的役割を担う文字種」に偏っており，文章全体の意味に大きな影響を与えにくい．一方で，文書の完全復元という観点では改善の余地が残るといえる．

表 1 実験 1 の結果

Table1 Results of Experiment 1

文字数(N)	21825
抽出文字数(M)	21074
置換(S)	197
挿入(I)	11
欠落(D)	762
文字エラー率(CER)	4.44%
Recall	95.61%
Precision	99.01%
F−score	97.28%

5.4 実験 2:表抽出性能検証

実験 2 では，表領域のみを対象とし，PP-StructureV3 と構造解析が失敗した場合にの OCR 座標情報に基づく再構築処理の精度を評価する． 表抽出の評価結果を表 2 と表 3 に示す．構造レベルの再現率，適合率，F 値がいずれも 100%となり，提案システムが表のセル構造を完全に復元できることを実証した．これは，PP-StructureV3 による構造解析と，失敗時の OCR 座標からの再構築を組み合わせたフォールバック機構が有効に機能したことを示している．

表 2 実験 2（構造解析）の結果

Table2 Results of Experiment 2 (Structural Analysis)

GTセルの個数	953
OCRセルの個数	953
セル一致の個数	953
文字も一致したセルの個数	449
セル完全一致率	47.11%
構造R	100.00%
構造P	100.00%
構造F	100.00%

一方，セル完全一致率は 47.11%，文字エラー率は 20.14%となり，文字レベルの認識精度には改善の余地がある．文字 F 値は 85.89%であり，テキスト抽出（97.28%）と比較して低い値となった．

表 3 実験 2（セル内の文字）の結果

Table3 Results of Experiment 2 (Text in Cells)

正解文字数(N)	4732
抽出文字数(M)	4270
置換(S)	317
挿入(I)	87
欠落(D)	549
CER	20.14%
文字R	81.70%
文字P	90.54%
文字F	85.89%

誤認識の内訳を分析したところ，置換や欠落が増える以下の傾向が確認された．

- 小さい文字サイズのセル
- 数値・記号を含むセル
- 複数行が詰まったセル

表には数値や記号が多く含まれ，OCR が正確に文字を認識できず誤認識したことや，表領域内の文字サイズは本文より小さいことが多く，OCR 認識が困難であるということが考えられる．

5.5 実験 3: 画像全体を対象とした手法との比較実験

実験 3 では，提案手法との性能差を定量的に評価するため，領域分割を行わず画像全体をそのまま PaddleOCR および PP-StructureV3 に投入し，画像全体に対しテキスト抽出と表抽出を実行する．この比較により，提案システムがどの程度性能向上をもたらしているかを評価する．

まず，実験 1（テキスト抽出）との比較結果を表 4 に示す．実験 3 では，置換(S)が 10,866 件，挿入(I)が 1,106 件，欠落(D)が 2,157 件発生し，文字エラー率(CER)は 64.02%，F 値は 41.99%となった．実験 1 と比較して大幅に性能が低下した．具体的には，F 値が 55.29%低下し，CER が 59.58%悪化する結果となった．

表 4 実験 1 と実験 3 の比較

Table4 Comparison of Experiment 1 and Experiment 3

	実験1	実験3
文字数(N)	21825	22069
抽出文字数(M)	21074	21018
置換(S)	197	10866
挿入(I)	11	1106
欠落(D)	762	2157
文字エラー率(CER)	4.44%	64.02%
R	95.61%	40.99%
P	99.01%	43.04%
F	97.28%	41.99%

次に，実験 2(表抽出)との比較結果を表 5 と表 6 に示す．

実験 3(全体 OCR)では 877 個のセルが OCR 検出され，セル一致の個数も 877 個であったが，実験 2 では 953 個すべてのセルを完全検出した．文字も一致したセルの個数は，実験 2 が 449 個，実験 3 が 408 個であった．構造 F 値については，実験 3 が 95.85%，実験 2 が 100.00%を達成し，実験 3 は 4.15%の低下となった. 文字 F 値は, 実験 3 が 79.75%，実験 2 が 85.89%であり，6.14%の低下が見られた．セル完全一致率は実験 3 が 42.81%，実験 2 が 47.11%となり，CER は実験 3 が 29.56%，実験 2 が 20.14%であった．

表 5 実験 2 と実験 3（構造解析）の比較
Table5 Comparison of Experiment 2 and Experiment 3 (Structural Analysis)

	実験2	実験3
GTセルの個数	953	953
OCRセルの個数	953	877
セル一致の個数	953	877
文字も一致したセルの個数	449	408
セル完全一致率	47.11%	42.81%
構造R	100.00%	92.03%
構造P	100.00%	100.00%
構造F	100.00%	95.85%

表 6 実験 2 と実験 3（セル内の文字）の比較
Table6 Comparison of Experiment 2 and Experiment 3 (Text in Cells)

	実験 2	実験 3
正解文字数(N)	4732	4732
抽出文字数(M)	4270	3892
置換(S)	317	347
挿入(I)	87	106
欠落(D)	549	946
CER	20.14%	29.56%
文字R	81.70%	72.68%
文字P	90.54%	88.36%
文字F	85.89%	79.75%

実験 3 のテキスト抽出における性能低下の主な原因として，以下の 2 点が挙げられる．第一に，読み順が検出順であり，自然な読み順ではなく，バラバラになることで置換エラーが約 55 倍に増加した．第二に，写真や図の中の文字も抽出してしまうため, 挿入エラーが約 100 倍に増加した．これらの結果から, 画像全体に OCR 処理することは，自然な読み順を保ったままのテキスト抽出の精度を著しく低下させることが明らかになった．

また，実験 3 における 76 個のセル未検出の主な原因として，罫線のない表や横線のみの表などが表として認識されないこと，および画像中に占める面積が小さい表も認識されにくいことが挙げられる．これらの結果から，表抽出においても, 画像全体をそのまま OCR 処理を行うよりも，実験 2 のように表領域を事前に検出してから処理する方が，より高い精度が得られることが示された．

6. 結論

本研究では，複雑なレイアウトを持つ画像 PDF から，文書構造を維持したまま構成要素を抽出するシステムの構築を目的とした．具体的には，地方自治体の議会だよりなどの広報文書のうち，文書全体が画像化されているものを対象として, 文章や表の内容を読み取るシステムを構築した．このシステムでは，YOLO を用いた先行研究で検出された文書領域分割の結果を入力とし，PaddleOCR と PP-StructureV3 を組み合わせることで，テキスト，表を高精度に抽出する．特に，日本語文書特有の縦書き・横書きの混在や複雑な配置に対応するため，カテゴリ ID に応じて行の向きを自動判別するソート処理を実装，抽出テキストの順序性の改善により読み順を保持したテキスト抽出を実現した．表領域においては, PP-StructureV3 による解析を優先しつつ，既存のライブラリでは認識が困難な表構造に対しては，OCR の出力座標情報を使用し，CSV/Excel 形式へと変換する独自のフォールバック機構を確立した．一方で，句読点や記号を含む箇所では認識誤りや欠落が発生し，細部の文字認識や表セル内文字の精度向上が今後の課題として残る．

謝辞 本研究は JSPS 科研費 JP22K12740 の助成を受けたものです．

参考文献

[1] Yasutomo Kimura, Keiichi Takamaru, Takuma Tanaka, Akio Kobayashi, Hiroki Sakaji, Yuzu, Uchida, Hokuto Ototake and Shigeru Masuyama: Creating Japanese Political Corpus from Local Assembly Minutes of 47 Prefectures, ALR12, The COLING 2016 Organizing Committee, pp.78-85, 2016.
[2] 乙武北斗，内田ゆず，高丸圭一，木村泰知: 構造化データ作成を 目的とした PDF 地方議会資料のテキスト抽出に関する分析， 第 37 回ファジィシステムシンポジウム講演論文集，pp.431- 436，2021.
[3] 片田江啄門，PDF 文書の階層構造と YOLO を用いた領域分割の性能向上に関する研究，福岡大学電子情報工学科卒業論文，2026.1.
[4] PaddleOCR（オンライン）入手先 https://arxiv.org/pdf/2507.05595 （参照 2026-01-28）.
[5] PP-StructureV3（オンライン）入手先 https://paddlepaddle.github.io/PaddleX/3.0/en/pipeline_usage/tutorials/ocr_pipelines/PP-StructureV3.html （参照 2026-01-28）.
[6] python-Levenshtein（オンライン）入手先 https://pypi.org/project/python-Levenshtein/ （参照 2026-01-28）.