和漢古典籍オープンデータセットを用いたOCRチューニングの試み

久保 旭 *

1 はじめに

近年，古典籍画像のオープンデータ化に伴って，古典籍向けの光学文字認識（Optical Character Recognition，OCR）技術の開発は進展している．古典籍向けのOCRとしてはNDL古典籍OCR[*1]，NDL古典籍OCR-Lite[1]，みんなで翻刻OCR[*2]が公開されている．NDL古典籍OCRはGPUが必須であり，依存するライブラリのインストールなどの難易度がやや高いが，NDL古典籍OCR-Lite，みんなで翻刻OCRはONNX Runtimeを採用しており，より簡単に導入可能である．これらは異なるモデルアーキテクチャをもつが，文書レイアウト解析（Document Layout Analysis，DLA）を行った後に検出された1行領域に対してOCRを行う，というパイプラインモデルを採用している点は共通している．DLA段階において最終的に推定されるラベルは，NDL古典籍OCR，NDL古典籍OCR-Lite，みんなで翻刻OCRのRTMDetモードにおいては1ラベルであり，みんなで翻刻OCRのYOLOモードにおいてはNDL-DocLデータセット[2]に由来する5ラベル[*3]である．

DLAに関してNDL古典籍OCR-LiteとNDL OCR-Lite[*4]を比較した場合，NDL古典籍OCR-LiteではRTMDet[3]が用いられているが，NDL OCR-LiteではDEIMv2[4]に変更されている．DEIMv2はより新しいモデルであり，COCO（Common Objects in Context）val2017データセットを用いたS, M, Lサイズの実験結果を比較すると，平均適合率（Average Precision，AP）がRTMDetよりも改善していることが分かる．

また，近年大規模言語モデル（Large Language Model，LLM）において開発が進められ

* 京都大学人文科学研究所

*1 `https://github.com/ndl-lab/ndlkotenocr_cli`（アクセス日：2026-07-16）

*2 `https://github.com/yuta1984/honkoku-ocr-web`（アクセス日：2026-07-16）

*3 NDL-DocLでは次のラベルが定義されている：資料範囲全体，くずし字の文字ライン，楷書体・行書体の文字ライン，イラスト，印影（蔵書印等）

*4 `https://github.com/ndl-lab/ndlocr-lite`（アクセス日：2026-07-16）

ている視覚言語モデル（Vision Language Model，VLM）を利用したエンドツーエンドのOCR を漢文に適用した研究も行われており，Qwen3-VL-8B を用いることで NDL 古典籍OCR よりも認識精度が改善することが報告されている [5].

こうした進展とともに古典籍データセットの整備も行われているが，物理的構造や論理的構造がアノテーションされ，かつオープンデータとして公開されているものは限られている．NDL-DocL データセットは Public Domain Mark の下で公開されており，1,219 点の古典籍に五つの物理構造がアノテーションされている．画像とアノテーションデータが含まれており，利用の自由度やデータセットの完全性は高いが，テキストは含まれておらず，論理的構造はアノテーションされていない．OCR 学習用データセット（みんなで翻刻）[6] は CC BY-SA 4.0 の下で公開されており，より大規模であるが，データセットは 1 行テキストと画像の組であり，当該領域がテキストであるかどうかの物理構造だけをもつ．また，画像はデータセットに含まれておらず各組織が個別に定めるライセンスに従う必要があったり，運用終了や移転に伴って画像が取得困難になっていたりするなど，ライセンス処理やデータセットの完全性の観点で課題がある．

そこで，本研究ではデータソースとして和漢古典籍オープンデータセット[*5]を用い，OCR データセットの構築を行う．また，レイアウト認識器と文字認識器の学習を行うことでデータセットに最適化し，従来手法に基づくレイアウト認識器，文字認識器の精度向上がどの程度可能かを検証する．このデータセットのほぼ全ての画像は国書データベースから入手可能かつリポジトリ内にも画像をもつことから完全性は高い．また，物理構造の情報は OCR テキストと IIIF 画像との対応関係として取得でき，さらに，一部の書誌は人手修正が行われ，論理構造が付与されていることから，これを用いることで DLA と OCR の学習用データセットとして利用可能と考えられる．

2　和漢古典籍オープンデータセットを用いた学習用データセットの構築

和漢古典籍オープンデータセットの翻刻テキストは GitLab Flavored Markdown 形式で符号化され，バウンディングボックス単位でメタデータ，クロップ画像 URL，テキストをもつ [7]．これを加工することで学習用データセットを構築する．

[*5] `https://gitlab.nijl.ac.jp/Kotenseki/item` (アクセス日: 2026-06-03)

2.1 対象となる Markdown の抽出

和漢古典籍オープンデータセットにおいては複数の手法で Markdown テキストが生成されている場合がある．そのような場合，手法に対応したディレクトリが作成され，その中に Markdown ファイルが格納される．

手法によって，Markdown として符号化される本文やメタデータは異なる．本研究では，学習の用途に応じて抽出対象のディレクトリを変更することとした．詳細は 3 節で述べる．

2.2 バウンディングボックス

画像 URL は IIIF Image API を用いたクロップが適用されていることから，この URL をパースすることでバウンディングボックスを得る．例えば，画像 URL が

```
https://kokusho.nijl.ac.jp/api/iiif/100348333/v4/KYUS/KYUS-01349/
KYUS-01349-00002.tif/1865,815,133,637/pct:40/0/default.jpg
```

である場合，`1865,815,133,637` はバウンディングボックスの x, y, w, h に対応する．

2.3 ラベル

各バウンディングボックスは紙面の物理構造や論理構造を示すラベルをもつ．和漢古典籍オープンデータセットにおいては画像の alt 属性に物理的な位置やラベルの情報が含まれていることから，これを用いる．alt 属性は ABNF 表現で

```
alt = ID [type *(pname"="val)]
```

と定義されており，`type` が物理構造又は論理構造に対応するラベルである．

原稿執筆時点の和漢古典籍オープンデータセットでは，`type` 情報を含むのはごく一部の書誌であり，多くの書誌は `type` をもたない．`type` をもたないテキストに対しては `ID` の体系を利用して暗黙的な `type` を仮定する．具体的には，`ID` の文字列の先頭は `0-9`[*6]，`c` の 2 種類があり，後者は前者とは独立した付番がなされている．`c` をもつバウンディングボックスは版心などに用いられ，本文と独立していることから，便宜的に `annotation` ラベル[*7]を割り当てる．

`type` 情報をもつテキストに対しては `type` をラベルに割り当てる．ただし，`ID` が `pillar`

[*6] “.” を含む．“.” は出現位置の順に数字を割り当てる場合に用いられる省略表現である．

[*7] 実際の `type` にも存在する．

である場合は pillar ラベルを割り当てる．

2.4　1 行テキストの取得

1 行テキストは画像の直後に
text
と符号化されており，この text を取得して用いる．

2.5　モデルの学習

学習は 2 段階で行う．まず，type をもたないテキストを用いて構築されたデータセットから DLA，OCR のモデルを学習する．次に，これらのモデルを人手修正済みデータを用いてファインチューニングし，最終的なモデルを得る．

3　実験

3.1　実験設定

3.1.1　データセット

和漢古典籍オープンデータセットから康熙字典と日本書紀[*8]を取り除いた上で，次のデータセットを生成した．いずれのデータセットについても，訓練：開発：テスト=8:1:1 に分割した．データセットに含まれるラベル数を**表 1** に示す．

stage1　nk3-segment_Qwen3.5-27B-heretic-8bit ディレクトリ，kotenocr3bid ディレクトリ[*9]の少なくとも一方をもつ書誌を母集団として，同一書誌が両方のディレクトリをもつ場合は前者を優先して取得し，元画像 7,921 枚からレイアウトデータセットを構築した．OCR 用データは，バウンディングボックスでクロップされた画像とテキストとの組とした．

stage2　人手修正済み Markdown がある場合は各書誌の proof ディレクトリに格納されるので，この Markdown からデータセットを生成した．該当する書誌は康熙字典，日本書紀を除き五つ存在した．type と ID による暗黙的な type が並立する場合は前者を優先した．OCR 用データは，バウンディングボックスとテキストの組から生成した．

また，康熙字典，日本書紀の人手修正済みデータから評価用データセットを生成した．ページ範囲は [5] と同一とした．

[*8] 両者は [5] における精度評価の対象であることから，比較のために学習データから除外する．

[*9] これらのディレクトリ内の Markdown に type をもつものはなかった．

表 1　データセットのラベル数

	stage1			stage2			康熙字典	日本書紀
ラベル	訓練	開発	テスト	訓練	開発	テスト	テスト	テスト
bodyseg	131,626	16,631	16,493	3,353	348	411	239	33
annotation	12,642	1,770	1,675	185	30	0	0	0
title	0	0	0	7	0	1	8	1
seal	0	0	0	2	0	0	0	0
indent	0	0	0	72	4	2	0	25
pillar	0	0	0	33	4	3	0	0
number	0	0	0	2	0	0	7	0
item	0	0	0	14	0	0	0	0
continuation	0	0	0	18	0	0	0	0
idno	0	0	0	0	0	2	0	0
institution	0	0	0	0	0	1	0	0
chapter	0	0	0	1	1	0	8	1
trailer	0	0	0	83	10	0	0	0
stamp	0	0	0	1	0	0	0	0
name	0	0	0	11	1	1	0	0
volume	0	0	0	18	0	0	8	0
head	0	0	0	1,214	134	141	39	0
front	0	0	0	0	0	1	0	0
predicate	0	0	0	0	0	0	11	0
section	0	0	0	0	0	0	11	0
合計	144,268	18,401	18,168	5,014	532	563	331	60

なお，stage1，stage2，康熙字典，日本書紀の各データセットは COCO 形式に準拠するように生成した．COCO 形式は CVAT（Computer Vision Annotation Tool）[*10]などの主要なアノテーションツールがサポートしている．

[*10] https://github.com/cvat-ai/cvat（アクセス日：2026-07-16）

3.1.2 モデル

パイプラインモデルを採用し，DLA，OCR それぞれに対して stage1，stage2 のデータセットを用いたモデル学習を行った．学習設定は NDL OCR-Lite で用いられたものを参考に一部を変更して用いた[*11]．

DLA　DEIMv2 を用い，公式から提供されているバックボーンのチェックポイントを起点としてファインチューニングを行った．開発データセットの AP@[0.50:0.95][*12]を用い，最高値のチェックポイントを採用した．主な設定を次に示す．

- モデルサイズ：S
- バッチサイズ：32（stage1），8（stage2）
- 最大エポック数：132（stage1），372（stage2）
- 画像サイズ：800*800

OCR　PARSeq[8] を用い，スクラッチから学習を行った．NDL OCR-Lite の変換スクリプトと同様に，データセットのうち文字長が 100 字を超えるサンプルは除外した．開発データセットの正解率を用い，最高値のチェックポイントを採用した．主な設定を次に示す．

- 実験設定：parseq-tiny
- 文字セット：NDL OCR-Lite の全文字及び stage1，stage2 に出現する全文字
- バッチサイズ：256（stage1），64（stage2）
- 最大エポック数：100（stage1），200（stage2）
- 画像サイズ：16*768
- 開発データセットの評価インターバル：1 エポック

OCR の評価に当たっては，正解率に加え，F 値も用いる．F 値の定義は [1] に従い，予測文字，正解文字の多重集合を基にスコアの計算を行う．

3.2 実験結果

DLA の実験結果を**表 2** に，OCR の実験結果を**表 4** に示す．全体的に康熙字典，日本書紀でのスコアは低い．

DLA においては，stage1 は開発データセット，テストデータセットともに高精度でレイアウト検出ができている一方，stage2 ではテストデータセットのスコアが低下した．また，

[*11] 学習設定の詳細は次の URL を参照：https://codeberg.org/kubo/oricom41-ocr/
[*12] IoU（Intersection of Union）を 0.50-0.95 で段階的に変化させた場合の AP の平均

表 2　各段階における DEIMv2 の AP@[0.50:0.95]

段階	開発	テスト	康熙字典	日本書紀
stage1	0.74	0.75	0.08	0.06
stage2	0.64	0.29	0.03	0.17

ラベル単位の実験結果を**表 3** に示す．ラベル単位では事例数が多いものが高精度な傾向にあり，bodyseg はいずれのデータセットにおいても最高精度となったが，康熙字典，日本書紀においては bodyseg 単独で評価した場合も低精度であった．また，全く認識できていないラベルが複数存在した．ただし，康熙字典，日本書紀の textseg，annotation 以外のラベルを全て textseg に置き換えて評価した場合，stage1 ではそれぞれ AP=0.49，AP=0.58 となった．したがって，物理構造と論理構造が混在しているラベル設計が stage2 における精度低下に影響した可能性がある．

OCR の場合は最適化対象である学習用データセットのスコアが低く，出現順序を考慮しない指標である F 値においても高いとはいえない．[5] における Qwen-VL-8B を用いた実験では康熙字典が F=0.860，日本書紀が F=0.910 と報告されており，同一ソフトウェアである PARSeq を用いた [1] においては，対象史料が異なるが F=0.894 と報告されている．よって，更なる精度向上の余地があると考えられる．なお，正解率が低いにも関わらず F 値が高いのは，文字は正しく認識したが出現位置が誤っていることによる影響と考えられる．

また，いずれの場合にも stage1 よりも stage2 の方が康熙字典において低スコアとなった．stage2 は小規模であり，含まれている書誌が少ないことから，ファインチューニングに用いられた学習用データセットと康熙字典の差異が大きく，精度に影響を与えた可能性がある．

3.2.1　計算資源の使用状況

本研究の実験は NVIDIA Grace Blackwell GB10 を具備する DGX Spark System[*13] 上で行われ，**表 5** に示す使用状況となった．いずれのソフトウェアも stage1 の方が資源を使う処理であり，これはデータセットの規模に起因すると考えられる．また，GPU メモリは最大 42.0GB 使用しており，これを満たすように GPU を構成する必要がある．DEIMv2，PARSeq はいずれもデータ並列によるマルチ GPU に対応していることから，現行ラインナップの構成例は次のようになる．

[*13] NVIDIA による製品と複数他社による OEM 製品が存在しており，本稿では総称して DGX Spark System と呼ぶ．これらの製品はほぼ同一仕様である．

表 3 stage2 における DEIMv2 のラベル単位 AP．空欄はデータセットに正解データが存在しないことを示す．

ラベル	テスト	康熙字典	日本書紀
bodyseg	0.73	0.12	0.27
annotation			
title	0.20	0.00	0.00
seal			
indent	0.90		0.26
pillar	0.00		
number		0.00	
item			
continuation			
idno	0.00		
institution	0.00		
chapter		0.00	0.15
trailer			
stamp			
name	0.00		
volume		0.03	
head	0.78	0.08	
front	0.00		
predicate		0.00	
section		0.00	

- GeForce RTX 5060Ti（16GB）× 3 個
- GeForce RTX 5070（12GB）× 4 個
- RTX PRO 4000 Blackwell（24GB）× 2 個
- DGX Spark System（128GB ユニファイドメモリ）
- RTX PRO 5000 Blackwell（48GB）

リストは原稿執筆時点においてコストが低い順に並べられており，少メモリの複数 GPU 構成が導入コストにおいて有利な傾向にあるが，複数 GPU の使用は消費電力やオーバヘッドの増加など，運用上の新たな課題を生じる可能性がある．また，本結果はあくまでも最小

表 4　各段階における PARSeq の正解率，F 値

段階	正解率		F 値					
	開発	テスト	康熙字典	日本書紀	開発	テスト	康熙字典	日本書紀
stage1	0.30	0.28	0.24	0.35	0.80	0.81	0.80	0.79
stage2	0.54	0.54	0.15	0.25	0.79	0.78	0.74	0.80

表 5　計算資源の使用状況

ソフトウェア	タスク	実行時間 [hrs]	GPU メモリ [GB]
DEIMv2	stage1	13.2	42.0
	stage2	0.75	9.3
PARSeq	stage1	13.1	44.9
	stage2	0.66	4.6

のモデルサイズを用いた場合であり，より大きなモデルサイズを用いる場合，前述の構成では Out of Memory が発生して学習できない可能性がある．他方で DGX Spark System を用いる場合，より新しいアーキテクチャが採用されていることから，ライブラリの互換性によるエラーの発生などに対処する必要がある．

4　おわりに

本研究では和漢古典籍オープンデータセットを用いて OCR データセットを構築し，レイアウト認識器，文字認識器の学習と精度評価を行った．

今後の課題としては，正解データセットの拡張が挙げられる．精度上の課題はデータセットの規模によって解決する可能性があるが，人手による作業コストは大きいことから，既存のデータセットから効率的に OCR データセットを生成する手法を検討する必要がある．また，みんなで翻刻 OCR の技術情報[*14]によれば，同 OCR の学習用データセットがオープンライセンスで公開される予定であり，これを併用することで精度向上に寄与する可能性がある．DLA については，ラベルを物理構造と論理構造に分け，文書構造の解析により特化した PP-Structure[9] などのモデルを用いてファインチューニングを行うことで，物理構造の推定精度向上を図り，LLM や VLM との組合せによって論理構造の推定精度向上を図る，といった方向性が考えられる．また，本研究では行間の読み順推定を取り扱わなかったが，複雑な

[*14] https://yuta1984.github.io/honkoku-ocr-web/tech.html（アクセス日：2026-07-16）

レイアウトをもつ書誌ではルールベースによる推定が失敗するケースがあり，そのような書誌は人手の整序が負担となることから，読み順推定も重要な課題といえる．これに対しては，和漢古典籍オープンデータセットの ID をパースして読み順の正解データを構築し，機械学習を用いて読み順推定器を学習することが考えられる．また，NDL OCR，NDL OCR-Lite では物理行のレイアウト認識だけでなく，複数の物理行からなるテキストブロックの認識も行われており [10]，ID を加工することでそのようなレイアウト情報も付加できる可能性がある．

参考文献

[1] 青池亨. CPU 環境で高速に動作する軽量 OCR 「NDL 古典籍 OCR-Lite」 の開発. じんもんこん 2024 論文集, Vol. 2024, pp. 181–186, 2024.

[2] 国立国会図書館. NDL-DocL データセット (資料画像レイアウトデータセット). `https://github.com/ndl-lab/layout-dataset`. アクセス日：2026-07-15.

[3] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv preprint arXiv:2212.07784*, 2022.

[4] Shihua Huang, Yongjie Hou, Longfei Liu, Xuanlong Yu, and Xi Shen. Real-Time Object Detection Meets DINOv3. *arXiv preprint arXiv:2509.20787*, 2026.

[5] 守岡知彦. 視覚言語モデルを利用した漢籍用 OCR 実現の試み. 東洋学へのコンピュータ利用第 40 回研究セミナー, 2026.

[6] 国立国会図書館. OCR 学習用データセット（みんなで翻刻）. `https://github.com/ndl-lab/ndl-minhon-ocrdataset`. アクセス日：2026-07-15.

[7] 守岡知彦. Markdown を用いた古典籍 OCR テキストの簡易マークアップの試み. 研究報告人文科学とコンピュータ, Vol. 2025-CH-139, No. 8, pp. 1–5, 2025.

[8] Darwin Bautista and Rowel Atienza. Scene Text Recognition with Permuted Autoregressive Sequence Models. *arXiv preprint arXiv:2207.06966*, 2022.

[9] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025.

[10] 国立国会図書館. 令和 4 年度 NDLOCR 追加開発事業及び同事業成果に対する改善作業. `https://lab.ndl.go.jp/data_set/r4ocr/r4_software/`. アクセス日: 2026-07-15.