

博物館における OCR を使用した展示パネルからの展示物名・説明文の取得 Acquisition of exhibit names and descriptions from display panels using OCR

山里 拓夢[†]
Takumu Yamazato

赤嶺 有平[‡]
Yuhei Akamine

根路銘 もえ子[§]
Moekoo Nerome

1. はじめに

新型コロナウイルス感染症の流行により、博物館等、多くの文化教育施設において利用制限が行われ、デジタル化の必要性や有効性などが明らかになった。その結果、博物館では資料をデジタル化して保存するデジタルアーカイブ化が促進され、デジタルコンテンツの作成および公開が義務付けられている[1]。しかし、調査結果をみると全国の博物館 1530 館のうち 49.2%と約半数が、デジタルアーカイブの実施有無について「実施予定なし」と解答している。更にデジタルアーカイブに関する専門知識を持った職員の有無について約 75%が在籍していないとなっている[2, 3]。デジタルアーカイブ化には、テキストデータの入力といった人依存の作業が多い。また、DB 構築などの専門的な技術が要される場合があり、人や時間、技術コストがかかる。そのため、中小規模の博物館においては、デジタルコンテンツの作成はハードルが高い。

以上のことを踏まえて、筆者らの研究チームでは、深層学習や自律制御ロボットを用いて作業を自動化するシステムを構築し、コンテンツの作成から利用者へのコンテンツ提供までを完全に自動化するシステムの検討構築を行っている。このシステムの流れを以下に示す。

1. 自律走行ロボットが館内を走行することで、館内のマップを作成し、館内の撮影を行う[4]。
2. 撮影された画像から、物体検出モデルを用いて、画像内の展示物と展示パネルを検出する。
3. 展示パネルが検出されたら、その展示パネルに対して、OCR を用いて文字認識を行うことで、展示物名と説明文を取得する。
4. 展示パネルがどの展示物を説明しているのかの結び付けを、展示物と展示パネルの距離や、展示パネルなどから取得した情報などを用いて行う。
5. 今まで取得した情報をデータベースに保存し、デジタルアーカイブを作成する。

本研究では、展示物の名前や説明文を取得するタスクに焦点を当てている。博物館のデジタルアーカイブ化において、展示物の名前を正確に取得することは重要である。デジタルアーカイブに名前での検索機能をつけるためには、展示物の名前が正確に取得されていなければならない。誤った名前で登録していると、検索できなくなってしまう。

また、誤った名前で登録すると、同じ展示物が異なる名前で登録される可能性があり、データの重複が生じる可能性がある。そのため、名前を正確に取得するタスクを解決

するのは、重要なことである。また、間違った説明文を取得してしまうと、その展示物の情報が正確に伝わらないため、説明文を正確に取得することも重要である。

展示物全ての名前や説明文を手入力で取得するには莫大な時間が掛かってしまう。博物館では、展示物の隣に説明パネルが設置されており、そこに展示物の名前や説明文が書かれていることが多い。そのため、説明パネルからテキストデータを取得することができると考えられる。そこで、本研究では、光学式文字認識(OCR)を説明パネルに実行することで、展示物の名前や説明文を取得する研究を行う。

2. 関連研究

画像から文字列を取得する手法として OCR がよく知られている。

Saravanan らは、手書きの処方箋の自動認識に OCR を活用し、医療現場における処方情報のデジタル化とヒューマンエラーの低減を図っている。この研究では、処方箋から Tesseract OCR や Google Vision API といった深層学習ベースの OCR モデルを利用することで、テキストデータを取得し、それに対して NLP(自然言語処理)や NER(固有表現抽出)を用いて医薬品名などの重要な情報を抽出している。また OCR の生データにはスペルミスや誤認識が含まれているため、医薬品データベースと照合させることで、誤認識の訂正を行っている[6]。

このほかにも、Samarth R hegde らは、OCR を用いて、レシートなどの書類から、経費情報を取得する Web アプリケーションを設計している。手書きの領収書や請求書、銀行明細などから、OCR を用いて、テキストを取得し、この取得したテキストから、NLP を用いて金額や日付といった財務情報を抽出している。この研究でも、取得したテキストに対してミスの検出を行っており、金額は数字であるなどのルールベースの手法に加えて、過去のデータを使って機械学習をすることで、誤認識されやすいパターンを把握することで、OCR の出力を補正している[7]。

本研究では、OCR に YomiToku を利用した。YomiToku は、日本語の OCR に特化したオープンソースの OCR エンジンであり、7000 文字を超える日本語文字の認識、縦書きなど日本語特有のレイアウト構造の文書画像の解析も可能である。日本語に特化しているが、英語にも対応している。Python3.10+, Pytorch が実行に必要で pip でインストールすることができる。指定した画像ファイルを読み込み、文字認識の結果を json, csv, html, md(markdown) 形式で確認ことができ、解析結果を可視化した画像を出力することもできる[8]。

[†] 琉球大学理工学研究科 Graduate School of Engineering and Science, Ryukyu University

[‡] 琉球大学工学部 University of Ryukyus

[§] 沖縄国際大学経済学部 Okinawa International University

3. 予備実験

筆者らが作成しているデジタルアーカイブ自動化システムは、実際の博物館などで使用されることを想定している。そのため、商用利用できるような精度が求められる。

実際に商用利用されている OCR に、Google レンズがある。Google レンズは、Google が提供する画像認識技術で、OCR だけでなく、物体検出や画像検索なども行うことができ、一般ユーザーに利用されている。Google レンズは、日本語に対して高い認識精度を持つ。また、Google レンズは、傾いている文字や、斜めや傾いている文字に対しても高い精度で認識することができる。しかし、Google レンズはクラウドサービスなのでサービスが継続される保証がないことや、認識対象となるテキストが膨大なのでコストがかかるなどの問題がある。

本研究での提案手法の精度目標設定のため、Google レンズを用いて、説明パネルからテキストデータの取得精度を調べる予備実験を行った。

3.1 実験内容

実際に博物館で撮影した画像 64 枚に対して、Google レンズを使用し、テキストデータを取得した。取得したテキストデータの精度を以下の式で求め、平均値で評価する。

$$\text{精度} = \frac{\text{正しく読み取れた文字数}}{\text{画像に写っている文字数}} \cdot 100$$

3.2 実験結果

実験の結果、Google レンズの平均精度は 99.5%であった。図 1 は、説明パネルが傾いている画像であり、これに対して読み取れた結果が図 2 である。比較結果から、説明パネルが傾いている場合でも、高い精度で認識できることがわかる。

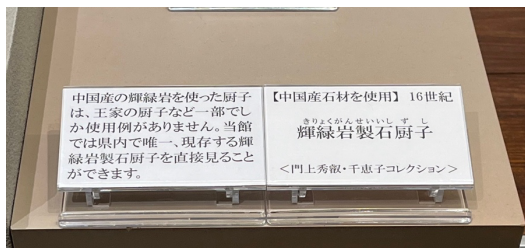


図 1 説明パネルが傾いている画像

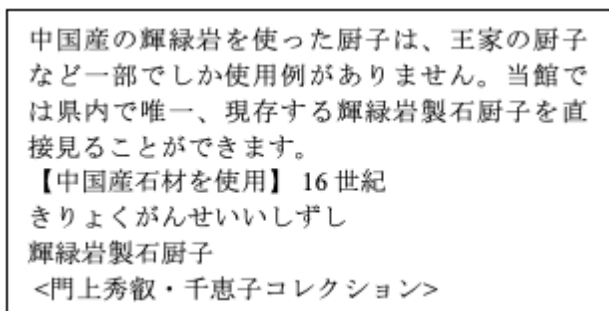


図 2 Google レンズの実行結果

4. 提案手法

本研究では、展示物の名前や説明文を説明パネルから取得することを目的としている。そのため的手法とし、以下の手順を提案する。

1. 画像から説明パネルの領域を推定
2. 推定した領域から画像を補正
3. 補正画像からテキストデータを取得
4. 取得したテキストデータに対するミスの指摘

4.1 セマンティックセグメンテーションを用いた画像補正

この研究では、自律走行ロボットが説明パネルの画像を撮影することを前提としており、説明パネルが画像に対して正面を向いていない可能性がある。そのため、取得した画像に対しての補正機能を実装する。

画像補正のための説明パネルの領域推定は、U-Net を使用したセマンティックセグメンテーションで行う。

4.1.1 モデルの学習について

U-Net の学習は Pytorch を使用した。学習には多くの説明パネル画像が必要であり、実際の博物館で撮影した画像では十分な学習を行うことができなかった。そのため、画像を自作した。自作した画像は、実際の説明パネルの切り取り画像を、博物館で撮影した背景画像にランダムな位置と角度で貼り付けたものである。この結果、実際に撮影した画像と、自作した画像合わせて 660 枚の画像を用意した。

660 枚の画像のうち、Google レンズの予備実験で使用した画像 66 枚をテストデータとし、残りの 594 枚でクロスバリデーションを行った。594 枚を 9 分割し、8 つを訓練データ、1 つをバリデーションデータとした。学習率は 0.001 で最適化手法は Adam を使用し、Early Stopping を採用した。各 fold ごとのバリデーション損失が最も小さいモデルを選択し、保存した。

4.1.2 画像補正について

作成したモデルは、入力画像に写っている説明パネルの部分をマスクした画像を出力する。この出力された画像に対して以下の手順により画像補正を行った

1. マスクされた部分の輪郭を抽出
2. 抽出した輪郭の凸包を計算
3. 凸包内の最大の四角形を計算
4. 四角形の 4 つの頂点座標を取得
5. 取得した座標により射影変換のための行列を計算
6. 行列を元に元画像に対して射影変換

4.2 OCR ミス指摘

OCR により取得したテキストデータは、誤りが含まれている可能性がある。そのため人による目視の確認が必要になる。これに対応するために、LLM によるテキストデータのミス検出を行う。本研究で使用するのは、OpenAI が 2025 年の 4 月に公開した GPT-4.1 であり、従来のモデルよりもコーディング能力や出力速度が向上したものである。また OpenAI は API 機能の Structured Output を提供している。この機能は、モデルの応答を予め指定した JSON Schema に従って整形するための機能である。OCR のミス指摘では GPT-4.1 と Structured Output を組み合わせて指摘を行う。

5. 実験結果

5.1 セマンティックセグメンテーションの精度

セマンティックセグメンテーションのモデルを Pytorch を用いて実装し、学習データを用いて学習を行った。学習した結果が表 1 である。これを見ると IoU の平均値は 0.8970 で、精度の平均値は 0.9809 であった。平均 IoU が最も高い fold1 の test データに対する結果を見てみると、図 3 のような説明パネルに影がある場合は、パネルの領域が誤検出されることがあった。また、図 4 のようにパネルが白色でない場合は、パネルの領域が誤検出されることがあった。これは学習データの説明パネルのほとんどが白色のパネルであったため、白色以外のパネルに対しては、モデルが適切に学習できていないことが原因であると考えられる。

fold	val		test	
	平均 IoU	平均精度	平均 IoU	平均精度
0	0.8356	0.9597	0.8732	0.9714
1	0.9178	0.9821	0.9253	0.9837
2	0.8784	0.9719	0.8445	0.9759
3	0.9277	0.9840	0.8969	0.9819
4	0.9303	0.9846	0.9047	0.9825
5	0.9131	0.9835	0.9120	0.9836
6	0.8998	0.9770	0.8841	0.9804
7	0.9006	0.9786	0.9138	0.9842
8	0.9379	0.9868	0.9187	0.9846
平均	0.9046	0.9787	0.8970	0.9810
標準偏差	0.0299	0.0080	0.0243	0.0042
最大値	0.9379	0.9868	0.9253	0.9846
最小値	0.8356	0.9597	0.8445	0.9714
最大 - 最小	0.1023	0.0271	0.0808	0.0132

表 1 Cross Validation の実行結果

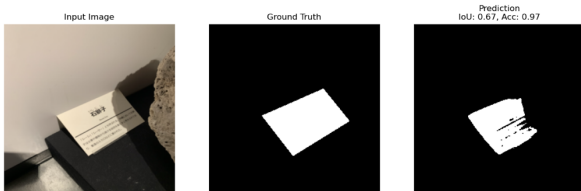


図 3 説明パネルに影がある場合の実行結果

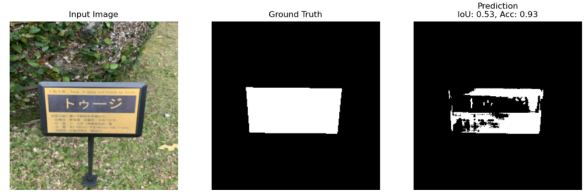


図 4 説明パネルの色が白でない場合の実行結果

5.2 画像補正の精度

Google レンズの精度を確かめるために使った画像 56 枚に対して、表 1 の中で最も IoU の精度が高かった fold1 のモデルを使用して、画像補正を行った。結果 56 枚の画像のうち 44 枚の補正に成功した。このことから、画像補正の精度は約 78.6%であった。画像補正が失敗する場合は、セマンティックセグメンテーションで正確な説明パネルの輪郭を取得できていないことで失敗していた。そのため、セマンティックセグメンテーションの精度がまだ足りていないことが示された。

5.3 YomiToku の精度

補正に成功した画像 44 枚の補正前と後に対して YomiToku を実行し、YomiToku の精度を確認すると共に、画像補正が認識精度の向上に関与するのかを調べた。OCR を実行して認識できた文字数を確認すると、7645 文字の内、補正なしの場合は 7345 文字、補正ありの場合は 7352 文字であった。このことから、補正の有無では YomiToku の精度はほとんど変化しないことが明らかになった。

しかし、補正により認識できた文字が 10 文字以上増えた画像があり、それが図 5 である。画像に対する補正前と補正後の結果を比べると、補正前は 87 文字、補正後は 99 文字であった。この画像の説明パネルは他のテスト画像に比べて説明パネルが大きく傾いている。このことから、傾きが大きい場合に画像補正が有効である可能性があることが示唆された。

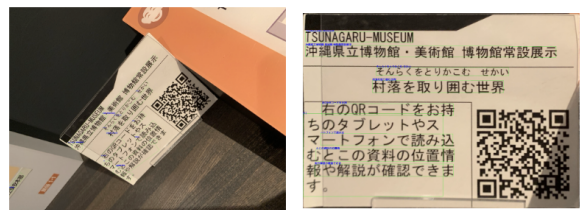


図 5 左:説明パネルが傾いている画像(補正前) 右:左の画像に対して補正を行った画像(補正後)

5.4 OCR のミス指摘

ミス検出の精度検証をするために、YomiToku が実際に起こすミスについて調べる必要がある。そのため、以下の手順を用いて、YomiToku が実際に起こすエラーを含んだ文章を取得した。

1. Wikipedia からランダムで文章を取得
2. 取得した文章を用いて低解像度の説明パネル画像を作成
3. この画像に対して YomiToku を実行

上記の手順で取得した文章に対して GPT-4.1 と Structured Output を用いてミス検出を行った。今回の実験で使用したプロンプトと JSON Schema は図 6 に示す。

```
class Error(BaseModel):
    error_position: str
    type: str
    confidence: float
    suggestions: str

class WordCheck(BaseModel):
    errors: list[Error]

prompt = """
次の文章には本来、誤字や脱字、不自然な表現などの間違いは含まれていません。
しかし、これは OCR で取得されたテキストであるため、誤字や脱字が含まれている可能性があります。
OCR は「計」が「言」のように文字の見た目が似ている誤字を起こすことはありますが、
漢字への変換ミスなどはありません。

あなたのタスクは、以下の文章の中に「意味をなさない部分」があるかを検出し、それが「誤字」か「脱字」かを分類し、エラーを挙挙してください。
修正のための提案は単語単位で行ってください。間違っていることはわかるが、修正方法がわからない場合は、空で返してください。

文章全体を対象とし、意味をなさない語句・表現をすべて対象としてください。"""
```

図 6 実験で使用したプロンプト

これを用いて、ミス 63 箇所に対してどの程度ミスを発見できるかを確認したところ、46 箇所のミスを発見することができた。このことから、ミスの検出精度は約 73%であることが示された。

6. おわりに

本稿では、博物館での自動デジタルアーカイブシステムのための、OCR を利用した展示物名と説明文の取得方法を検討した。結果として、YomiToku の精度は高いことが示されたが、説明パネルが大きく傾いている場合には精度が落ちることが明らかになった。それに対応するためにセマン

ティックセグメンテーションを利用した画像補正は有用であることも示唆された。現在の補正手法の精度は約 79%であり、全ての説明パネル画像に対して行うことができなかった。GPT を利用した OCR のミス検出に関しては約 73%の精度で発見できた。

今後の展望として、補正精度の向上を目指す。またミス検出の精度に関しても、現在は 73%ほどであり、実運用にはまだ精度が足りていないため、精度の向上を目指す。更に、LLM を利用すると、ミスの修正候補などを提案できる。そのため、現在はミスの場所のみを指摘しているが、修正候補などを提供できるようにする。その後、取得した情報をもとに効率的なデータベースの作成手法を検討する。

謝辞

本研究は JSPS 科研費 JP22K01014 の助成を受けたものです。

参考文献

- [1] 文化庁, 博物館 DX の推進に関する基本的な考え方(案), https://www.bunka.go.jp/seisaku/bijutsukan_hakubutsukan/pdf/94126601_01.pdf, 2020/2/13.
- [2] みずほ総合研究所株式会社, -文化庁委託事業-令和 2 年度「博物館ネットワークによる未来へのレガシー継承・発信事業」における「博物館の機能強化に関する調査」, https://www.mizuho-rt.co.jp/archive/case/pdf/museum2020_01.pdf, 2021/3.
- [3] 文化庁, 博物館資料のデジタル・アーカイブ化の目的・状況について, https://www.bunka.go.jp/seisaku/bunkashingikai/hakubutsukan/hakubutsukan04/02/pdf/93734001_01.pdf
- [4] 赤嶺 有平, 上原 和樹, 根路銘 もえ子, 當間 愛晃, “RODM: 普及型モバイルデバイスの LiDAR センサを活用した自律走行ロボット開発フレームワーク”, 情報処理学会論文誌, Vol. 65, No. 12, (2024).
- [5] Saravanan G., Gowri R. P., Naveen P. N., Sankar R., Sudharsan R., “Smart Prescription Recognition And Symptom Analysis For Healthcare Solutions”, *International Research Journal on Advanced Engineering Hub (IRJAEH)*, Vol.03, No.03 (2025).
- [6] Hegde S. R., Gangatkar H., Pradyumna V. G., Hayavadana M. B., Sudha M., “Automated Expense Tracking with OCR: Enhancing Financial Management through Technology”, *International Advanced Research Journal in Science, Engineering and Technology*, Vol.12, No.1 (2025).
- [7] Kotaro-kinoshita/yomitoku, YomiToku, <https://github.com/kotaro-kinoshita/yomitoku>