WASEDA University

Smart SE

IoT/AIコース　修了制作

AI-OCR機能の実装

酒井辰則

開発における問題点

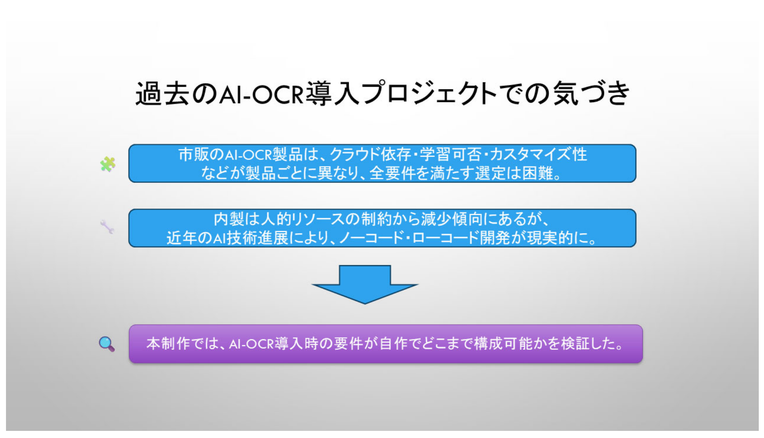
AI-OCR製品は数多く販売されているが、クラウド利用が前提であったり、継続的な学習の可否、カスタマイズによる機能拡張性、AIを適用する範囲など、製品によって特徴が様々なため、利用者側で想定する全要件を満たす製品を選定するのは難しい。

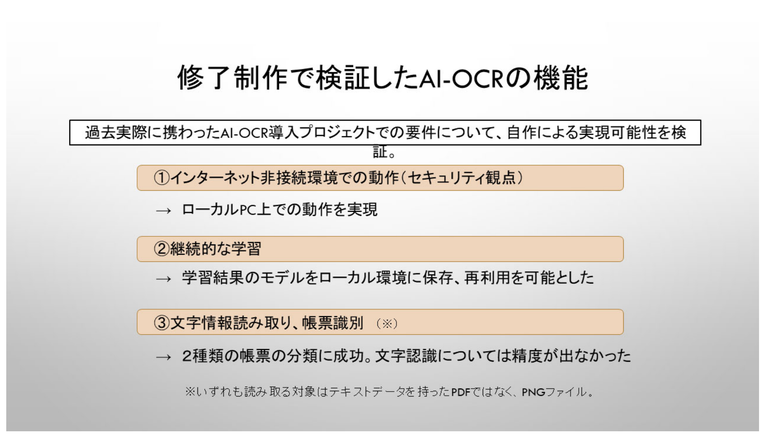
製品の選定にあたっては、必ず必要な機能を持つ製品の中から選ぶ必要があり、選択肢が限られてしまう。

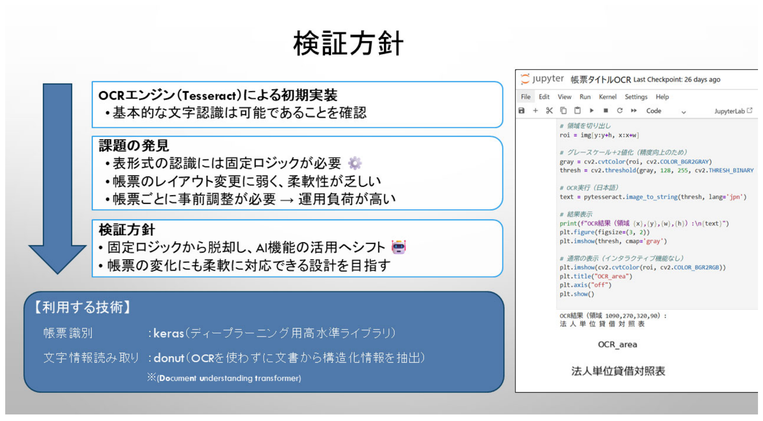
手法・ツールの適用による解決

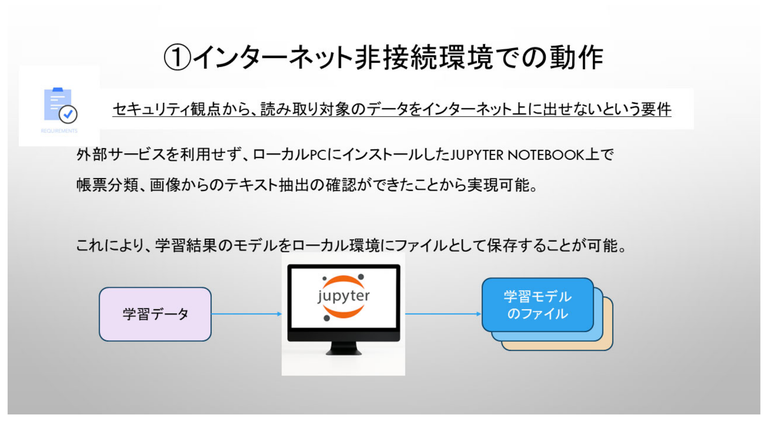
昨今AI技術の発展により、ChatGPT・Copilotなどに指示を出すことで、ノーコード・ローコード開発が可能となってきている現状がある。

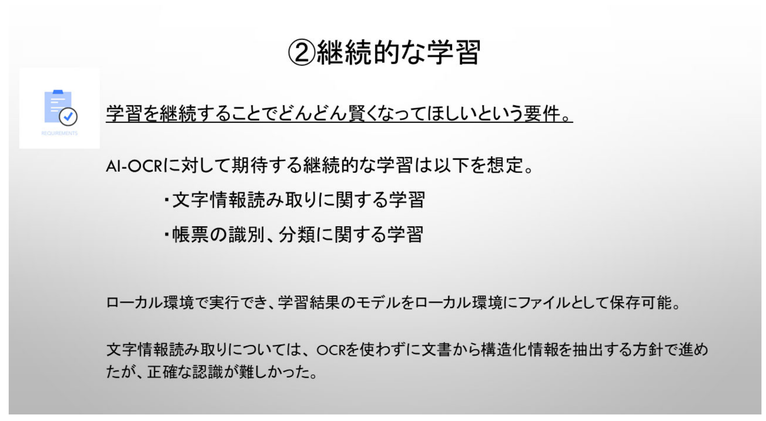
そこで本修了制作では、過去に経験したAI-OCR導入プロジェクトで、製品の選択肢を狭める要因となったいくつかの要件（特に帳票識別機能）について、自作でどこまで実装が可能か検証を行った。

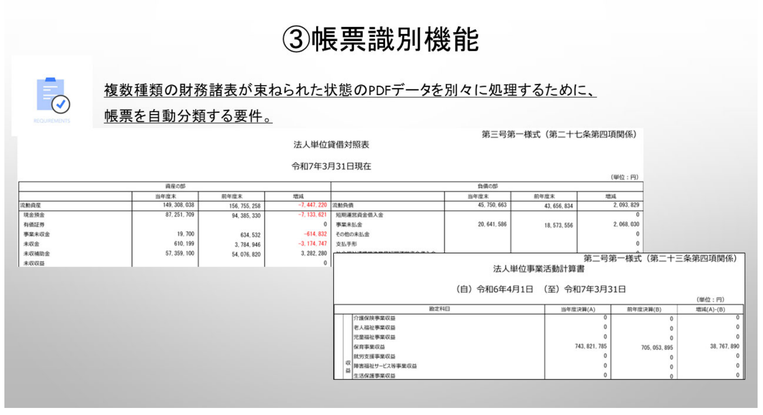


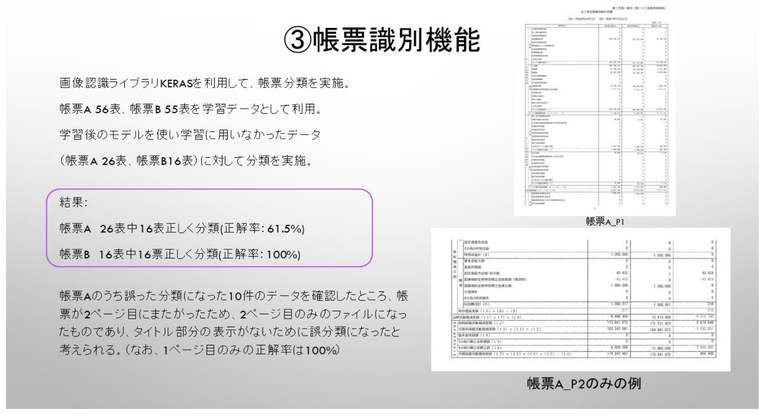


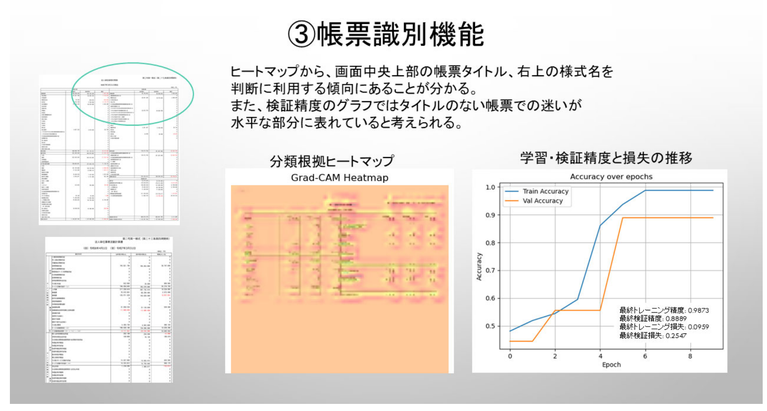












まとめと課題

ローカルPCでの動作を実現したことにより、インターネット非接続環境で運用可能。
・セキュリティを確保しつつAIの機能が利用可能

学習結果をローカル環境に保存可能。
・帳票の変更や、新帳票の追加時も既存の学習結果に影響を与えず新たな学習が可能。
・年度ごとに学習しなおして履歴管理するなど、柔軟な運用が可能。

帳票の識別機能単体の実装が可能。
・事前に帳票分類ができるため、識別機能のない製品も導入候補になり、製品選択の幅が広がる。

これからの課題
・OCRを使わずに文書から構造化情報を抽出するための実装、学習方法の改善。
donutの初期学習モデルで精度が出なかったためファインチューニングを行うも改善されず。
・LLM基盤自体をローカル環境で動作させる検証。
インターネット非接続かつコーディングの負荷も抑えて画像からの文字抽出の仕組みを構築できる可能性。

制作を通しての気づき

帳票分類や画像からのテキスト読み取りといったAIに関するプログラムの製作自体よりも、学習に利用するデータの整備、またその作業に必要となるプログラム作成も同等あるいはそれ以上の作業負荷がかかると感じた。(AI：70時間、周辺PG：80時間程度)

【作業に必要となったプログラムの例】

学習用データが大量に必要となるためWEBからデータをダウンロードするためのスクレイピングとZIPからのファイル抽出、PDFファイルの画像化、ファインチューニングのための構造を表すJSONファイルの作成(それぞれ約230法人分の処理に使用)、OCRによる処理検証用に、画像と座標を渡してその部分のOCR結果（TESSERACTによる）を返すプログラム。

👉対応として、COPILOTやCHATGPTを積極的に活用した。

copilotやchatGPTによって、プログラムを書くことの効率化が図れる一方、プロンプトに対して再現性のある形で正確な指示を出すことの難しさを体験できた。

【混乱した例】

先行PGのファイル出力先フォルダが変わったのに、作成された後続PGは元のフォルダから入力していた。
エラー内容の対応が堂々巡り(データ型エラーの都度、PG間受渡しのデータ型を変えて戻しを繰り返すような回答)。

👉対応として、プロンプトに対してはテンプレート化した書式で指示することで抜け漏れがないようにした。

スマートエスイー：スマートシステム＆サービスおよびDX推進を担う人材の産学連携育成