# Literature Review

## Receipt Extractor

### Paper 1 - Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms.

This study made by Joel Odd and Emil Theologou research and develop on “investigated if it was feasible to use machine learning tools on OCR extracted text data to classify receipts and extract specific data points” (Odd and Theologou, 2018).

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Technologies** | **Pros** | **Cons** |
| **Optical Character Recognition (OCR)** | Azure Computer Vision API | Good accuracy in text recognition. | The security is uncontrollable due to third party dependance.  May require significant data preprocessing. |
| Google Drive REST API | Good performance and time response. | The security is uncontrollable due to third party dependance.  Managing variations in receipt formats can be complex |
| Tesseract OCR | Deployable locally and not dependent of third party. | Varying receipt formats may affect accuracy. |
| **Machine Learning Models** | Custom N-gram Model | The model is efficient with classifying receipt. | Subjective classification dependent on training data labels. |
| Scikit-learn | The model is good in automated classification and data extraction. | Small dataset size may affect model generalization to receipt generalization. |

Has showed in the table 1, it is showed that two elements compose the receipt extractor, the Machine learning model and the Optical Character Recognition (OCR) and that for these elements multiple technologies were tested. For the OCR, we have the Azure Computer Vision APIU and Google Drive Rest API which are efficient but are third party dependent and may require data processing, on the other hand tesseract OCR can be deployed locally and result more quickly but not has efficient. For the machine learning model two technologies were tested, the Custom N-Gram Model which is dedicated to the purpose with efficient classifying and share the same weakness has the Scikit-learn which is the generalization problem due to a small dataset.

### Paper 2 - Information Extraction from Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques

For the research paper made by W.B. Darsha which solve the fooling problem “Extracting information from scanned invoices (images) is a challenging task”(Darsha, 2023) using an model and a OCR.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Technologies** | **Pros** | **Cons** |
| **Object detection model** | YOLO version 5 | Efficient image detection model.  Fast inference time. | Require extensive computation power for many classes. |
| **Optical Character Recognition (OCR)** | Tesseract OCR | Include many languages supported | May struggle with text in complex layout or low-quality images. |

Has showed in Table 2, this paper used the YOLO object detection (You Only look Once) to identify the different aspect of the receipt follow by the usage of the Tesseract OCR mentioned in the earlier section of the Literature Review. The YOLO model can be easily call and train through the import of Ultralitics python package.

### Paper 3 - Computer Vision for Document Image Analysis and Text Extraction

This study has for purpose to “investigates in depth a major component used in Document Image Processing known as Optical Character Recognition (OCR).”(Benchekroun, 2022), it hasn’t the same goal but share the same step of analysis and text extraction with a study using synthesis data..

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Technologies** | **Pros** | **Cons** |
| **Optical Character Recognition (OCR)** | Convolutional Neural Networks (CNN) + Long Short-Term Memory (LSTM) | Effective for non-handwritten text and shows significant accuracy improvement with synthetic data. | Are computationally intensive and may require large dataset for high accuracy. |
| Convolutional Neural Networks + Transformer architecture | Efficient for handwriting text and handling diverse text styles |

The study experiments on the usage have showed in table 3 of two OCR within the use synthetic data which improve of 24% bringing the accuracy on non-handwritten to 97%, giving also a solution for handwriting with CNN+Transformer architecture which in our project is not relevant since the main data will be printed. Furthermore, the usage of synthetic data since promising to increase the training dataset and allow to the model a better globalization of the extraction.

## Web App

### Paper - React Native vs Flutter, cross-platform mobile application frameworks.

This research paper done by Wenhao Wu confront two mobile programing language, react native and flutter to “execute a comprehensive study on React Native and Flutter”(Wu, 2018)

|  |  |  |
| --- | --- | --- |
| **Programming Language** | **pros** | **Cons** |
| React Native | Strong Community Support with big tech contribution.  Bring modern web techniques to mobile support.  Use JavaScript syntax extension for designing UI.  Access native hardware feature like camera and storage.  Encourage modularity and reusable of component. | Have performance limitation compared to native app in complexes scenario using JavaScript thread or memory optimization.  Highly dependable on third-party libraries for navigation and file system operations which may affect consistency and the reliability. |
| Flutter | Hight-Performance using his own rendering engine for view component offering a close performance to native application.  Using Dart Programming language which is efficient for memory management and garbage collection offering fast performance.  Providing customizable widget for development.  Hot-Reload feature for development.  Assuring a consistency across the different platform. | Produce larger application size due to the different widget renderer that may affect the app size.  Relatively new community compared to react native which may affect the resources available. |

From the table 4 we can see the advantage and limitation of React Native and Flutter programing language, both cross-platform and with their own strength in different domain. React Native having a big community and using modern web technique while Flutter having some high rendering performance with consistent UI across platform offer a good alternative for mobile application development.

## Summary

Has we saw in the previous sections of the Literature Review, the project can be separate in two parts, the receipt extraction model and the mobile application having both multiple technologies usable. For the receipt extraction model which can also be separate into two technology such has the Object detection Model followed with Optical Character Recognition technologies. From the different research, the YOLO object detection model version 8 seems the most suitable option with his fast efference time and accessibilities with python language combine with Tesseract OCR also fully compatible to python and his efficiency to extract printed text suite perfectly my requirement for receipt extraction being both state-of-the-art technologies in their domain. As for the mobile application development, Flutter programming language is more appropriate compared to React Native with is high performance in rendering and his consistent UI across platform.