Table of Contents

[Background of study 2](#_Toc148706253)

[Problem Statement 2](#_Toc148706254)

[Information Extraction From Scanned Invoices using Machine Learning, OCR and Spatial Feature Mapping Techniques 2](#_Toc148706255)

[Computer Vision for Document Image Analysis and Text Extraction 3](#_Toc148706256)

[Utilize OCR text to extract receipt data and classify receipts with common Machine Learning algorithms. 3](#_Toc148706257)

[Description of the Project 4](#_Toc148706258)

[Aim and Objectives 5](#_Toc148706259)

[Key Activities of the Project 5](#_Toc148706260)

[Literature Review 5](#_Toc148706261)

[Systems Analysis and Design 5](#_Toc148706262)

[Implementation and Testing 5](#_Toc148706263)

[Evaluation 5](#_Toc148706264)

[Project Plan (Gant Chart) 6](#_Toc148706265)

[Project Resources 6](#_Toc148706266)

[Bibliography 6](#_Toc148706267)

[References 6](#_Toc148706268)

# Background of study

* Saw a family member taking a long time to manually extract receipt data.
* Big loss of time.
* Repetitive work.

# Problem Statement

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

The thesis discusses an experiment conducted with a focus on information extraction from scanned invoice/receipt images using various technologies and techniques. Here is a list of technologies used and not used, along with their benefits and drawbacks:

**Used Technologies:**

1. **YOLO (You Only Look Once) version 5:** This machine learning algorithm is employed for text detection/localization and classification in scanned receipts.
   * **Benefits:** YOLO is a powerful object detection algorithm known for its speed and accuracy.
   * **Drawbacks:** Training YOLO models for many classes may require extensive computational resources and time.
2. **OCR (Optical Character Recognition):** OCR is used to extract text from scanned receipt images.
   * **Benefits:** OCR is an efficient way to convert scanned text into machine-readable text.
   * **Drawbacks:** OCR accuracy can be affected by image quality and layout complexity.

**Not Used Technologies:**

1. **Format-Agnostic Techniques:** The thesis suggests using format-agnostic techniques, but it doesn't provide specific details about implementing them.
   * **Benefits:** Format-agnostic techniques could potentially make the system more adaptable to various document layouts.
2. **Cloud Server (Google Compute Engine):** The thesis recommends training a multi-class text detection model on a robust cloud server.
   * **Benefits:** Cloud servers offer scalable computing power for intensive tasks, potentially reducing training time.

### Computer Vision for Document Image Analysis and Text Extraction

The thesis focusses on the Optical Character Recognition (OCR) to extract text information from real-world images, especially handwritten documents, and license plates. It seeks to improve OCR accuracy and generalization, primarily using synthetic data and different OCR architectures.

**Technologies Used**:

* **PyTorch under Python**: Used for developing and training OCR models.
* **Tesla K80 GPU**: Employed for neural network training acceleration.
* **Canny's Edge Detector and Hough Transform**: Used for orientation correction in real-world images.
* **Shallow CNN+LSTM architecture**: Utilized for OCR of printed text. Benefits include lightweight and faster inference.
* **Deep CNN + Transformer architecture**: Deployed for OCR of handwritten text, which is more complex. Benefits include improved accuracy on complicated cases.

**Technologies Not Used**:

* **Advanced OCR techniques**: The thesis primarily focuses on different architectures and data sources, so it does not explore advanced OCR methods, such as deep learning models like BERT for text recognition.
* **Hardware Accelerators**: The thesis uses a GPU for neural network training, but it does not delve into the use of dedicated hardware accelerators like TPUs or FPGAs, which can further speed up inference.
* **OCR Post-processing**: While the thesis discusses preprocessing, it does not extensively cover post-processing techniques, which can be crucial for improving OCR results.

**Benefits**:

* The use of synthetic data allows for scalable training data generation, reducing the need for extensive manual labelling.
* Experimentation with different OCR architectures helps in identifying the most suitable model for specific tasks.
* The use of GPU accelerates the training process, making it faster and more efficient.
* Addressing the quality variance in real-world data through opening operations demonstrates the adaptability of synthetic data.

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

The thesis is focused on receipt data processing, classification, and information extraction using machine learning. It aims to explore the possibilities of classifying receipt categories and extracting specific data points, such as price and date, from OCR-processed receipt images.

Technologies Used:

* Optical Character Recognition (OCR): OCR technology is employed to convert printed text on receipts into machine-readable data.
* Azure Computer Vision API
* Google Drive REST Api
* Tesseract

A close-up of a data sheet

Description automatically generated

Overall, the second thesis highlights the potential of machine learning in processing and managing receipt data, but it also underscores the importance of addressing challenges related to OCR accuracy, data variability, and model generalization.

# Description of the Project

* A smart receipt manager and extractor
* A service provider within:
  + Database
  + Receipt detection model ?
  + OCR extraction and processing
* A mobile application:
  + Upload picture or pdf.
  + Take a picture from the application.
  + Correct extraction error before inputting into the database

Feature:

* Server side:
  + Text extraction model
  + Image processing for model
  + Pdf extraction (conversion to image, separate multiple receipt)
  + Generation of report
* Mobile application side:
  + Upload picture to server side for processing.
  + Upload pdf to server side for processing.
  + Receive and display old receipt data from servers database.
  + Receive extracted data from model.
  + Correct received data if needed.
  + Display analytics about the user purchase.
  + Export data to csv and excel.
  + Send exported data by email.
  + Edit exported format.

## Aim and Objectives

[aim: what is the project and purpose.

Objective: what is the action to realize the aims of the project in point / not text (objective and milestone will overlap]

Aim:

Make a receipt data extraction and management to allow to an affordable and easy to use.

Objective:

* Gather receipt data for model training.
* Develop model.
* Train model.
* Test model.
* Develop UI front end.

# Key Activities of the Project

## Literature Review

## Systems Analysis and Design

Separate in two parts:

* Mobile Application :

For user.

Send receipt to model for data extraction.

User data validation of model result.

Purchase analytics.

* Server side :

Contain model for data extraction.

Contain database.

Send / Receive data to the mobile app.

## Implementation and Testing

Explain simply how I plan in what order to make the project.

Testing -> different features of the app, server, UI testing

Different validation options will depend on the dataset size.

Holdout Validation:

* Split your dataset into two parts: a training set and a validation set.
* Train your YOLO-based OCR model on the training set and validate its performance on the validation set.
* This method is straightforward and is a good starting point for model validation.

K-Fold Cross-Validation:

* Divide your dataset into 'k' subsets (folds) where 'k' is typically 5 or 10.
* Train and evaluate your YOLO model 'k' times, each time using a different fold as the validation set and the remaining folds for training.
* This helps assess how well your model generalizes to different data splits and can provide more robust performance estimates.

Bootstrapping:

* Create multiple random subsets of your data through bootstrapping (random sampling with replacement).
* Train and validate your YOLO model on these subsets to estimate performance variability.
* Useful for understanding how stable your model's performance is with different data samples.

## Evaluation

[ check meeting those objects, and respecting my guideline, do a critical assement of my system]

# Project Plan (Gant Chart)

Meeting after each milestone

# Project Resources

Hardware:

* Smartphone for UI and camera
* Server in best case otherwise locally on my computer

Software:

* Database : Postgres Locally, otherwise database within platform use (redshift if amazon…)
* Code IDE : Visual Studio Code
* Flutter (or React Native)
* Postman

Data:

* Reicept image

# Bibliography

# References