Problem Statement

Given a dataset of images organized into multiple folders based on the type of documents they represent, the primary objective is to extract and analyze text. The first task involves converting these images into text using Optical Character Recognition (OCR) technology. Multiple OCR tools will be evaluated to determine the most effective one for accurate text extraction. Once the text is extracted, it will be preprocessed using Natural Language Processing (NLP) techniques, followed by the application of text analysis methods such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging. A comprehensive report detailing the findings and evaluations from this task is required.

The second task focuses on leveraging a language model (LLM) to summarize the extracted text. These summaries will be stored in both a SQL database and a vector database, allowing retrieval through semantic searches and keyword queries. Various embedding models will be implemented to facilitate these searches, and a report on their effectiveness will be provided. This approach aims to optimize both text processing and retrieval, enhancing the accessibility and utility of the extracted information.

Solution 👍

First task is to take an image from the folder and extract the image text using different different ocr technique the basic and first most in code of the solution contain the jupyter notebook that

Folder Name ⏭ notebook/ocr_notebook

Contain 4 notebook

- image_analysis.ipynb(contain the image analysis of the dataset)
- pytesseract_ocr.ipynb (it contain the overall implementation of the pytesseract to extract the image text)
- easy_ocr.ipynb (contain the overall implementation of easy_ocr)
- langchain_image.ipynb (contain all the implementation of langcahin unstructured image loader based on the top of pytesseract image analysis)

This above analysis will be present in the word file present in the same folder

Basics content of the each notebook are

- Dependency and the installation of the ocr tool
- The basic implementation to understand the working of the tool
- And the modular code to implement that function on top of the image

Utils Functions

Get_random_image_path()

As the dataset was too huge so to test the image analysis on each image would have been very difficult . So to test the ocr tool we would randomly choose one image path from the dataset directory and perform all the tools stated in the ocr tool analysis on that image and if found something that would have missed in the initial analysis then that can be refactored in the early stage of development process .
So this function takes the folder path in list and then randomly give out the image path .

display()
This is basically an function that used to view an image in the notebook to and do some preprocessing on the image if needed

@time_it()

It is a decorator function used on the function to get the time it takes to implement its logic and get output.

Folder Name ⏭ notebook/nlp_notebook

It contain two notebook for now

- Optamized_image_analysis.ipynb(it contain the code to do the image analysis for preprocessing the image)

- nlp_feature_extraction.ipynb (this notebook contain the most around the text preprocessing and feature extraction)

Folder Name ⏭ notebook/llm

It contain only one notebook for now

- antropic_llm.ipynb (it contain code code to use antaopic_llm to do the summary part of the text)

Plan to implement the complet code can be explained in the diagram

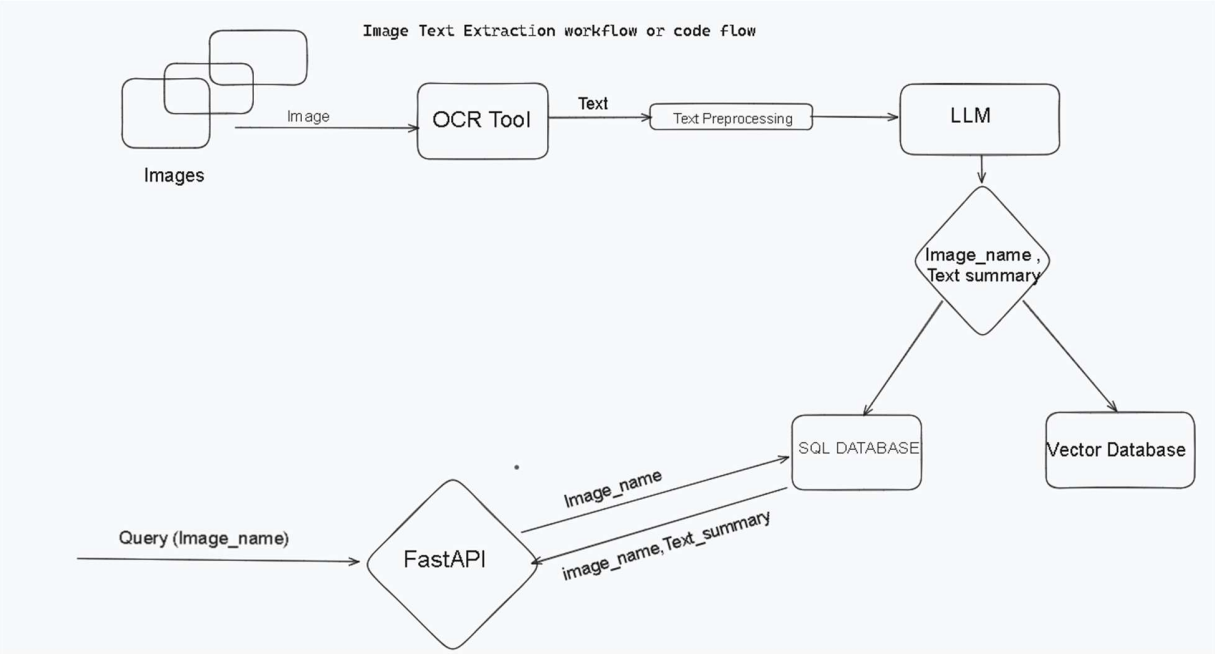

Image Text Extraction workflow

Regarding the Vector Database, I need some time to get what actually and how can I implement it . I have never worked with image embedding .
And with the Task 3 will also be there in the next report

Note 👍

The above code is implemented in such a manner such that each ocr can be used as a service and the project structure will be modular .