Deccan AI Submission:

By: Tanishq Selot (<https://github.com/tanishq150802>)

Task 1:

1. The steps involved in preprocessing the images were as follows:
   1. Converting to grayscale
   2. Applying Gaussian blur – Using higher dimension kernel gave worse results.
   3. Thresholding – Basic Thresholding gave better results than adaptive thresholding.

I also tried Otsu thresholding for scientific papers but the results were similar to basic thresholding.

* 1. I used pytesseract for OCR.
  2. I experimented with Spacy for NER. (Can be seen in experiments – tasks.ipynb)

1. Far better OCR results were obtained using Multimodal large language model (“Gemini-1.5-pro”) which was used for adding into vector database, and later in the chainlit app.

Task 2:

1. I used sqlite3 for SQL DB part for this task. The first preprocessing pipeline discussed above was used here for email image detail insertion into the database.
   1. The database contains filename (primary key), type (eg. Email) and description (OCR text) columns.
   2. “Gemini-1.5-flash” (LLM) model was used to summarize the contents of the inserted emails. Default values for temperature, top\_k and top\_p parameters were considered.
   3. This system was deployed on FastAPI swagger UI (‘/query\_file/’). The email was summarized by entering it’s filename which was used to query the SQL DB.
2. I used chromadb for storing the passage chunks as vectors. The vector DB was created using persistent client to allow saving and reloading. A scientific paper was used for chunking.
   1. “text-embedding-004” was used as the embedding function. The collection is named “rag”.
   2. Top 1 result from the collection is used as the context for LLM. We can use more results to be able to answer more general queries related to the document queried. [RAPTOR](https://github.com/tanishq150802/chat_and_validate) can be a better option for longer but concise context (giving the “larger picture” of the document)
   3. I had tried milvus DB previously but found that it will only work for UNIX-based operating systems.

Task 3:

1. It was fun playing around with chainlit. Surprised by how easy it is to build it. Here’s a demo of me chatting using its chat interface about a scientific paper:

