TEAM SUDO SU

Overview:

Our script starts with processing images to extract entities (such as measurements or product information) using OCR and some Named Entity Recognition (NER) techniques. It stores these entities in ChromaDB for querying and searching similar entities resulting in better outputs.

We came up with this approach accidentally when we stumbled upon the code for another similar project that we were working on and thought that an approach with vector database querying would yield better results.

We started off with normal OCR and a basic Random Forest Classifier but we did not find the outputs in the csv file to be similar to the actual results we got.
Then after lots of trial and error with various machine learning models, we decided to approach this problem using a vector database that will essentially do the same thing but in a more efficient way.

We found that using a vector database can be efficient while doing similarity searches since they are optimised for finding similar items based on distance metrics. This can be essential for quick searches.
We also avoid the hassle of training and fine tuning the machine learning model which can save lots of time and compute power which can be diverted to improving the existing OCR methodology for better feature extraction.
Also, vector databases can handle large datasets and generalise well with no problem of overfitting in the process.

Libraries Imported:

We start off with importing some necessary libraries for the purposes listed down below:

1. Image Processing using **PIL and OpenCV**.
2. OCR using **pytesseract**.
3. Basic Libraries like **numpy, os and pandas**.
4. Regular Expression matching using **re**.
5. Entity Recognition using **spacy and transformers**.
6. Handling entity vectors using **chromadb and sentence_transformers**.
7. For calculating string similarity we make use of **fuzzywuzzy**.

Descriptions of the Functions implemented:

`preprocess_image(image_path):`

- Takes an image path as input.
- Converts the image to grayscale, denoises it, and applies thresholding.
- Returns a preprocessed image as a PIL object.
- **Libraries Used**: `cv2`, `Image`.

`extract_text_from_image(image_path)`:

- Takes the image path as input.
- Preprocesses the image and extracts text using OCR (`pytesseract`).
- Returns the extracted text.
- **Libraries Used**: `pytesseract`, `cv2`.

`extract_entities(text)`:

- Takes a block of text as input.
- Uses regular expressions to identify numeric values with associated units (e.g., "10 kg").
- Uses `spacy` and a BERT-based NER pipeline to detect named entities in the text.
- Returns a dictionary of extracted entities.
- **Libraries Used**: `re`, `spacy`, `transformers`.

`store_entities_in_chroma(entities, image_filename)`:

- Takes a dictionary of entities and an image filename.
- Vectorizes the entities using Sentence-BERT.
- Stores the entities and their vectors in a ChromaDB collection.
- **Libraries Used**: `SentenceTransformer`, `chromadb`.

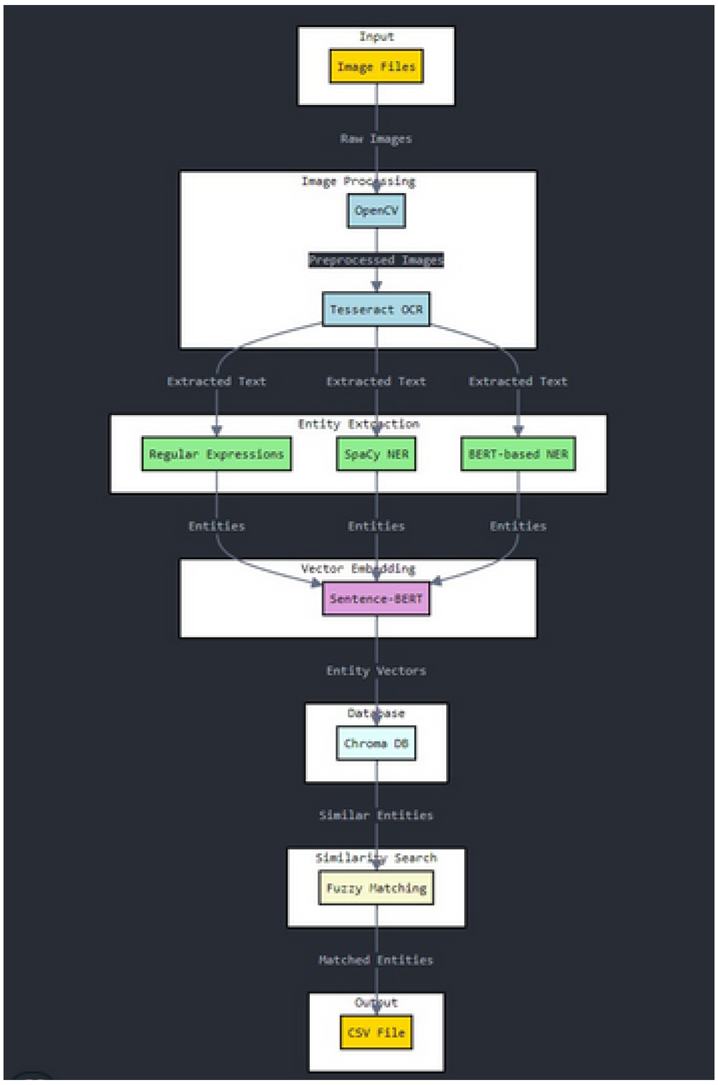
`search_similar_entity_in_chroma(query)`:

- Takes an entity as a query.
- Vectorizes the query and searches for the most similar entities in ChromaDB.
- Uses fuzzy string matching to determine the best match.
- Returns the best match and its similarity score if it exceeds a predefined threshold.
- **Libraries Used**: `fuzzywuzzy`, `chromadb`.

`create_output_csv_from_images(image_folder, output_csv, max_images)`:

- Takes a folder of images, processes each image to extract text, entities, and store the entities in ChromaDB.
- Compares entities from each image to similar ones in the database and stores predictions in a CSV.
- Limits processing to a maximum of max_images. This was something that we used while testing the code quickly. It can be removed.
- **Libraries Used**: `os`, `pandas`, `cv2`.

Architecture Diagram:


Input
Image Files
Raw Images
Image Processing
OpenCV
Preprocessed Images
Tesseract OCR
Extracted Text
Extracted Text
Extracted Text
Entity Extraction
Regular Expressions
SpaCy NER
BERT-based NER
Entities
Entities
Entities
Vector Embedding
Sentence-BERT
Entity Vectors
Database
Chroma DB
Similar Entities
Similarity Search
Fuzzy Matching
Matched Entities
Output
CSV File