Approach Overview

This project presents a robust and scalable method for extracting text from images and performing entity mapping using a large language model (LLM). The process involves image preprocessing, text extraction using EasyOCR, and data enhancement by applying a transformer-based model to the extracted text for accurate entity identification and mapping.

1. Image Handling and Preprocessing

- Images were loaded into the system, ensuring appropriate resolution and format for optimal functioning with the EasyOCR model.
- Each image was downloaded locally, allowing for faster access and minimizing network overhead during processing.
- Post-processing, the images were deleted to conserve storage and enhance efficiency.

2. Text Extraction with EasyOCR

- The EasyOCR model was employed to extract text from the locally stored images, benefiting from GPU resources to accelerate the computation.
- This GPU-powered approach significantly reduced the processing time for complex and large-scale images.

3. Data Preparation and CSV Generation

- The extracted text was combined with pre-existing data into a consolidated CSV file. This CSV file contains both the initial dataset and the `text_img` column, which holds the text extracted from images.
- This dataset now serves as the input for subsequent processing steps, feeding directly into a Large Language Model (LLM) for entity extraction.

4. Entity Mapping and Processing with LLM

- The combined dataset, containing both the initial data and the extracted text, was processed using the LLaMA model, a transformer-based architecture fine-tuned for entity mapping tasks.
- The model performed token classification, identifying key entities such as width, depth, height, weight, and voltage, based on patterns recognized from the text.
- This step was executed using the BERT tokenizer to align input text with predefined entity categories, ensuring the text was properly annotated for model training and prediction.

5. Model Training and Entity Extraction

- The dataset was split into training and validation sets. These were tokenized and fed into the LLaMA model for token classification.

- The model was trained over several epochs, using GPU resources for efficient parallel processing.
- After training, the model was able to predict and classify entities with high accuracy, mapping values such as dimensions, weight, and other attributes to corresponding units.
- The final trained model was saved for future inference on unseen data.

6. Inference and Prediction

- The trained model was then used to make predictions on a test dataset. Each text entry in the test dataset was processed to predict relevant entity values, which were subsequently mapped to their respective units (e.g., “width in inches” or “weight in kilograms”).
- The predictions were formatted and stored in a new CSV file for further analysis or integration into downstream applications.

7. Output Generation

- The final output consisted of a structured and clean dataset where entities such as width, height, weight, voltage, and others were identified and mapped to their respective units.
- The results were saved as a CSV file for use in further analytical tasks or decision-making processes.