Approach Overview

This solution presents a robust and scalable method for extracting text from images and performing entity mapping using a large language model (LLM) as part of a hackathon problem statement. The key stages in the process are 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 with careful attention to resolution and format for optimal performance with the EasyOCR model.
    - Each image was downloaded locally to ensure faster access and reduce network overhead during the processing phase.
    - After processing, the images were deleted to conserve storage and improve system efficiency.
2. **Text Extraction with EasyOCR**
    - The EasyOCR model was used to extract text from locally stored images, leveraging GPU resources to speed up computation.
    - This GPU-accelerated approach significantly reduced the processing time, even when working with complex or large-scale images.
3. **Data Preparation and CSV Generation**
    - The extracted text was consolidated with pre-existing data into a CSV file. This file contained both the original dataset and a `text_img` column, where the text extracted from images was stored.
    - This CSV dataset served as the input for further processing, leading into a Large Language Model (LLM) for entity extraction.
4. **Entity Mapping and Processing with LLM**
    - The combined dataset, which included both initial data and the extracted text, was processed using the LLaMA model, a transformer-based architecture designed for entity mapping.
    - The model executed token classification, identifying key entities such as dimensions (width, depth, height), weight, and voltage, based on patterns in the text.
    - Input text was aligned with predefined entity categories using the BERT tokenizer, ensuring accurate annotation for model training and prediction.
5. **Model Training and Entity Extraction**
    - The dataset was split into training and validation sets, tokenized, and then processed by the LLaMA model for token classification.
    - GPU resources were used for efficient parallel processing during training, allowing the model to classify entities with high accuracy after several training epochs.
    - The trained model was saved for future inference on unseen data.
6. **Inference and Prediction**
    - The trained model was used to make predictions on a test dataset, identifying relevant entity values for each text entry.

- These predicted entities were mapped to appropriate units (e.g., "width in inches" or "weight in kilograms").
- The predictions were formatted and saved in a new CSV file for further analysis or integration into other applications.

7. Output Generation

- The final output was a clean, structured dataset where entities like dimensions, weight, and voltage were accurately identified and mapped to the correct units.
- These results were stored in a CSV file, ready for further analytical tasks or decision-making processes.