**Machine Learning Approach and Models Used**

**Team Name: ML Hooks**

**Team Members:**

* **Anubhav Vashishtha**
* **Manhvi Yadav**
* **Arnav Gupta**
* **Sarthak Agrawal**

**Problem Statement**

The goal of this approach is to extract specific entity values, such as weight, volume, and dimensions, from product images using Optical Character Recognition (OCR) and Named Entity Recognition (NER). These entities are essential for applications like e-commerce, where detailed product specifications are required. The core components involve processing images, extracting text, identifying relevant entities, and mapping them to predefined units.

**Workflow Overview**

1. **Image Preprocessing**: The images are downloaded using the URLs provided in the dataset and stored locally. The OCR engine is then applied to extract any text from these images.
2. **OCR (Optical Character Recognition)**: The PaddleOCR engine is used to extract text from the product images. The extracted text is processed and cleaned to prepare it for the next stage.
3. **NER (Named Entity Recognition)**: A pre-trained spaCy model (en\_core\_web\_sm) is employed for identifying entities such as quantities and units from the extracted text. This helps in detecting numeric values associated with measurements (like 10 kg, 5 m, etc.).
4. **Entity Mapping and Unit Normalization**: Once entities (values and units) are detected, units are normalized using a predefined unit map (e.g., cm becomes centimetre). These units are then mapped to relevant entity categories (like weight, length, volume).
5. **Final Output**: The final step involves structuring the extracted entities into a format that can be evaluated. The output consists of predicted values and units for each image, which are saved in a CSV format.

**ML Models and Techniques Used**

* **PaddleOCR**: The OCR engine is responsible for reading text from images. This tool is particularly efficient for recognizing English characters and includes angle detection to correctly interpret skewed or angled text in images.
* **spaCy NER Model**: The `en\_core\_web\_sm` spaCy model is used to detect entities like quantities (NUM) and measurement units in the text. SpaCy is an open-source NLP library that provides accurate NER functionalities, which help in identifying numeric values and their associated labels.

**Source Code Description**

**Key Functions**

1. **clean\_text(text)**: Removes non-alphanumeric characters from the detected text. This is a preprocessing step to ensure cleaner data for NER.
2. **extract\_value\_and\_unit(text)**: Uses a regular expression to extract numeric values and associated units from text. This function returns a tuple containing the value and the unit if detected, or None otherwise.
3. **map\_unit\_to\_entity(unit)**: Maps units (like kg, cm) to their respective entities (e.g., weight, length) using a predefined dictionary (entity\_unit\_map). This is critical for ensuring correct identification of measurement types.
4. **normalize\_unit(unit)**: Converts units to a normalized form (e.g., cm → centimetre). This function allows for more consistent unit handling across images.
5. **apply\_ner(text)**: Utilizes spaCy's NER model to extract named entities from the cleaned text. Entities like quantities and measurement units are identified here.

**Experimentation and Results**

**Experiment Setup**

The dataset consists of images containing product information, such as dimensions, weight, and volume, in text format. These images are processed using OCR to extract the text, which is then analysed using spaCy's NER to detect values and units.

**Performance and Results**

OCR successfully extracted text from the images, and the NER model was able to detect values like weight and dimensions from the extracted text. The output was structured into a CSV file containing the predicted values and their associated units for each image.

The results were stored in the predictions.csv file, which contains two main columns:

* **Index**: A unique identifier for each prediction.
* **Prediction**: The extracted value and unit for each image.

The output was evaluated using metrics like precision, recall, and the F1 score. The extraction process showed promising results for images with clear and distinct text but faced challenges with noisy or low-quality images.

**Conclusion**

The proposed method combines OCR for text extraction and NER for entity recognition to successfully identify and extract values from product images. Future improvements could involve using a more advanced NER model, such as a custom-trained one for units and quantities, and improving OCR handling for low-quality or complex images. The model shows promise in automating data extraction for applications like e-commerce and healthcare.