Wizzers - Amazon ML Challenge

1. Introduction

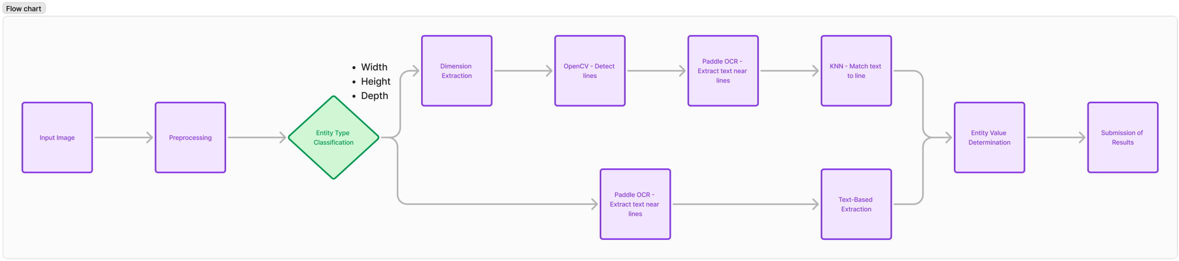
In this hackathon, our objective was to develop a machine learning model capable of extracting entity values from product images. This capability is crucial in fields such as e-commerce, healthcare, and content moderation, where extracting precise product information like weight, volume, and dimensions is essential for operations like online product listings.

2. Approach Overview

Our approach is hybrid, combining Optical Character Recognition (OCR) and image processing techniques with traditional machine learning methods. The solution extracts entity names and values directly from images while handling the complex cases for physical dimensions such as width, depth, and height.

The workflow can be summarized as follows:

1. **OCR-based Text Extraction**: For most entity types, we utilized PaddleOCR to extract text from images.
2. **Rule-Based Extraction**: A series of regular expressions (regex) was crafted to match and extract entity types and values from the OCR-extracted text. We refined these patterns based on an extensive manual examination of 100 images for each entity to ensure accuracy.
3. **Image Processing for Dimensions**: For the three dimensional attributes—width, depth, and height—we employed image processing techniques using OpenCV to locate horizontal and vertical lines in the images. This helped identify labels for dimensions. We used KNN to match the nearest values extracted using PaddleOCR to these labels.
4. **Test Dataset Splitting and Evaluation**: The test dataset was split into 8 parts, each containing only one class of entities. We tested and submitted predictions separately for each of these classes to improve accuracy.



3. Detailed Methodology

a. OCR Model (PaddleOCR)

PaddleOCR was selected due to its high accuracy in recognizing textual content in images, especially for structured text like product labels. It was applied across all product images to extract potential entity names and values.

b. Regular Expression-Based Extraction

After obtaining text from the images using OCR, we applied entity-specific regular expressions to extract entity names and their corresponding values. For instance:

- **item_weight**: We designed patterns like `r'(\\d+(\\.\\d+)?\\s*(gram|kilogram|ounce|pound))'` to capture weights.
- **voltage**: Regex like `r'(\\d+(\\.\\d+)?\\s*(volt|kilovolt|millivolt))'` was used for voltage values.

Each pattern was fine-tuned by manually examining around 100 images per entity type, ensuring that our rules could handle the variability in image formats.

c. Handling Dimensions (Width, Depth, Height)

Extracting dimensions (width, depth, and height) from images posed a unique challenge. These values are often visually associated with lines and labels in the images. To address this:

- We used **OpenCV** to detect horizontal and vertical lines, which commonly represent boundaries or scales in images for physical dimensions.
- Once lines were identified, we applied **K-Nearest Neighbors (KNN)** to find the text closest to these lines, likely representing the dimension names (e.g., width, height, depth) and their corresponding values.

- Finally, PaddleOCR was used again to read the nearest text, and regex was applied to extract the numerical values with the correct units (e.g., `inch`, `centimetre`).

d. Test Dataset Strategy

To further optimize performance, we split the test dataset into 8 smaller datasets, each corresponding to one class of entities (e.g., item_weight, item_volume, width, depth, height). This approach allowed us to run specialized models and regex patterns for each class, improving both efficiency and accuracy. Each split was tested separately, and submissions were made based on these class-specific evaluations.

4. Challenges & Solutions

1. **Dimensional Extraction**: Extracting dimensions was challenging as it required understanding both the textual content and the image structure. Our use of OpenCV for line detection and KNN for nearest text association was key to solving this problem.
2. **Regex Tuning**: Designing regex patterns that could handle a wide variety of unit and entity formats was difficult but crucial. By manually reviewing hundreds of images, we were able to craft robust patterns.

5. Results

Our final model was evaluated based on F1 score. The combination of PaddleOCR, regex-based extraction, and image processing techniques provided accurate results, especially for entities like weight, volume, and dimensions.

6. Conclusion

Our hybrid approach—combining OCR, regex-based extraction, and image processing—proved effective for this challenge. By splitting the dataset and tailoring our methods to specific entity types, we were able to handle the diversity of images and formats, delivering a solution that extracted precise product information from images.