

# Amazon ML Challenge

Team Zen ji

## 1 Problem Overview

The task is to extract entity values (e.g., weight, dimensions) from product images using machine learning, targeting applications in fields such as e-commerce and healthcare. The extracted entities are required to match specific units and follow a strict format. The evaluation is based on the F1 score, considering precision and recall.

## 2 ML Approach

### 2.1 Image Preprocessing

Images were fetched and preprocessed to enhance OCR accuracy. Simple methods like resizing and grayscale conversion were applied to standardize the inputs for further processing.

### 2.2 OCR (Optical Character Recognition)

We employed **EasyOCR**, which proved effective for extracting textual data from images. Pre-trained models in EasyOCR facilitated recognizing numeric values and units. The extracted text was cleaned and normalized to standardize unit formats, converting abbreviations like "kg" to "kilogram".

### 2.3 Post-processing

Extracted text was processed using **regular expressions** to identify numeric values and units. Custom logic matched the values to entity types (e.g., item weight, item volume), and any invalid units were filtered out.

### 2.4 Formatting Predictions

For each image, the final output was formatted as "numeric value + unit" (e.g., "2.5 kilogram"). Empty strings were returned when no valid entity could be extracted. The mapping to valid units adhered to the predefined rules specified in the constants file.

## 3 Experiments

- **OCR Accuracy:** Tested various image preprocessing techniques to boost OCR success rates.
- **Regex Matching:** Enhanced regex patterns to better match numerical values and unit combinations.
- **Entity Mapping:** Experiments with handling ambiguous dimensions and weights.

## 4 Conclusion

Our solution successfully extracted numerical entity values from images by leveraging EasyOCR and regex-based post-processing. The key challenge was handling inconsistencies in image quality, which we mitigated through preprocessing. Future improvements could include using advanced image processing techniques or vision-language models to better handle noisy or low-quality images.

## 5 Code Overview

### 5.1 Core Functions

- `extract_ocr_text`: Performs OCR on the image and cleans the extracted text.
- `normalize_short_forms`: Converts unit abbreviations to their full forms.
- `match_entity_to_ocr`: Applies regex to match extracted text to entity types.
- `predict`: Runs the full entity extraction process on the given image.

### 5.2 Main Pipeline

The code iterates through the test dataset, calls the `predict` function for each image, and outputs a CSV file with the predictions. Key functions are well-commented for clarity and extensibility.

The code is designed with modular functions, and each function is documented to explain its input, output, and role within the overall pipeline.