Code Explanation for Submission

This document explains the code used in the provided script, which processes a dataset (likely related to product images) using Optical Character Recognition (OCR) to extract and normalize product attributes such as dimensions, weight, voltage, and more. The code leverages the PaddleOCR library for text extraction and uses Pandas for data handling.

# Key Libraries Used

1. \*\*paddleocr\*\*: Used for Optical Character Recognition (OCR) to extract text from images.  
2. \*\*OpenCV (cv2)\*\*: For image processing, including resizing images.  
3. \*\*Requests\*\*: Handles HTTP requests to download images from URLs.  
4. \*\*NumPy and Pandas\*\*: Essential for numerical and tabular data manipulation.  
5. \*\*PIL (Python Imaging Library)\*\*: Converts and processes images in different formats.  
6. \*\*Joblib\*\*: Enables parallel processing, improving performance.

# Main Workflow

The main functionality is structured around extracting text from images (either from URLs or locally) using PaddleOCR.

## Image Processing and OCR

- \*\*upscale\_image(image, scale\_factor=2)\*\*: This function enhances image resolution using OpenCV, improving OCR results.  
- \*\*download\_image\_from\_url(url)\*\*: Downloads images from URLs and converts them for OCR.  
- \*\*extract\_text(image\_path, scale\_factor=2)\*\*: This function performs OCR, optionally upscaling images for better results.

## Parallel Processing

The code uses \*\*joblib.Parallel\*\* to enable multiple images to be processed simultaneously, reducing the time taken for OCR.

# Data Processing

## Pattern Matching and Normalization

Regular expressions and unit normalization ensure that product attributes such as weight, dimensions, and wattage are extracted consistently.  
- \*\*extract\_quantity(row)\*\*: This function extracts and normalizes values from the OCR-extracted text using regular expressions.

# DataFrame Manipulation

After the text extraction, the script processes the dataset using Pandas. It removes irrelevant columns, applies the extraction functions, and creates a final CSV file (`submission\_1.csv`) containing normalized predictions.

# Error Handling

Errors during image download or OCR are handled gracefully to ensure that the processing continues without interruption.

# Conclusion

This script efficiently automates the process of extracting and normalizing product attributes from images. It combines OCR, image processing, and parallel computation with Pandas to handle large datasets with ease.