The following project utilizes EasyOCR for text extraction and processes measurement units using Regex and Fuzzy word matching. The detailed workflow and code implementation are outlined below:

**Project Workflow:**

**Text Extraction Using EasyOCR**

- **Objective:** Extract text from images, particularly numerical data with units.
- **Steps:**
    1. **Loading the Image:** Downloads and loads images using PIL from a given URL.
    2. **Pre-Processing:**
        - **Grayscale Conversion:** Simplifies text extraction by converting images to grayscale.
        - **Inversion Check:**If the image has a dark background and light text, invert the image for uniform OCR processing i.e converting it into light background with dark text.
        - **Sharpening:** Enhances image sharpness and improves OCR performance.
    3. **OCR Processing:** EasyOCR is used to extract text, identifying numbers and measurement units from the image.

**Unit Identification Using Regex and Fuzzy Matching**

- **Objective:** Correctly identify and normalize unit abbreviations that may be incomplete or misspelled.
- **Steps:**
    1. **Regex Matching:** Extracts numbers and unit abbreviations from the OCR text.
    2. **Fuzzy Matching via rapidfuzz:** Matches possible unit abbreviations against a predefined dictionary to handle misspellings and similar errors.
    3. **Unit Conversion:** Converts identified abbreviations into their full canonical forms (e.g., 'cm' becomes 'centimetre').

**Processing and Conversion to Base Units**

- **Objective:** Convert extracted units into base forms, such as millimeters for length and milligrams for weight, to maintain consistency and comparability, where we find the most optimal value after which we convert it back to the original unit and display the results.
- **Steps:**
    1. **Categorization:** Classifies extracted units into categories such as length, weight, or volume.
    2. **Unit Conversion:** Uses predefined conversion factors to convert extracted numbers into base units.
    3. **Max Value Detection:** Detects the largest value for base units in each measurement category.
    4. **Dimension Comparison:** Compares the dimension that was extracted and differentiates between width, height and depth.

**Challenges & Solutions:**

- **OCR Accuracy:**
    - **Problem:** Variations in image quality or text formatting led to inaccurate extraction.
    - **Solution:** Applied pre-processing techniques like grayscaling, image inversion, and sharpening to improve text recognition accuracy.
- **Handling Unit Abbreviations:**
    - **Problem:** Extracted units sometimes contained various abbreviations or misspellings.
    - **Solution:** Used fuzzy matching techniques with rapidfuzz to ensure extracted abbreviations matched known units.

- **Complex Regex Patterns:**
    - **Problem:** Flexibly identifying numbers and units in various formats was challenging.
    - **Solution:** Iteratively refined regex patterns and fuzzy matching for more flexibility and accuracy in unit identification.
- **Conversion & Standardization:**
    - **Problem:** Managing various unit conversions into a standardized format was complex.
    - **Solution:** Implemented a dictionary of units and conversion factors to automate conversions efficiently.

**Code Overview:**

- **Text Extraction & Preprocessing:** Images are loaded and pre-processed using PIL and OpenCV. This includes converting to grayscale, inverting if necessary, and applying sharpening filters. EasyOCR is then used to extract text, focusing on numbers and unit abbreviations.
- **Unit Identification & Conversion:** Regex is applied to detect number-unit pairs, and fuzzy matching via rapidfuzz helps correct unit abbreviations. The identified units are converted to standardized full forms using a predefined dictionary.
- **Final Processing:** Extracted units are categorized (e.g. dimensions, weight, volume) and converted into base units (e.g., millimeters, milligrams) to maintain consistency for comparison. Maximum values are detected in each category.