ML Challenge Problem Statement

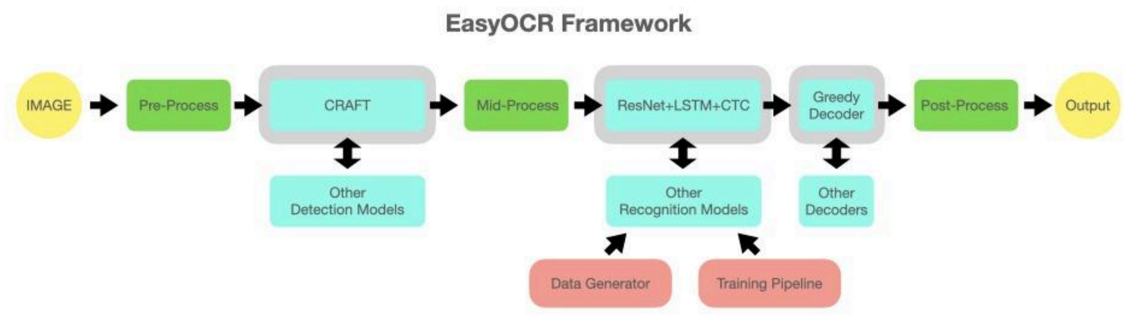
Feature Extraction from Images

In this challenge, the goal is to create a machine learning model that extracts entity values from images. This capability is crucial in fields like healthcare, e-commerce, and content moderation, where precise product information is vital. As digital marketplaces expand, many products lack detailed textual descriptions, making it essential to obtain key details directly from images. These images provide important information such as weight, volume, voltage, wattage, dimensions, and many more, which are critical for digital stores.

Key Technologies

1. Optical Character Recognition (OCR):is a technology that converts images of text into machine-readable text. It uses image processing and pattern recognition techniques to identify individual characters and words in an image. OCR is commonly used for tasks like scanning documents, digitising handwritten text, and extracting information from images.

OCR used here is EasyOCR:
EasyOCR is a Python library that simplifies the process of extracting text from images. It's designed to be user-friendly, even for those without a background in OCR or computer vision. EasyOCR supports a wide range of languages and writing scripts, making it a versatile tool for various applications.



2.PyTorch: PyTorch is a popular open-source machine learning framework widely used for deep learning applications. It provides a flexible and dynamic computational graph, making it easy to experiment and prototype new models.

3.Deep Learning: Deep learning is a subset of machine learning that uses artificial neural networks with multiple layers to learn complex patterns from data. Deep learning models can automatically extract features from raw data, making them highly effective for tasks such as image recognition, natural language processing, and speech recognition.

4.Computer Vision: Computer vision is a field of artificial intelligence that deals with enabling computers to understand and interpret visual information from the real world. It involves processing and analysing images and videos to extract meaningful information, such as objects, scenes, and actions.

Summary of the Source Code

The code defines a deep learning model for text recognition, combining a ResNet-based feature extractor with a bidirectional LSTM for sequence modelling and a linear layer for prediction.

Key Components:

1. **Feature Extraction**
2. **Sequence Modelling**
3. **Prediction**

The model takes an input image and processes it through the feature extraction stage to obtain visual features. These features are then fed into the sequence modelling stage to capture contextual information. Finally, the contextual features are passed to the prediction stage to generate class probabilities.

Text Recognition Framework

The script implements a robust text recognition system using PyTorch. It involves the following steps:

1. **Preprocessing:** Images are resized, padded, and normalised. Contrast adjustments may be applied for low-quality images.
2. **Text Recognition:** A deep learning model processes images and decodes the output using greedy decoding or beam search.
3. **Confidence Calculation:** A confidence score is assigned to each prediction.
4. **Handling Low Confidence:** Images with low confidence are reprocessed with contrast adjustments.
5. **Final Prediction:** The highest-confidence prediction is selected.

This framework uses a two-pass approach for improved accuracy and handles various image quality challenges.

Text Extraction Script

The Python script utilises the EasyOCR library to extract text from a large number of images. It iterates through a specified directory, processes each image using OCR, and concatenates the extracted text into a single line for each image. The resulting text is then written to an output file.

Key Steps:

1. **Import Libraries:** Imports necessary libraries for OCR and file operations.
2. **Initialize EasyOCR Reader:** Creates an EasyOCR reader for English text recognition.
3. **Specify Output File and Image Directory:** Sets paths for the output file and image directory.
4. **Iterate Over Images:** Loops through all images in the specified directory.
5. **Check File Existence:** Verifies if the image file exists.
6. **Run OCR:** Extracts text from the image using EasyOCR.
7. **Concatenate Text:** Joins extracted text into a single line.
8. **Write to Output File:** Writes the concatenated text to the output file.

Text Measurement Extraction

This script extracts measurement magnitudes from text based on entity types specified in a CSV file. It uses regular expressions to match numerical values with corresponding units and writes the results to separate CSV files.

Key Features:

- **Entity-Unit Mapping:** Associates units with specific entity types.
- **Regular Expression Matching:** Uses regex to find numerical values followed by units.
- **Multiple Output Files:** Generates two CSV files for different unit types.

Workflow:

1. **Read Input Files:** Reads CSV file with entity information and text file.
2. **Extract Entity Type:** Gets the entity type for each line.
3. **Match Measurements:** Uses regex to find measurements based on entity type.
4. **Write Results:** Saves extracted measurements to CSV files.

Output:

- **result.csv:** Measurements with full unit names.
- **possibleresult.csv:** Measurements with possible shorthand unit names.

<u>Conclusion</u>

Overall the code represents a comprehensive text extraction and analysis pipeline. They effectively address various aspects of processing and interpreting text data; these code snippets provide a solid foundation for building applications that involve text processing and analysis tasks, offering flexibility and adaptability for various use cases.