Amazon ML Hackathon

Rajas Bhosale, Kishan Chaudhary, Vishal Bokare, Darshan Biradar

Machine Learning Approach:

1. Objective

This code's objective is to extract and normalise features related to images (weight, volume, wattage, dimensions, etc.) from images that are listed in a CSV file. Textual data is extracted from the images and used to classify entity values according to predefined units such as "cm", "kg", "inch", and so on.

2. Dataset and Input Structure

- The program receives a CSV file as input, with each row representing an image and associated metadata (entity names, for example).
- The photos are kept in the designated dataset folder. Every image has textual data that needs to be extracted about an entity (weight, width, etc.).

3. OCR Engine

- **Text from images is extracted using EasyOCR. English language support is included when the OCR engine is first started (easyocr.Reader(['en'])).**
- When the GPU is available, the code automatically detects it and uses it, which increases processing efficiency for large batches of images.

4. Regular Expressions for Feature Extraction

- To capture feature units and the numerical values that go along with them, a regular expression pattern is defined.
- The pattern works with a variety of units, including centimetres, inches, pounds, litres, and so on. It matches both single values (like "10 cm") and float values (like "10.20cm").
- This pattern is used by the extract_features method to retrieve possible feature values from the text that has been OCR-extracted.

5. Unit Normalization

- Various notations can be used to represent units taken from text, such as "cm" for centimetres and "kg" for kilogrammes. A normalisation dictionary is used to standardise this.

- The normalize_units method maps units to a normalised form based on the type of entity (weight, length, volume, etc.). This makes it possible for the extracted features to be consistent, which facilitates processing and categorisation.

6. Entity-Unit Mapping

- Specific allowable units are linked to entities such as "width", "depth", "height", "item_weight", etc. For example, the weight can be expressed in "kg", "g", "lb", and so on, while the width can be expressed in "cm", "foot", "inch", and so on.
- When generating the output, this mapping makes sure that only valid units for each entity are taken into account.

7. CSV Processing in Batches

- Images are processed in batches to ensure efficient handling of large datasets. Up to 50 images can be included in each batch, which helps control memory consumption during data processing.
- For each image in the batch:
    - The OCR engine reads the image and extracts the text.
    - The extract features method receives the text after that in order to determine possible measurements.
    - The normalize_units method is used to normalise the identified features according to the entity type.
    - Measurements are written to the output CSV file if they are found to be valid.

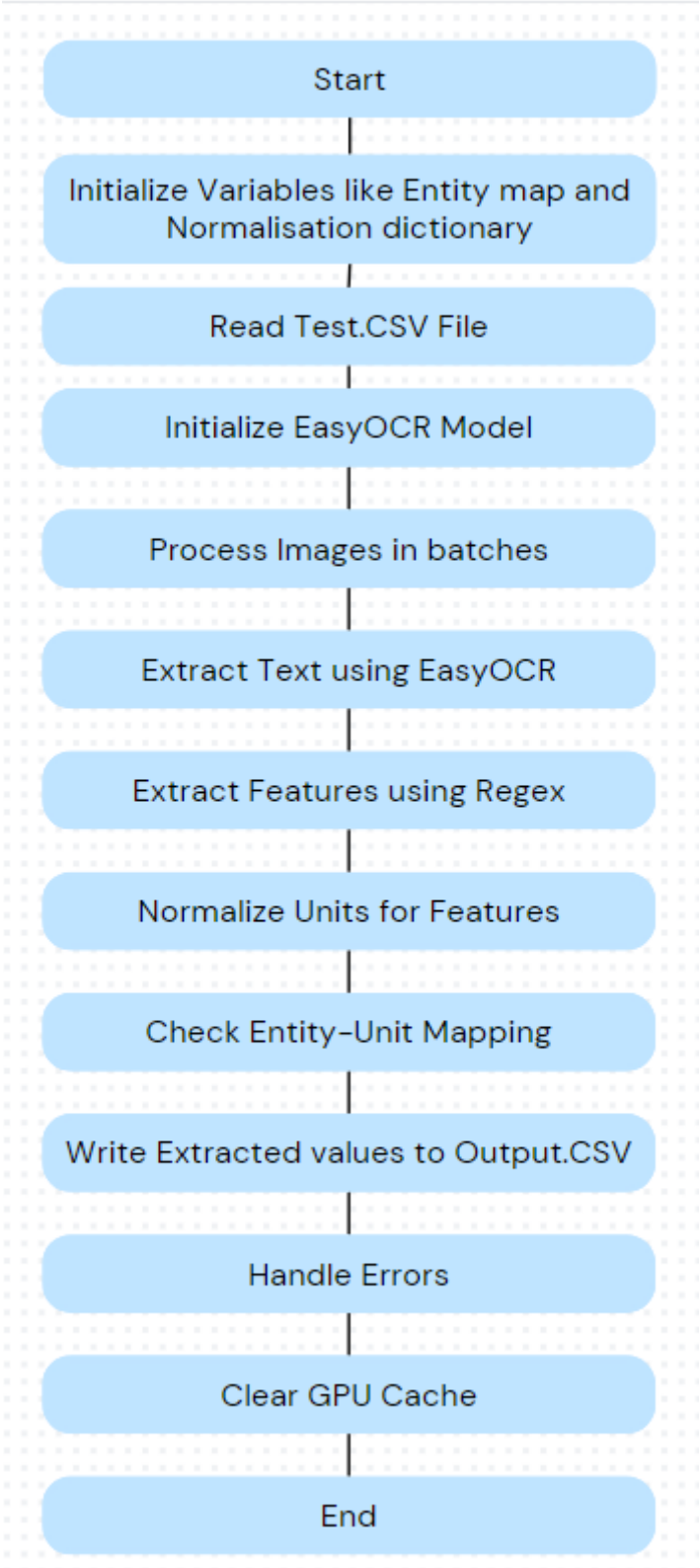
8. Output Structure

- A new CSV file called test_out.csv is created, with two columns: the extracted value and the image index.
- An empty value appears in the corresponding row when no valid features are extracted from the image.

9. Error Handling and GPU Memory Management

- When an image processing error occurs (such as when an image cannot be processed or is not found), the program logs the error and moves on to the next image.
- Torch.cuda.empty_cache() optimises memory usage by clearing the GPU cache after each batch is processed.

Flowchart:


Start
Initialize Variables like Entity map and Normalisation dictionary
Read Test.CSV File
Initialize EasyOCR Model
Process Images in batches
Extract Text using EasyOCR
Extract Features using Regex
Normalize Units for Features
Check Entity-Unit Mapping
Write Extracted values to Output.CSV
Handle Errors
Clear GPU Cache
End