AI PIONEERS

FEATURE EXTRACTION FROM IMAGES

The goal of this problem is to develop a machine learning model that can accurately extract entity values, such as weight, dimensions, or volume, directly from product images. This capability is critical for fields like e-commerce, where product listings may lack detailed textual descriptions, but images contain important information needed for product identification and categorization. By extracting these values from images, the system will automate and streamline the process of obtaining precise product details, which are essential for inventory management, search optimization, and enhancing customer experience on platforms like Amazon.

DATASET

1. **Columns**: Includes *`index` (ID)*, *`image_link`* (URL to the product image), *'group_id`* (product category), *`entity_name`* (e.g., "item weight"), and *`entity_value`* (target value, e.g., "34 grams").
2. **Training vs. Test Data**: The training set has labeled *`entity_value`*, while the test set requires predicting these values.
3. **Image Download**: Images are accessible via provided URLs and downloaded using a utility function.
4. **Target Variable**: The prediction format must be "value unit" (e.g., "15 grams").
5. **Allowed Units**: Only specific units are permitted for each entity, as defined in `constants.py`.
6. **OCR Use**: Optical Character Recognition (OCR) can help extract text-based values like weight and dimensions from the images.

TASK

The task is to develop a machine learning model that extracts entity values (such as weight, volume, or dimensions) from product images. This is crucial for e-commerce platforms, where textual information may be lacking, but critical product data can be obtained directly from images. Our approach focuses on leveraging deep learning for image feature extraction, combined with text recognition (OCR) to accurately predict these entity values.

APPROACH

1. **DATA PREPROCESSING**
   - **Image Downloading**: We used the *`download_images`* function provided in `utils.py` to download images from the provided URLs.
   - **Data Augmentation**: To increase model robustness, we applied transformations such as random rotations, horizontal flips, and brightness adjustments to the training images.
   - **Text Preprocessing**: Since we are predicting entity values, we preprocessed text extracted from the images to standardize units (e.g., converting "kg" to "kilogram").

2. **FEATURE EXTRACTION**
   - **OCR (Optical Character Recognition)**: We employed Tesseract OCR to extract text features from images, focusing on numerical values and units (e.g., "30 grams", "5 watts").

- Preprocessing included filtering non-relevant text and converting the text into a uniform format for further model processing.

3. **MODEL ARCHITECTURE**
   - **Convolutional Neural Network (CNN)**: We used a CNN-based architecture, such as *EfficientNet* or *ResNet*, for image feature extraction. These models were pre-trained on ImageNet for robust feature extraction and fine-tuned on our specific dataset.
   - **Multimodal Learning**: The extracted image features and OCR text were combined using a fusion layer to predict the entity values. This was key to allowing the model to correlate visual information with extracted textual values.

4. **UNIT CONVERSION AND POST-PROCESSING**
   - Using the *`constants.py`* file, we mapped recognized units to standard units (e.g., converting "grams" to "g" or "centimeters" to "cm") to ensure that our output conformed to the required format.
   - Post-processing rules were applied to handle cases where no value was found or when the image contained multiple values (e.g., average over multiple predictions).

EXPERIMENTS

1. **EXPERIMENTATION WITH OCR**
   - Incorporating OCR improved our F1 score significantly.
   - By extracting numeric text and units from the images, the model was better able to predict values like "weight" and "dimension".

2. **HYPERPARAMETER TUNING**
   - **Learning Rate**: The optimal learning rate was 0.001, identified through grid search.
   - **Batch Size**: A batch size of 32 was chosen based on memory constraints and model performance.
   - **Training Epoch**: We trained the model for 20 epochs with early stopping based on validation loss.

3. **EVALUATION METRICS**
   - We used the F1 score for evaluating the performance, calculated based on true positives, false positives, false negatives, and true negatives.

CONCLUSION

Our solution successfully extracts product entity values from images by combining CNN-based image feature extraction with OCR-based text extraction. Post-processing ensures that the output format adheres to the required format, including standard units. Future work could focus on improving accuracy through better OCR models or using attention mechanisms to focus on relevant parts of the image.

GROUP MEMBERS: 1. *Vedang Sakpal* 3. *Anusha Khare*
2. *Akash Jangid* 4. *Manasvi Swarnkar*