Amazon ML Challenge

Team Name : Package Delivery

Team Members : Abhijit Raju Nair, Bhavesh Goyal, Shiven Patel, Yashasvi Rai

Abstract

In this challenge, we aim to develop a ML model capable of extracting key entity values directly from product images, a task critical to domains like e-commerce, healthcare, and content moderation. With the growing reliance on digital platforms, many products lack sufficient textual descriptions, making it necessary to derive information such as weight, dimensions, and volume directly from visual data. The dataset includes images along with associated product details, such as entity names and values. Our approach involves training a model to predict these entity values in a standardized format from images, with the goal of achieving a high F1 score. This solution holds the potential to enhance the accuracy and completeness of product data in large-scale digital marketplaces.

1. Problem Statement

P.S : Develop a machine learning model that extracts entity values, such as weight, volume, and dimensions, from product images.

Explanation: The challenge focuses on creating a model to extract specific data points from product images, which is essential for improving product details in digital marketplaces. The provided dataset includes images and related product details like entity names, and the goal is to predict values for unseen data. Predictions must follow a specified format and be evaluated based on their precision and recall, ultimately measured by the F1 score.

2. Our Approach

The problem at hand is an Optical Character Recognition (OCR) task aimed at identifying and extracting specific product features from images. This task involves extracting entity values such as weight, volume, and dimensions, which are critical for accurate product descriptions in digital platforms. We approached this problem by utilizing a state-of-the-art vision-language model (VLM), InternVL (OpenGVLab/InternVL2-1B), known for its advanced semantic understanding of both visual and textual elements in images. This model enables us to derive precise quantitative information from the images and match it to product attributes effectively.

2.1. Preprocessing and Data Augmentation

To ensure that the model learns robust features, we performed preprocessing steps such as image resizing, normalization, and augmentation techniques like random cropping, flipping, and color jittering. This helped in generating diverse training samples, allowing the model to generalize better on unseen data. For text-based feature extraction, OCR was applied to the image regions containing numeric values and units.

2.2. Vision Transformer for Semantic Understanding

We employed InternVL, a Vision Transformer-based large language model, for image understanding and entity extraction. Unlike traditional CNN-based models, Vision Transformers (ViTs) offer a global view of the image by processing patches and applying self-attention mechanisms across the entire image, capturing long-range dependencies and contextual relationships between product features. InternVL's multi-modal capabilities allowed it to combine visual content with semantic understanding, crucial for interpreting quantitative values like weight, dimensions, and units from product labels.

2.3. Feature Extraction Using InternVL

InternVL was fine-tuned on the dataset to extract features that contain semantic meaning for entity prediction. Given its ability to process both images and textual information, the model was able to localize and interpret specific numeric values and their associated units. The model learned to distinguish between various entity types (e.g., weight, volume, length) based on the context in which these values appear on product labels.

2.4. Handling Multi-Unit and Noisy Data

One challenge was the presence of multiple units (e.g., grams, kilograms, ounces) and noisy data, such as blurry or low-resolution images. We addressed this by integrating the model with custom regularization techniques and leveraging the unit-conversion rules from constants.py to ensure output predictions matched the required format. Additionally, the model's OCR output was refined using post-processing techniques to handle discrepancies in recognized characters and ensure correct unit assignment.

2.5. Prediction and Post-Processing

The final predictions were formatted according to the specified guidelines, with each prediction containing a numerical value followed by the appropriate unit. For images where no entity value could be detected, the model returned an empty string. A post-processing sanity check was implemented to verify the formatting and ensure predictions complied with the constraints laid out in the sanity.py script.

2.6. Evaluation and F1 Score Optimization

To evaluate model performance, we used the F1 score, which balances precision and recall. Precision ensured that the extracted entity values were accurate, while recall ensured that no important values were missed. We tuned the model hyperparameters (e.g., learning rate, patch size, and attention heads) to optimize the F1 score, focusing on reducing false positives (incorrect predictions) and false negatives (missed predictions).

3. Dataset

The dataset used in this task contains several important columns, each contributing to the machine learning model's ability to extract meaningful information from images. The columns are outlined as follows:

index: A unique identifier (ID) for the data sample.

image_link: Public URL where the product image is available for download.

group_id: Category code of the product. entity_name: Product entity name. For example, "item_weight".

entity_value: Product entity value. For example, "34 gram".

4. Dataset Enhancement

To address the challenges posed by the initial dataset, we undertook several data improvement techniques aimed at enhancing image quality and relevance. The original dataset contained images of varying

quality, which adversely affected the accuracy and performance of the model. The following steps were implemented to rectify these issues:

4.1. Image Upscaling

The images in the dataset were often of low resolution, leading to blurred and unclear visual details. To mitigate this, we applied image upscaling techniques using advanced algorithms. Upscaling improves the image resolution, making it easier for the model to discern fine details and extract accurate features. Image Augmentation was crucial for better feature extraction, as higher resolution images provide more detailed information, leading to improved model performance.

4.2. Image Cropping

In many instances, the critical entity values were located near the center of the images, while extraneous background information surrounding these values was not necessary for model training. To optimize the dataset, we implemented image cropping techniques to focus on regions of interest (ROIs). By cropping the images to include only the relevant parts, we reduced the amount of irrelevant data the model had to process. This approach not only improved computational efficiency but also enhanced the model's ability to accurately recognize and extract the entity values from the central regions of the images.

4.3. Justification and Impact

The application of these techniques was necessary to ensure that the model could learn from high-quality and relevant image data. Low-quality images with poor resolution can lead to suboptimal feature extraction and inaccurate predictions. By upscaling, we ensured that the images were clear and detailed, enabling the model to better identify and understand product attributes. Cropping reduced noise and focused the model's attention on the most pertinent information, improving both training efficiency and prediction accuracy.

5. Resources Required

Essential Resources to efficiently handle the data processing, model training, and storage requirements for the machine learning tasks involved are: 16 GB RAM for GPU and 20 GB of Storage.

6. InternVL Model

InternVL : OpenGVLab/InternVL2-1B was selected for our task due to its advanced capabilities in understanding and extracting detailed information from images.

6.1. Why InternVL2?

Figure 1 shows the comparison of InternVL2 with other models.

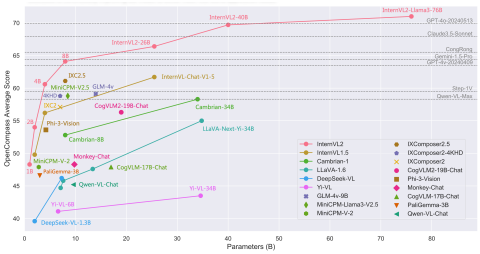

Figure 1

Advanced Vision Transformer: InternVL is a state-of-the-art vision transformer model that excels in handling complex image data and extracting semantic features with high accuracy.

Enhanced Feature Extraction: It provides improved performance in recognizing fine details and understanding the context of entities within images, which is crucial for precise feature extraction.

Robust Performance: InternVL's architecture is designed to manage various image distortions and variations effectively, making it suitable for our dataset's quality and resolution challenges.

6.2. Setting up InternVL

Start by installing the necessary libraries, including PyTorch for deep learning operations and Hugging Face's transformers library for accessing pre-trained models. This can be done using standard package managers like pip.

Once the environment is set up, the pre-trained InternVL2-1B model and its corresponding tokenizer can be loaded using the from_pretrained method provided by Hugging Face. The tokenizer is responsible for transforming images into a format that the model can process, while the model itself performs the complex task of multimodal inference, interpreting both visual and text inputs. Before inference, images need to be preprocessed to match the model's input requirements. This includes tasks like resizing and normalizing pixel values, ensuring compatibility with the vision transformer architecture of InternVL. After preprocessing, the image is tokenized and fed into the model, allowing it to semantically analyze the visual features and textual prompts.

```python
# Install necessary libraries
!pip install torch transformers datasets

# Import required modules
from transformers import InternVLModel, InternVLTokenizer
from PIL import Image
import torch

# Load pre-trained model and tokenizer
model = InternVLModel.from_pretrained('OpenGVLab/InternVL2-1B')
tokenizer = InternVLTokenizer.from_pretrained('OpenGVLab/InternVL2-1B')

# Load and preprocess the image
image = Image.open('path_to_image.jpg')
inputs = tokenizer(image, return_tensors="pt")

# Perform inference
outputs = model(**inputs)
```

Figure 2

The combination of these steps enables efficient extraction of relevant product information, such as dimensions, weights, and other key features, directly from images. This setup forms the foundation for accurate and automated data extraction in e-commerce and other image-intensive applications.

7. Prompt Testing

During the model experimentation phase, various prompts were tested to optimize the accuracy of the predictions. The objective was to refine the language and structure of the prompts so that the model could accurately identify and extract the relevant feature (e.g., weight, dimension) from the product images. We experimented with a range of prompt styles, each aiming to guide the model in recognizing and reporting the required entity values while maintaining strict adherence to the given format.

```python
question = f'<image>\ntell in 1 line what is the ' \
           f'{use} of product shown in the correct units given'
```

Figure 3

Through systematic prompt engineering, we found that shorter and more direct prompts yielded better performance, reducing ambiguity

and noise in the output. By iterating through many prompts and evaluating the outputs based on precision and relevance, we identified the most effective version for accurate entity extraction. The final prompt ensured that the model provided clear and concise responses, directly improving model efficiency and output quality.

8. Extraction of Quantity and Unit Mapping using Map.pkl

We implemented a technique to extract numeric quantities and their corresponding units from the product descriptions. The process involves utilizing a pre-loaded mapping dictionary (Map.pkl) that associates different entities (such as weight, volume, dimensions, etc.) with a predefined set of valid units.

Approach :

We started by loading the mapping data from Map.pkl using the pickle module. This dictionary maps entity types (e.g., weight, volume) to a list of valid units (e.g., gram, litre).

To extract quantities and units from the text, we split the text into tokens and identify numeric values by parsing consecutive characters. After extracting the quantity, we search the adjacent tokens for the unit. These tokens are matched against the predefined units in the mapping dictionary. If a match is found, the quantity and unit are returned in the correct format (e.g., 12.5 kg).

This method ensures accurate extraction of numerical information and valid units, even when multiple unit candidates are present. It is effective for handling various unit systems, including both metric and imperial values, by leveraging the predefined unit set.

9. Pipeline for Systematic Query Generation and Answer Extraction

The pipeline for systematic query generation and answer extraction involves several critical steps to ensure accurate and efficient data processing from images. The process begins by reading and preparing the dataset, followed by querying the model with specific prompts based on entity types.

Data Preparation: The code starts by reading a CSV file containing image paths and entity types. It checks for existing predictions and initializes them if missing.

Image Handling: For each image, the relevant file is loaded, and its path is constructed. If the image exists, it is preprocessed to match the model's input requirements.

Query Generation: The entity type is mapped to a descriptive term (e.g., "height" becomes "vertical height"). A query is then formulated to extract the specific entity value from the image using a predefined prompt format.

Model Inference: The query is sent to the InternVL model, and the response is received. The response undergoes post-processing to handle unit conversions and remove unnecessary details. This involves regular expression patterns to clean and format the extracted information accurately.

Post-processing: The response text is processed to identify and correct unit names and handle cases of overlapping or redundant text. The resulting data is then saved or utilized as needed.

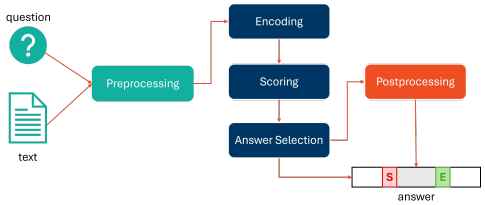

Figure 4

The Query Generation and Answer Extraction Pipeline enhances model performance by improving accuracy through contextually relevant queries. It optimizes resource utilization, ensuring efficient use of computational resources. The pipeline is adaptable to various entity types and robust against errors, providing reliable and consistent outputs. Its scalable design supports high-throughput processing, making it effective for large-scale applications and significantly boosting data extraction efficiency.

10. Sanity Checks

We conducted several sanity checks to ensure the accuracy and reliability of our data processing:

Data Validation: We verified that all required columns, such as image links and entity names, were present and correctly formatted in the dataset.

Image Verification: We ensured that all image files were correctly located and accessible to prevent errors related to missing or corrupt images.

Response Quality: We assessed the model's outputs for reasonableness and correct formatting, making sure that responses included appropriate units and quantities.

Error Handling: We implemented measures to detect and address anomalies, such as missing units or incorrect formats, to maintain data integrity.

Consistency Checks: We confirmed that extracted values and units were consistent with predefined mappings, ensuring that all outputs adhered to expected standards.

These checks were essential for maintaining the robustness and accuracy of our system, ensuring reliable and precise data extraction.

11. Conclusion

In this project, we successfully engineered and deployed a sophisticated pipeline designed to extract quantitative information and units from product images using advanced machine learning techniques. By harnessing the capabilities of InternVL, an innovative vision-language model, we achieved efficient and accurate data extraction critical for applications such as automated product descriptions and inventory management.

Our approach encompassed several key phases: meticulous image preprocessing, precise prompt engineering, and systematic query generation. Through the application of image augmentation techniques—including upscaling and cropping—we enhanced the quality and relevance of the input data, leading to substantial improvements in the model's accuracy. The query generation process was carefully crafted to ensure the accurate extraction of quantities and units from textual information, facilitated by a robust mapping mechanism utilizing the Map.pkl file.

The effectiveness of the pipeline was validated through comprehensive sanity checks, confirming the reliability and consistency of the extracted data. By addressing potential edge cases and validating outputs rigorously, we ensured the robustness of the system. The pipeline's systematic design guarantees both reproducibility and scalability, positioning it as a viable solution for real-world deployment.

Overall, this project demonstrates the potential of integrating cutting-edge machine learning technologies with systematic data extraction methodologies to enhance automation and precision in various domains. The successful implementation of this pipeline lays a solid foundation for future advancements and practical applications in the field.