ML Hackathon Report

By:
Team: ML Mavericks (Charotar University of Science and Technology)

1. Neel Shah (mail: neeldevenshah@gmail.com)
2. Sneh Shah (mail: 22aiml049@charuat.edu.in)
3. Harsh Maheshwari (mail: 22aiml019@charusat.edu.in)

1. Introduction: Llama and Vision-Language Models

1.1 Llama Architecture

Llama (Large Language Model Meta AI) is a family of large language models developed by Meta AI. Key features of the Llama architecture include:

1. Efficient scaling: Designed to be computationally efficient while maintaining high performance
2. Open-source nature: Allows for community contributions and adaptations
3. Transformer-based: Utilizes the transformer architecture, which excels at processing sequential data

The Llama architecture has been pivotal in advancing the field of natural language processing and serves as the foundation for many derivative models, including the one used in our project.

1.2 Vision-Language Models

Vision-Language Models (VLMs) represent a significant advancement in AI, combining computer vision and natural language processing capabilities. These models can:

1. Process both images and text simultaneously
2. Understand the relationship between visual elements and textual descriptions
3. Generate text based on visual inputs and vice versa

Notable examples of VLMs include CLIP (Contrastive Language-Image Pre-training), DALL-E, and various multimodal transformers.

2. Model Overview: MiniCPM-Llama3-V-2.5

2.1 Architecture

MiniCPM-Llama3-V-2.5 is a state-of-the-art vision-language model that builds upon the Llama architecture, integrating advanced visual processing capabilities. Key architectural features include:

1. Multi-modal encoder-decoder structure
2. Cross-attention mechanisms for image-text alignment
3. Specialized visual encoding layers
4. Llama-based language modeling components

2.2 Key Features

- Parameter Count: 8 billion parameters, balancing model capacity and computational efficiency
- Multi-modal Learning: Seamlessly integrates textual and visual information processing
- Multilingual Support: Capable of processing over 30 languages, including English, German, French, Spanish, and Chinese
- Enhanced OCR: Improved optical character recognition capabilities for text extraction from images
- Instruction Following: Advanced ability to follow complex, multi-step instructions
- Efficient Deployment: Optimized for deployment on end-side devices through quantization and compilation techniques

3. Methodology

3.1 Data Analysis and Preparation

Our exploratory data analysis revealed:

- A diverse set of product images
- Text in both German and English languages
- Various product attributes (e.g., weight, dimensions, capacity)

3.2 Model Implementation

1. Image normalization and resizing
2. Text extraction and cleaning from image metadata (if available)
3. Attribute categorization and labeling

3.3 Model Implementation

1. Base Model Selection: We chose the pre-trained MiniCPM-Llama3-V-2.5 model due to its multilingual capabilities and superior performance on similar tasks.
2. Integration: Implemented the model within our processing pipeline, ensuring efficient data flow and resource utilization.
3. Query Formation: Developed a standardized query template to extract specific attributes:
   ```
   f"Give me the {row['entity_name']} of the product with units,
   given to point answer and in short within 5 words only"
   ```

Where `entity_name` corresponds to the specific attribute we're extracting from the image.

4. Batch Processing: Implemented batching techniques to optimize throughput and resource utilization.

3.4 Post-processing Pipeline

1. Text Extraction:
   - Utilized regex patterns and Python's natural language processing libraries to extract numerical values and units from the model's output.
   - Implemented custom extractors for complex attributes (e.g., dimensions with multiple values).
2. Unit Standardization:
   - Developed a comprehensive rule-based system to convert non-standard units to standard ones.
   - Example conversions:
     - "g" or "gm" to "gram", "cm" to "centimeter", "L" to "liter"
3. Edge Case Handling:
   - Created specific rules for special cases, such as:
     - Replacing quotation marks (") with "inch" for measurements
     - Handling fractional values in dimensions or weights
4. Precision Adjustment:
   - Conducted in-depth exploratory data analysis (EDA) to identify attribute classes requiring higher precision.
   - Implemented conditional formatting to adjust decimal places based on attribute type and value range.

4. Results and Analysis

4.1 Overall Performance

1. Processing Speed: Average processing time per image ranged from 2 to 3.5 seconds.
2. Scalability: Our approach of dividing the dataset and using multiple GPUs allowed us to process the entire dataset efficiently despite resource constraints.

4.2 Error Analysis

1. Common Error Types:
   - Misinterpretation of units in complex measurements
   - Occasional confusion between similar attributes (e.g., width vs. length)
   - Rare hallucinations in highly stylized or unconventional product images
2. Edge Cases:
   - Products with multiple conflicting measurements listed
   - Handwritten text in images posed challenges in some instances

4.3 Comparative Analysis

When compared to other models we tested:

1. Donut Model Variants:
   - Vision question-answering version: 20% (approx.) lower accuracy
   - Text extraction version: 25% (approx.) lower accuracy
2. Generic OCR + NLP Pipeline: 50% lower accuracy and 60% (approx.) lower accuracy
3. Proprietary Vision-Language Models: While we couldn't directly test these, based on published benchmarks, MiniCPM-Llama3-V-2.5 showed comparable or superior performance in similar tasks.

4.4 Key Strengths of Our Approach

1. Multilingual Robustness: The model's ability to handle both German and English text without significant performance degradation is a major advantage for diverse e-commerce platforms.
2. Scalable Processing: Our strategy of dividing the dataset and utilizing multiple GPUs demonstrates the scalability of our approach, allowing for efficient processing of large datasets even with limited resources.
3. Adaptability: The post-processing pipeline's rule-based components allow for easy adaptation to new attribute types or units.

4.5 Limitations and Challenges

1. Resource Intensity: The model's size and processing time (2 to 3.5 seconds per image) may pose challenges for real-time applications or deployment in resource-constrained environments.
2. GPU Availability: Our reliance on free GPU resources from Kaggle, while cost-effective, may not be sustainable for large-scale or ongoing production use.

5. Conclusion

Our implementation of the pre-trained MiniCPM-Llama3-V-2.5 model, coupled with an advanced post-processing pipeline, has demonstrated state-of-the-art performance in product attribute extraction from images. The system's multilingual capabilities, high accuracy across various attribute types, and efficient processing make it a powerful tool for automating product information extraction in global e-commerce and inventory management applications.