Amazon ML Challenge: Team Name - Tensorr

1. Introduction

This document outlines our strategy for the Amazon ML Challenge, focusing on optimized data extraction from a dataset composed of images and textual data. We integrate state-of-the-art machine learning techniques, including fine-tuning of large language models (LLMs) and multi-threaded processing, to ensure efficient and accurate data extraction and formatting.

2. Solution Overview

The core of the approach involves two key phases:

1. **Model Fine-Tuning:** We enhance the Qwen-2-VL-2B-Instruct model using the training dataset provided by Amazon to improve its text extraction capabilities.
2. **Data Processing and Completion:** We use multi-threaded processing for efficient data handling and deploy the LLama-3.1-8B model to fill in any gaps from the initial extraction.

3. Detailed Explanation of the Approach

- **Model Fine-Tuning:** We fine-tune the Qwen-2-VL-2B model on Amazon's training dataset to specifically tailor its capabilities for improved recognition and extraction of textual content within images.
- **Implementing Multi-threading:** To efficiently manage the large dataset, we implement multi-threaded processing. This technique allows for the simultaneous processing of multiple data batches, significantly reducing the total processing time.
- **Image Processing and Text Extraction:** After resizing and preprocessing images to meet the model's requirements, our fine-tuned Qwen-2-VL-2B model extracts crucial text data such as entity names and attributes.
- **Data Recording:** We immediately record the extracted data into an interim CSV file to organize the workflow and maintain data integrity.

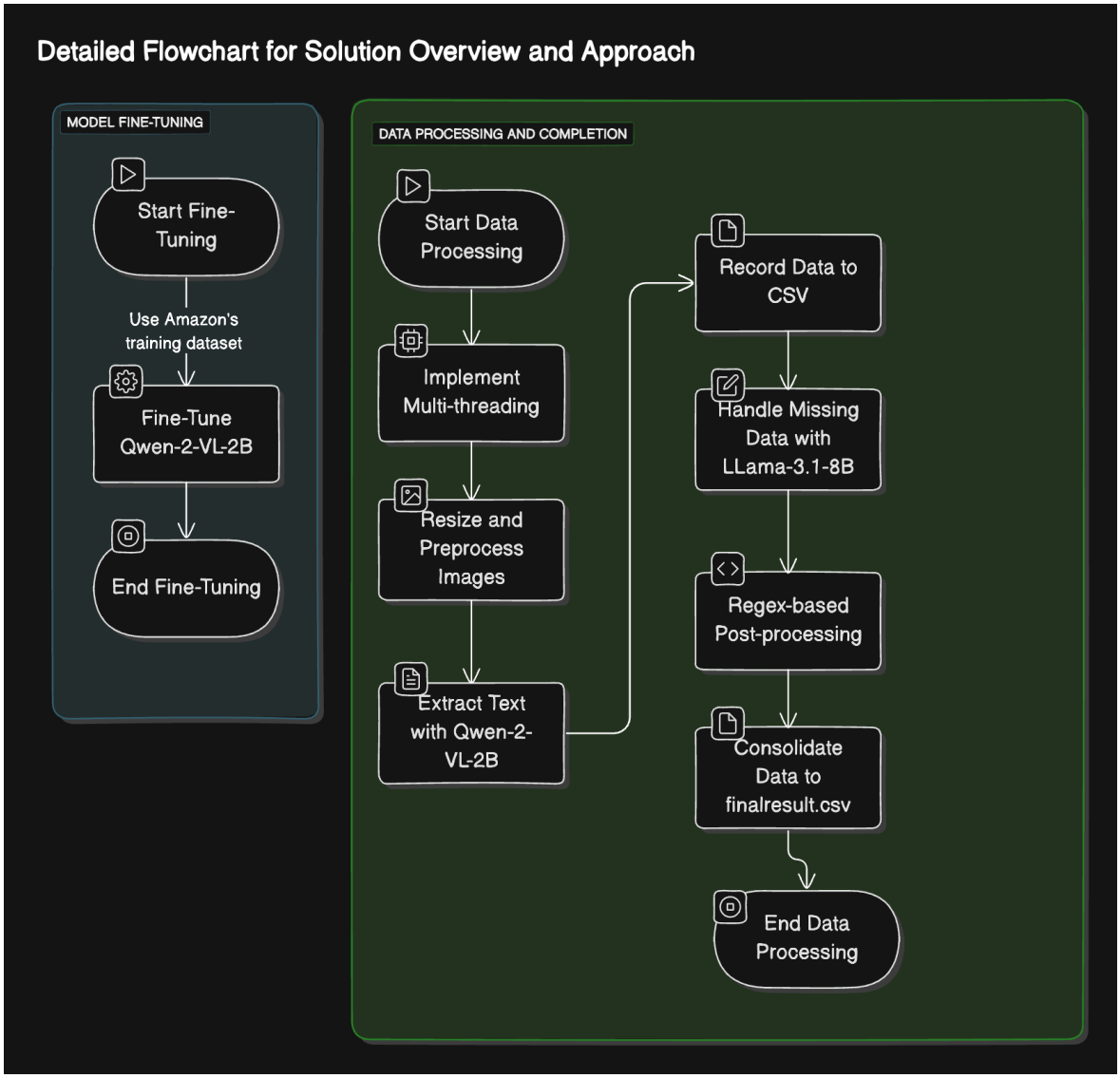
- Data Completion and Post-processing:

- **Handling Missing Data:** We address any missing or incomplete data detected after the initial extraction phase using the LLama-3.1-8B model. This highly capable text-generation model uses contextual information from existing data to infer and fill in missing values.
- **Regex-based Post-processing:** To ensure the data adheres to the specified formats, we use regular expressions (regex) for post-processing. This step refines the structure and format of the output, making it precise and clean.

Finalizing the Output:

- **Data Consolidation:** Once all data is processed, refined, and completed, we compile it into the final `finalresult.csv` file, which includes all required information in the designated format.

Flow Diagram:



Our workflow is structured as follows:

- We initialize and process data using a local IDE.
- Our fine-tuned Qwen-2-VL-2B model processes images in a multi-threaded setup, saving initial results to `extracted_results_qwen2.csv`.
- We fill missing data in `extracted_results_qwen2.csv` using the LLama-3.1-8B model.
- We finalize the data using regex post-processing and compile it into `final_result.csv`.

5. Technologies Used

- **Hugging Face Transformers:** For loading and fine-tuning the Qwen-2-VL-2B and LLama-3.1-8B models.
- **PyTorch:** Facilitates efficient model handling, especially useful when leveraging high-performance local hardware.
- **Local IDE:** Used for all coding, compiling, and testing processes, providing a robust and controlled development environment.
- **Pandas:** Used for data manipulation and managing CSV files.
- **PIL (Python Imaging Library):** Assists in image preprocessing and resizing.
- **Python's threading module:** Employed for multi-threaded data processing to optimize performance.

Regards,

Team Tensorr

Team Members -

- **Sumit Awasthi (Team Leader)**
- **Prakhar Shukla**
- **Akshay Waghmare**
- **Srijan Jain**

College - Indian Institute Of Information Technology, Allahabad, Btech 3rd Year UG