Entity Value Extraction from Images

Deepminders

Project Overview

This project, developed for the Amazon ML Challenge, focuses on extracting entity values from images. The solution employs a two-step approach combining a decision tree classification model and Optical Character Recognition (OCR) to identify both the entity unit and its corresponding value.

System Architecture

1. **Decision Tree Classification Model**
   - **Purpose**: Predict the entity unit based on `group_id` and `entity_name`.
   - **Input**: `group_id` & One-hot encoded `entity_name`
   - **Output**: Predicted entity unit (labeled from 0 to 31)
2. **OCR Model**
   - **Purpose**: Extract the numerical value associated with the entity from the image.
   - **Input**: Image file downloaded from `image_link`
   - **Output**: Extracted text containing numerical values
3. **Integration using String Manipulation**
   - **Purpose**: Combine the outputs from the classification and OCR models.
   - **Input**:
     - Predicted unit from the classification model
     - Extracted numerical value from the OCR model
   - **Output**: Concatenated string of entity value and unit

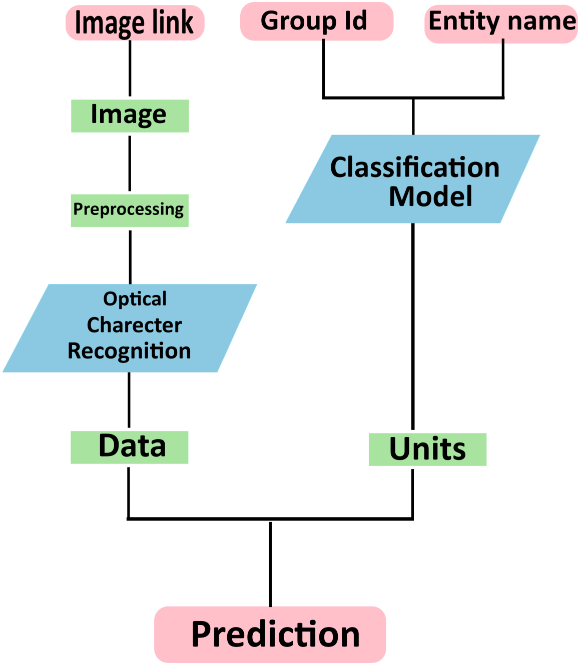

Figure 1: Flowchart of the Entity Value Extraction Process

Workflow

1. **Data Preprocessing**:
   - Apply one-hot encoding to the `entity_name` column.
   - Split `entity_value` into separate value and unit columns.
   - Label distinct units from 0 to 31.
2. **Classification Model Training**:
   - **Input**: `group_id` and one-hot encoded `entity_name`
   - **Output**: Labeled unit (0-31)
   - Train a decision tree classification model using this data.
3. **Image Processing and OCR**:
   - Download images from the provided `image_link`.
   - Apply OCR to extract text from the images.
   - Parse the extracted text to identify numerical values.
4. **Prediction and Integration**:
   - Use the trained classification model to predict the unit for new inputs.
   - Extract numerical values from the image using OCR.
   - Combine the predicted unit and extracted value.
5. **Output Generation**:
   - Return the concatenated string of value and unit as the final output.

Implementation Details

1. **Decision Tree Classification Model**
   - **Features**:
     - `group_id` (numerical)
     - `entity_name` (one-hot encoded)
   - **Target**: Unit (labeled 0-31)
   - **Model**: Decision Tree Classifier
2. **OCR Process**
   - **Image Acquisition**: Download images from the `image_link` provided in the dataset.
   - **OCR Technology**: Tesseract, Google Vision API
   - **Text Extraction**: Extract all text from the image.
   - **Value Parsing**: Implement logic to identify and extract numerical values from the OCR output.
3. **Data Preprocessing**
   - One-hot encoding of `entity_name` column
   - Splitting `entity_value` into value and unit
   - Labeling distinct units from 0 to 31

Challenges and Solutions

1. **Data Preprocessing Challenges** Implemented one-hot encoding for entity names and standardized labeling (0-31) for units.
2. **Image Quality Variability** Applied resizing and contrast adjustment to improve OCR accuracy.
3. **Model Generalization** Used cross-validation and pruned the decision tree to avoid overfitting.
4. **Performance Optimization** Implemented batch processing to handle large volumes of images efficiently.

Conclusion

This solution leverages both machine learning classification and OCR technologies to extract entity values from images. The modular approach allows for individual component optimization and provides a scalable solution for the given problem statement.