Methodology: Vision Feature Extraction from Images

Team Licht den Code
Siddhant, Arjun, Hetansh, Vedica

Abstract

This document presents a comprehensive methodology for extracting specific entity values from images using advanced machine learning techniques. Developed by Team Licht den Code, we detail the mathematical foundations, image processing algorithms, and vision-language model architecture used in our approach, with a focus on the Qwen2-VL model and associated utilities.

1 Introduction

The expansion of digital marketplaces necessitates accurate and detailed product information extraction directly from images. This methodology, developed by Team Licht den Code, outlines the creation of an AI-powered system capable of identifying and extracting specific entity values such as weight, volume, dimensions, and other critical product information from images.

2 Problem Formulation

Given an input image $I \in \mathbb{R}^{H \times W \times 3}$ and a set of target entities $E = \{e_1, e_2, ..., e_n\}$, our goal is to find a function $f : I \times E \to V$, where $V = \{v_1, v_2, ..., v_n\}$ represents the corresponding entity values. Each v_i consists of a numerical value and an associated unit (where applicable).

3 Methodology

3.1 Image Preprocessing

3.1.1 Smart Resizing

We employ a smart resizing algorithm to ensure optimal image dimensions while preserving aspect ratio:

$$(h', w') = \arg\min_{(h,w)} |hw - \alpha HW| \quad \text{subject to} \quad \frac{h}{H} = \frac{w}{W}, \quad h, w \in k\mathbb{Z}^+ \tag{1}$$

where H and W are original height and width, h' and w' are new dimensions, k is the dimension factor (typically 28), and α is a scaling factor to ensure the total number of pixels is within a specified range.

3.1.2 Aspect Ratio Constraint

To prevent extreme aspect ratios, we enforce:

$$\max\left(\frac{H}{W}, \frac{W}{H}\right) \leq R_{max} \tag{2}$$

where R_{max} is the maximum allowed aspect ratio (typically 200).

3.2 Vision-Language Model Architecture

We utilize the Qwen2-VL-7B-Instruct model, a large-scale vision-language model based on the transformer architecture. The model processes both image and text inputs to generate relevant textual outputs.

3.2.1 Image Encoding

The image I is encoded into a sequence of image tokens $T_I = \{t_1, t_2, ..., t_m\}$ using a vision transformer:

$$T_I = \text{ViT}(I) \tag{3}$$

3.2.2 Text Encoding

The prompt P for each entity e_i is tokenized into a sequence of text tokens $T_P = \{p_1, p_2, ..., p_l\}$:

$$T_P = \text{Tokenize}(P(e_i)) \tag{4}$$

3.2.3 Cross-Modal Attention

The model uses cross-modal attention to fuse image and text information:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where Q, K, and V are query, key, and value matrices derived from the combined image and text token sequences.

3.3 Inference Pipeline

3.3.1 Prompt Generation

For each entity e_i, we generate a prompt:

$$P(e_i) = \text{"Extract the value of } e_i \text{ from the image."} \tag{6}$$

3.3.2 Entity Value Extraction

The model generates a sequence of output tokens $O = \{o_1, o_2, ..., o_k\}$:

$$O = \arg\max_O P(O|T_I, T_P) \tag{7}$$

3.3.3 Post-processing

We apply regex patterns to extract numerical values and units from the generated text:

$$v_i = \text{Regex}(O, \text{pattern}_{e_i}) \tag{8}$$

4 Future Scope: Video Processing

For future extensions to video inputs, we propose the following methodology:

4.1 Frame Extraction

We sample frames at a specified FPS or total number of frames:

$$F = \{f_t | t = \text{round}(i \cdot \frac{T}{N}), i = 0, 1, ..., N-1\} \tag{9}$$

where F is the set of extracted frames, T is the total number of frames in the video, and N is the desired number of frames.

4.2 Frame Resizing

We apply smart resizing to each frame, ensuring consistent dimensions across the video:

$$(h'_v, w'_v) = \arg\min_{(h,w)} |hwN - \beta HWT| \quad \text{subject to} \quad \frac{h}{H} = \frac{w}{W}, \quad h, w \in k\mathbb{Z}^+ \tag{10}$$

where β is a scaling factor for video, and T is the number of frames.

5 Evaluation Metrics

We use the following metrics to evaluate our model:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

where TP, FP, and FN are true positives, false positives, and false negatives, respectively.