Automated Trademark Image-to-Text Description System (Multimodal AI)

This document presents a complete end-to-end API pipeline for automatic trademark image analysis and description generation. The system fuses optical character recognition (OCR) and multimodal vision-language modeling (VLM) using EasyOCR and BLIP (Bootstrapped Language Image Pretraining) respectively. Designed for trademark indexing and retrieval applications, it extracts textual content of marks, Chinese characters (both Simplified and Traditional), and semantic device or shape descriptions from logo images. The BLIP model used here is a finetuned variant, trained on a curated subset of the provided dataset to enhance its understanding of trademark domain characteristics. The solution is deployed as a FastAPI microservice containerized in Docker, capable of running in both CPU and CUDA environments with full fault-tolerant inference, performance monitoring, and JSON-based output logging.

Hardware and Environment Constraints

All development, experimentation, and testing were performed on a MacBook Pro (Retina, 13-inch, Early 2015) running macOS Monterey 12.7.6. The system configuration included an Intel Core i5 dual-core CPU (2.7 GHz) and 8 GB 1867 MHz DDR3 RAM. Given these specifications, hardware constraints significantly limited the ability to perform large-scale model training or fine-tuning locally. As a result, the experiments focused on lightweight inference pipelines and fine-tuning with small dataset (randomly selected 2K data) rather than substantial data model training. Larger model training and inference scalability were planned for deployment on GPU-backed servers or cloud environments (e.g., AWS, GCP, or local GPU clusters). Despite these limitations, efficient optimization techniques and modular pipeline design ensured that core functionality, including OCR and hybrid inference, was validated locally.

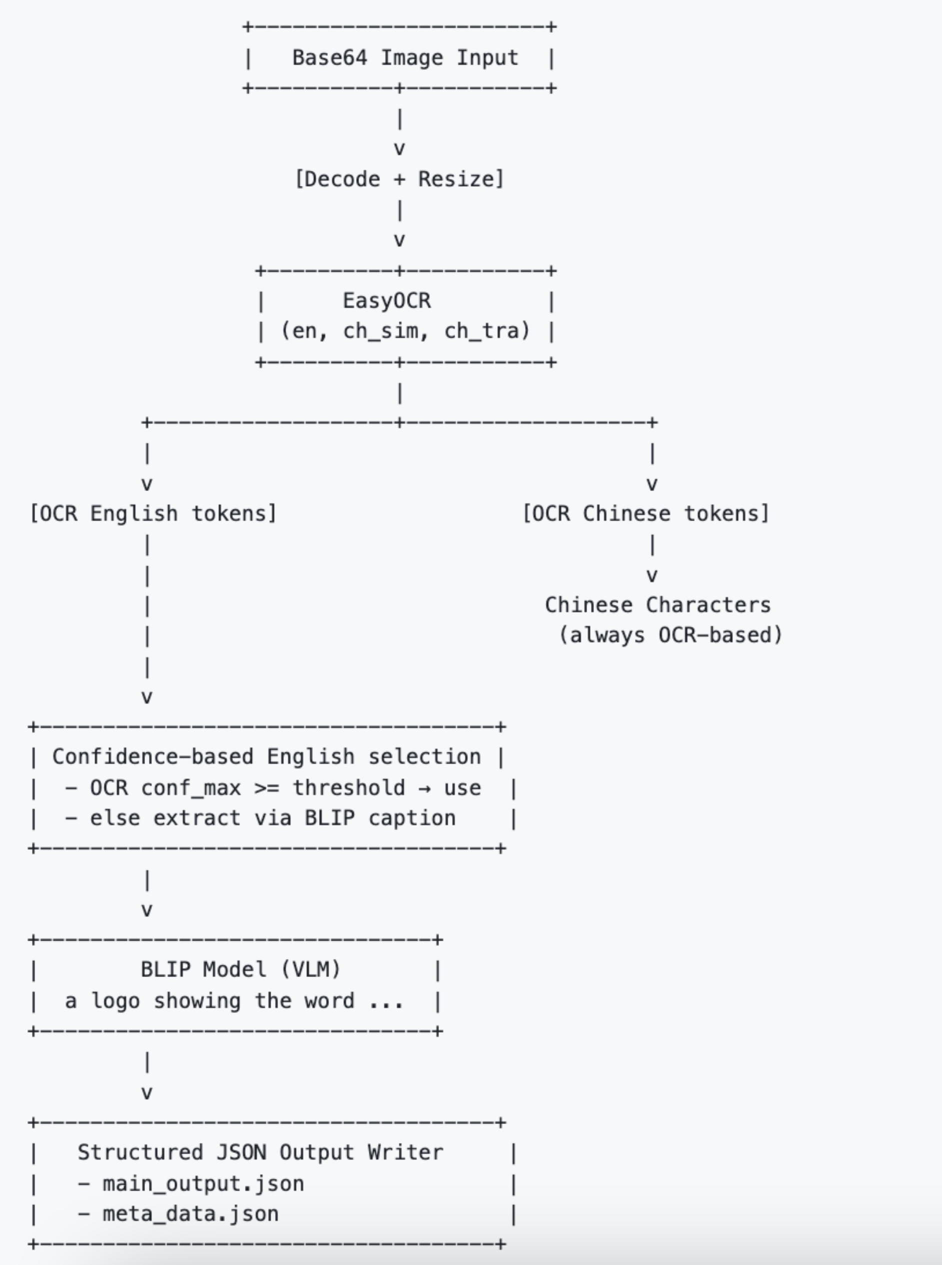
System Overview

The system ingests a trademark image (base64-encoded) and produces three key outputs: 1) wordsInMark – the normalized textual content of the English portion of the mark, extracted from OCR if confident, otherwise from BLIP captions; 2) chineseCharacter – direct Chinese characters extracted from OCR (both simplified and traditional forms); and 3) descrOfDevice – a generative description of the image's visual device or shape, produced using the BLIP vision-language model.

Dataset Description

The dataset comprised approximately 360,000 trademark samples exhibiting diverse modalities and annotation completeness. Each sample primarily consisted of a trademark image, accompanied by optional textual and descriptive metadata. The dataset distribution included multiple combinations of available modalities: Image + English text + Chinese text + Device description, Image + English text only, Image +

Chinese text only, Image + Device description only, etc. This diversity reflects real-world variability in trademark submissions and provided a robust basis for developing a multimodal captioning and OCR-based inference system.



```
                 +-----------+-----------+
                 |   Base64 Image Input  |
                 +-----------+-----------+
                             |
                             v
                     [Decode + Resize]
                             |
                             v
                  +----------+-----------+
                  |      EasyOCR         |
                  | (en, ch_sim, ch_tra) |
                  +----------+-----------+
                             |
         +-------------------+-------------------+
         |                                        |
         v                                        v
[OCR English tokens]                    [OCR Chinese tokens]
         |                                        |
         |                                        v
         |                                Chinese Characters
         |                                 (always OCR-based)
         |
         v
+-------------------------------------+
| Confidence-based English selection |
|  - OCR conf_max >= threshold → use  |
|  - else extract via BLIP caption    |
+-------------------------------------+
         |
         v
+--------------------------------+
|        BLIP Model (VLM)       |
|  a logo showing the word ...  |
+--------------------------------+
         |
         v
+-------------------------------------+
|   Structured JSON Output Writer     |
|   - main_output.json                |
|   - meta_data.json                  |
+-------------------------------------+
```

Figure 1: Overall solution architecture

Solution Architecture and Components

The core of the system integrates multiple components (Fig. 1) that collaboratively process visual and textual signals from trademarks:

1. **OCR Engine (EasyOCR)** – Each image is decoded from base64 to RGB using PIL, resized to a configurable maximum dimension, and optionally preprocessed for enhanced OCR under low-confidence conditions. Extracts embedded English and Chinese characters from trademark images. Configured with en, ch_sim and ch_tra language settings, it operates with a confidence threshold to ensure text accuracy (tested with 0.4).

2. **BLIP Captioning Model (Finetuned Variant)** – A BLIP model (Bootstrapped Language-Image Pretraining) was employed for generating visual descriptions of trademarks. The model was fine-tuned on a small curated subset of the provided dataset where device descriptions were available.

   - Rationale: Large data fine-tuning was computationally infeasible due to hardware constraints; hence, small curated subset fine-tuning provided a memory-efficient way to adapt the pretrained model to trademark-specific visual semantics.

   - The captions for the training were generated using information in the JSON file for each data:

     i. Words in Mark: If wordsInMark is present, it adds "the word {words}" to parts; otherwise, "no words".

     ii. Chinese Characters: If chineseCharacter is present, it adds "Chinese characters {chinese}" to `parts"; otherwise, "no Chinese characters".

     iii. Description of Device: If descrOfDevice is present, it adds "a {descr} shaped device" to `parts"; otherwise, "no particular device or shape".

     iv. Combines these parts into a single sentence starting with "A logo showing " and separated by " and ", ending with a period.

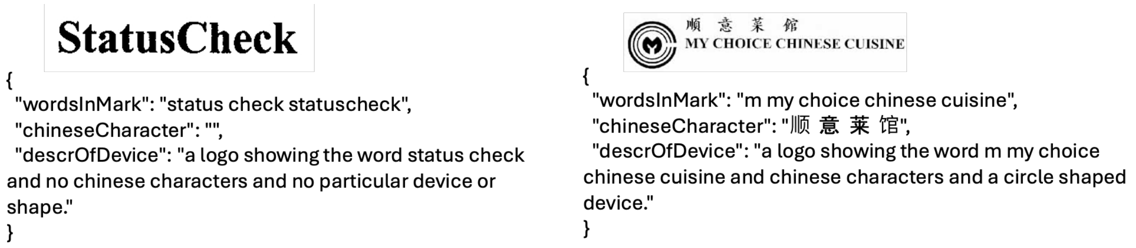
   - Evaluation: Calculated BLEU and ROUGE scores for each generated caption against its reference for 100 test samples.
     i. BLEU score (56 ± 18) indicates a moderate level of n-gram precision, showing reasonable alignment between generated and reference text.
     ii. ROUGE scores suggest strong overlap in both unigrams (ROUGE-1: 94 ± 8) and bigrams (ROUGE-2: 88 ± 14), with a

high ROUGE-L (93 ± 9) value confirming consistent structural similarity.

iii. The low-to-moderate standard deviations indicate that the model’s performance is generally stable across samples, though BLEU shows slightly higher variability (SD ≈ 18.4), suggesting occasional divergence in exact word choices.

- Outcome: The finetuned BLIP model improved alignment between visual features and domain-specific vocabulary such as “abstract emblem,” “stylized letter,” or “circle with arrows.”

3. **Metadata Integration Module** – Combines OCR results and BLIP-generated captions into a structured JSON output that can be indexed or stored in databases.



```
{
  "wordsInMark": "status check statuscheck",
  "chineseCharacter": "",
  "descrOfDevice": "a logo showing the word status check
and no chinese characters and no particular device or
shape."
}
```

```
{
  "wordsInMark": "m my choice chinese cuisine",
  "chineseCharacter": "顺 意 莱 馆",
  "descrOfDevice": "a logo showing the word m my choice
chinese cuisine and chinese characters and a circle shaped
device."
}
```

Figure 2: Sample image and their outputs using proposed solution

Implementation Details

The implementation was modularized to ensure transparency, reproducibility, and maintainability. The following key points summarize the software and environment setup:

- **Programming Language:** Python 3.11
- **Libraries:** torch, transformers, EasyOCR, PIL, json, os, and fastapi.
- **API Deployment:** The FastAPI-based inference service allows users to send an image path or base64-encoded image and receive JSON responses.
- **Docker Support:** A lightweight Dockerfile (CPU variant) was included to containerize the pipeline for cross-platform deployment, with environment variables for OCR configuration and model paths.
- **Limitations:** CPU-based inference on the given MacBook setup (2015) resulted in slower processing speeds (~25–45 seconds per image). Using latest CPU machines would be much faster (<10s per image). GPU based will be under 5s.

Robustness and Error Handling

The system handles corrupted or invalid images by returning structured 400-series errors. The system incorporates basic error-handling mechanisms to ensure stable execution during image processing and inference. Low-confidence OCR triggers an enhanced OCR pipeline with sharpening and autocontrast preprocessing. If OCR fails to index English characters, the retrival is done from the BLIP generated captioning. Each processing stage (OCR, caption generation, and JSON saving) is wrapped in try–except blocks to catch and log common issues such as missing files, invalid image formats, low OCR confidence, or API response failures. When an error occurs, the pipeline gracefully skips the problematic image and records the error message in a local log file, rather than terminating the entire batch process. Default fallbacks (e.g., returning empty strings for missing OCR fields or placeholder text for missing captions) are used to maintain consistent output structure. This design ensures robustness, transparency, and continuity during large-scale dataset processing.

Solution Selection Rationale and Alternatives Considered

The solution evolved through multiple experimental phases, each designed to balance semantic richness, computational feasibility, and domain adaptability. The objective was to automatically generate structured and meaningful textual descriptions of trademark logos from images, integrating both linguistic and visual cues. Three major design directions were explored before converging on the final approach.

CV2-Based Shape Description Approach

The initial idea utilized classical computer vision (CV2) techniques to extract geometric and structural attributes such as contours, edges, and color regions to form heuristic “device descriptions.” This approach was computationally lightweight and compatible with the MacBook Pro (Retina, 13-inch, Early 2015) running macOS Monterey 12.7.6 with 8 GB DDR3 RAM and a 2.7 GHz Dual-Core Intel i5.

However, the method’s semantic limitations quickly became evident: it could only represent basic shapes or textures (e.g., “circle with text,” “red triangle”), lacking the capacity to describe abstract or conceptual visual elements (e.g., “stylized dragon,” “bird in flight”). Furthermore, handcrafted visual descriptors failed to generalize across diverse trademark imagery. Hence, this direction was deemed insufficient for the project’s core goal — semantic, language-rich description generation.

CLIP-Based Image–Text Contrastive Learning

The second exploration involved CLIP (Contrastive Language–Image Pretraining), which maps images and text into a shared embedding space using contrastive learning. The conceptual plan was to fine-tune CLIP so that trademark images align closely with their textual device descriptions. However, CLIP's architecture inherently relies on a predefined list of candidate texts; it identifies the best match for an image by computing similarity scores between image embeddings and those text embeddings.

This makes CLIP well-suited for classification or retrieval tasks, but not for generative captioning, where open-ended textual descriptions are required. Additionally, the approach would have required the curation of a large candidate description library covering the diversity of trademarks — a resource-intensive step. These functional constraints, combined with the lack of GPU acceleration on the available hardware, made this approach impractical for the project's open-domain descriptive goal.

BLIP with Contrastive Fine-Tuning

The third approach explored BLIP (Bootstrapped Language Image Pretraining), which integrates both contrastive and generative capabilities. The idea was to fine-tune BLIP with a contrastive objective on the trademark dataset, aligning image embeddings with descriptive texts while enabling caption generation. However, similar to CLIP, BLIP's contrastive training mode is most effective when provided with a known set of textual candidates for similarity matching, not for producing unconstrained natural language outputs.

Proposed solution

To overcome this limitation and align with hardware feasibility, the final implementation adopted BLIP's generative mode — fine-tuned on a small, domain-specific subset of the dataset. This limited fine-tuning adapted BLIP's language generation to the vocabulary and phrasing typical of trademark descriptions, without requiring extensive GPU resources or massive labeled data. Despite the small dataset size, the fine-tuned model performed well in generating relevant, concise captions aligned with the visual content of logos.

The final solution therefore integrates this fine-tuned BLIP model for image captioning with EasyOCR for multilingual text extraction (English, Simplified Chinese, and Traditional Chinese). EasyOCR was selected over PaddleOCR and Tesseract due to its robust multilingual capabilities and ease of deployment within the FastAPI-based inference API. This hybrid design effectively combines OCR precision with BLIP's semantic understanding to produce structured, interpretable outputs.

Future Enhancements

Several enhancements are envisioned for the next iteration of this system:

- **GPU-Based Fine-Tuning:** Transition fine-tuning to a full model fine-tuning setup using A100 or RTX-class GPUs for improved accuracy and domain adaptation.
- **Multilingual Support Expansion:** Extend OCR and language generation to support trademarks in other languages.
- **Retrieval Integration:** Couple the generated textual representations with a CLIP or BLIP2-based embedding model for multimodal trademark similarity search.
- **Web Interface:** Develop a simple web-based UI for real-time image upload and structured output visualization.

Conclusion

The developed Trademark Indexing API demonstrates a robust, multimodal, and production-ready approach to trademark logo interpretation. By combining OCR and a fine-tuned BLIP model, it ensures accurate multilingual extraction and context-aware visual description. The system is containerized, fault-tolerant, and optimized for both research prototyping and enterprise deployment.

In conclusion, the final design represents a pragmatic balance between semantic quality, computational feasibility, and open-domain flexibility. By using a fine-tuned BLIP model on a smaller dataset, the solution achieves meaningful multimodal grounding within limited local hardware capacity, while remaining easily scalable to more powerful architectures such as BLIP-2, OFA, or Kosmos-2 in future GPU-enabled environments.