Preprint of Graduation Study August 1, 2024

Name Farros Alferro
Supervisor Prof. Takayuki Okatani
Assist. Prof. Masanori Suganuma

Title: Guidance from Keyword for Identifying Small Details in High-Resolution Images with Vision-Language Models

1 Introduction

Recent vision and language models (VLMs) have demonstrated remarkable success, surpassing human performance in certain tasks. However, these VLMs encounter difficulties when recognizing small objects in high-resolution images. To address this limitation, a study [1] introduces a visual search method in which VLMs iteratively search for clues to identify small target objects using fine-tuned vision modules. While this approach achieves superior performance in small object recognition, it incurs high computational costs due to its iterative operations, extensive use of vision modules, and their additional training.

We propose a plug-and-play method that leverages pre-trained model weights without requiring external tools or additional training. This method capitalizes on the vision encoder's inherent ability to contrast images and text, enabling it to selectively focus on image areas relevant to the queries. Consequently, this approach streamlines the process and reduces resource consumption. Check this link for demo.

2 Vision Language Model

2.1 Overall Procedure

Generally, the process of a VLM involves center-cropping and resizing the image to the native resolution of the vision encoder. The image is then processed by the vision encoder, producing image tokens. These tokens are subsequently passed to a Multilayer Perceptron (MLP) for dimension alignment, ensuring they match the embedding dimension of the prompt token. The prompt token itself is generated by passing the prompt through the Large Language Model (LLM) tokenizer. Finally, the image tokens are concatenated with the prompt token, and this combined sequence is processed by the LLM to derive the answer (Figure 1).

Contrary to this general method, we preprocess high-resolution images by resizing them while preserving the aspect ratio, then we crop them into several smaller patches, denoted as $\{P_i\}_{i=1}^{N} \in P_0$, enabling the LLM to recognize small objects. Then the vision encoder extracts the image tokens from these patches, resulting in $V_i^L \in \mathbb{R}^{N \times t_v \times d_v}$, where t_v and d_v denote the number of visual tokens and the visual embedding dimension, respectively.

2.2 Proposed Method

To reduce the computational burden and manage the LLM's context length, we select patches containing the queried object by calculating the cosine similarity between the image representative tokens of the patches (excluding P_0) $r_{i\neq0}^{L} \in \mathbb{R}^{(N-1)\times d_v}$ and the keyword representative token $K \in \mathbb{R}^{d_v}$ (Figure 2 right). Here, K is derived by passing the prompt through the native LLM via a few-shot approach and the text encoder (Figure 2 left). Given that the vision and text encoders were pre-trained through an image-text contrast task, the patch depicting the queried object is expected to exhibit the highest score. However, in some cases, the desired patch may not receive the highest similarity score and could instead be ranked second, third, or lower. To mitigate this, we select the indices of the top-k scoring patches H, where k is a hyperparameter.

$$H = top_k \left(\frac{r_{i\neq0}^{L} K^T}{\left\| r_{i\neq0}^{L} \right\| \left\| K \right\|} \right) \in \mathbb{R}^k \tag{1}$$

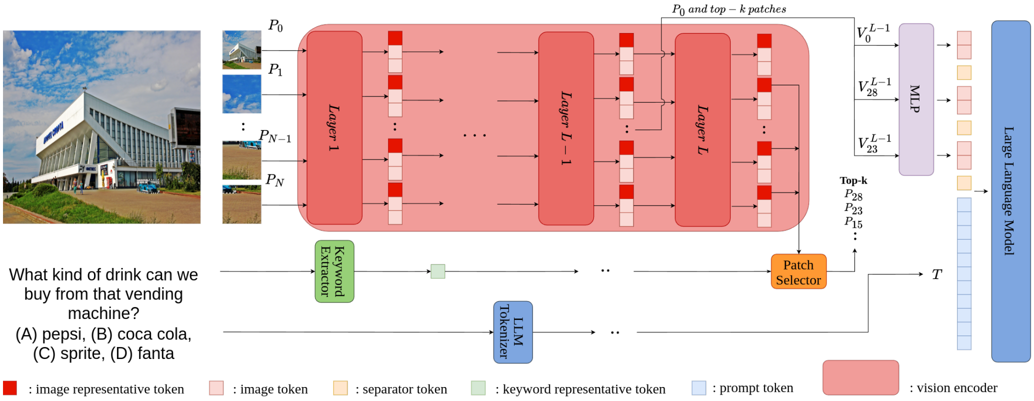

Figure 1: Diagram of the proposed model. We use $k = 2$ for this image.

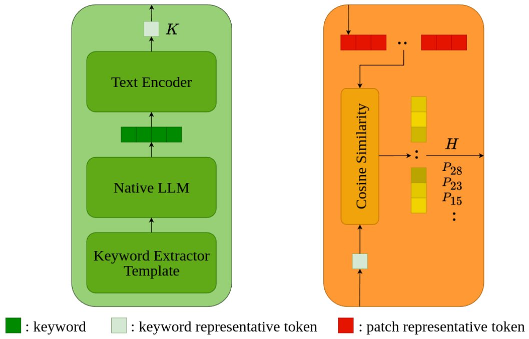
K
Text Encoder
Native LLM
Keyword Extractor Template
Cosine Similarity
H
P_{28}
P_{23}
P_{15}
🟩 : keyword
□ : keyword representative token
🟥 : patch representative token

Figure 2: Proposed plug-and-play modules: (left) keyword extractor module and (right) patch selector module.

Next, we index the image embeddings from the $L-1$ layer using the obtained indices and P_0, resulting in the top patch embedding $V_{\text{topk}}^{L-1} \in \mathbb{R}^{k+1 \times (t_v - 1) \times d_v}$. We then pass V_{topk}^{L-1} through the MLP. After that, we concatenate these embeddings with separator tokens *sep* and the main prompt embedding $T \in \mathbb{R}^{t_t \times d_t}$. This combined sequence is finally fed into the Large Language Model to obtain the answer Y:

$$Y = LLM\Big(\big[MLP(V_{topk}^{L-1}); sep; T\big]\Big) \qquad (2)$$

3 Experiment and Result

We adopted and adapted an existing Vision Language Model from Mantis [2], which was pre-trained on the CC3M 560k image-captioning dataset and fine-tuned using the Mantis-Instruct 721k dataset for multi-image instruction. The model integrates OpenAI's CLIP or Google's SigLIP as the vision encoder, a two-layer multilayer perceptron for alignment, Meta's Llama-3 8B instruct as the Large Language Model (LLM), and employs the text encoder counterpart for patch selection.

For evaluation, we use the V* Bench [1], a visual-question answering benchmark consisting of high-resolution images with abundant and complex information, making it challenging to spot small details. V* Bench consists of two sub-tasks: 1) an attribute recognition task with 115 samples, which requires the model to recognize a certain attribute (color, material, etc.) of an object, and 2) a spatial relationship reasoning task with 76 samples, which asks the model to determine the relative spatial relationship between two objects.

Table 1 shows our model's performance compared with other available vision language models. It can be seen that the proposed method significantly improves the baseline model's overall score where it is mainly boosted by the attribute score. This is due to because the VLM can actually see the queried object in patch level, thus allowing it to answer about object's details. In contrast, the spatial score does not show a significant increase. This may be due to the embedded positional information that does not fit the LLM, thus confusing the LLM about the object's locations even though it actually sees it.

Method	Score (%)↑ A/S/O
Human	98.26/100.00/98.95
Random Guess	26.73/50.00/35.99
Open-source models	
LLaVA-1.5	44.35/52.63/47.64
Mantis-CLIP‡[2]	35.65/61.84/46.07
Mantis-SigLIP‡[2]	33.04/59.21/43.46
High resolution models	
Mini-Gemini 8B	60.87/63.16/61.78
Dragonfly 8B	49.57/52.63/50.79
Tool-using models	
SEAL (SOTA) [1]	**77.39/77.63/77.48**
Private models	
Gemini Pro	40.86/59.21/48.16
GPT-4V	51.30/60.52/54.97
Ours-CLIP	46.96/64.47/53.95
Ours-SigLIP	63.48/61.84/62.83

Table 1: Evaluation of vision language models on V* Bench [1]. A=Attribute, S=Spatial, O=Overall scores. Best results are in bold, and the second-best performances are underlined. ‡ indicates our base model with their respective vision encoder.

Nevertheless, there is still significant score gap between our model and the current state-of-the-art model. This might be caused as our model naively rely on the built-in image-text contrast ability of the vision encoder. A study [3] shows that there is a modality gap between image and text embedding, which can be a plausible reason that in most of the cases the selected patches are not aligned with the keyword. Another plausible reason is the desired object is scattered across patches, thus making it more difficult to perform cosine similarity.

4 Conclusion

We introduce a plug-and-play method integrated into an existing VLM, enabling it to detect small details in high-resolution images without requiring external models or additional training. Our proposed model outperforms both current closed and open-source models in performance. However, further exploration and experimentation are necessary to surpass the current state-of-the-art model.

References

[1] Penghao Wu and Saining Xie. "V*: Guided Visual Search as a Core Mechanism in Multimodal LLMs". In: *CVPR* (2024). URL: https://vstar-seal.github.io/.

[2] Dongfu Jiang et al. "Mantis: Interleaved Multi-Image Instruction Tuning". In: *arxiv* (2024). URL: https://tiger-ai-lab.github.io/Mantis/.

[3] Weixin Liang et al. "Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning". In: *NeurIPS* (2022). URL: https://modalitygap.readthedocs.io/en/latest/.