Commonsense-Guided Open-World Object Detection Using LLMs and Visual-Semantic Matching

Ibrohimjon Muminov, Jihie Kim
Department of Computer Science and AI, Dongguk University, Seoul, South Korea
2024126770@student.dongguk.edu, jihie.kim@dgu.edu

Abstract

Traditional object detectors such as YOLO and Faster R-CNN are limited to fixed category labels and struggle with novel or unseen objects. Open-vocabulary models like CLIP offer greater flexibility but often misinterpret user intent due to a lack of commonsense reasoning.

To address these limitations, we propose a commonsense-guided open-world object detection framework that integrates YOLOv8 for fast region proposals, CLIP for visual-text alignment, LLaVA for scene understanding, and GPT-4 for trait-based reasoning. By pre-generating over 100 object descriptions with GPT-4, our system embeds functional and contextual knowledge that enables intent-aware detection beyond static labels. Experiments on COCO, Open Images, and a custom dataset of unseen objects demonstrate that our approach significantly improves recall on novel queries while maintaining high precision. These results highlight the importance of combining vision-language models with commonsense reasoning for open-world detection.

For implementation details, https://github.com/ibrohimgets/CommonsenseVision.git

I. Introduction

Object detectors like YOLO [1] are efficient but limited to fixed classes, typically trained on datasets like COCO with 80 common categories (e.g., ”dog,” ”car,” ”chair”). However, COCO omits many everyday items (e.g., pens, pencils, markers), causing YOLO to misclassify or miss such objects. This closed-set limitation prevents YOLO from handling flexible, commonsense-driven requests like “something to write with” or “a food item,” where users describe objects by purpose or attributes rather than explicit names. Recent advances in vision-language models like CLIP [2], GPT-4V [3], and LLaVA [4] offer a way to overcome this gap. These models can match images to text descriptions (zero-shot) and reason about objects using commonsense knowledge.

Motivated by this, we propose a commonsense-guided open-world detection framework. Our system combines YOLOv8 for proposing regions, CLIP for matching user prompts, GPT-4 for generating trait-based object descriptions, and LLaVA for scene-level understanding. This enables flexible detection — such as grounding “something to write with” to a pencil / pen / marker — without requiring real-time LLM queries.

Our main contributions are:

- **Commonsense-Guided Detection:** A system combining YOLOv8, CLIP, GPT-4 trait knowledge, and LLaVA to detect objects based on user intent.
- **Trait-Based Matching:** A pre-generated GPT-4 knowledge base enables fast, flexible matching without live inference costs.
- **Improved Flexibility and Accuracy:** Our system outperforms CLIP-only and YOLO-only baselines, successfully interpreting natural, commonsense queries.

In the following sections, we review related work, describe our methodology (Figure 1), present experiments, and discuss future directions.

II. Related Work

A. Closed-Set Object Detection

Traditional object detectors like Faster R-CNN, SSD, and YOLO are limited to fixed classes and struggle with recognizing unseen concepts [1]. For example, YOLOv8 performs well on COCO’s 80 categories but fails to detect objects like ”pen” or ”pencil” if not included in training. YOLO-World addresses this by extending YOLOv8 with a text encoder and vision-language pretraining [5], using a ”prompt-then-detect” strategy with an offline vocabulary [6]. While it improves flexibility, it still relies on keyword matching and lacks the contextual reasoning needed to truly understand user intent.

B. Vision-Language Models for Open-Vocabulary Detection

Recent methods integrate vision-language models to move beyond fixed labels. CLIP-based approaches such as ViLD, RegionCLIP, and Detic use image-text embeddings for large-vocabulary detection [2], with RegionCLIP enhancing region-level alignment [7]. GLIP reformulates detection as a query-based task using free-text prompts [8]. However, studies like DeSCo [9] show that these models often miss subtle context and over-rely on object names. Our method builds on these works by using an LLM to enrich queries with commonsense traits and leveraging LLaVA for deeper scene understanding [4].

C. Large Language Models for Vision and Reasoning

Multimodal LLMs like GPT-4V and LLaVA show that LLMs can interpret and reason about images by integrating

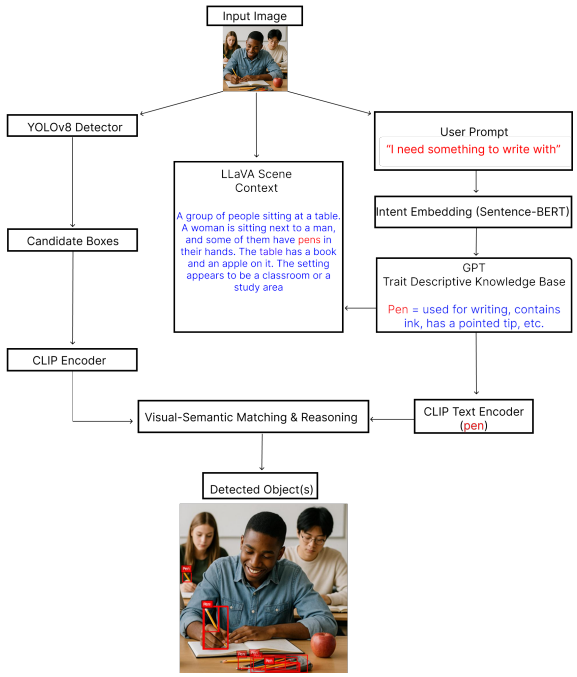

Fig. 1. Overview of our commonsense-guided open-world detection framework. Given an image and a user prompt (e.g., "I need something to write with"), YOLOv8 proposes regions, and CLIP encodes visual features. Sentence-BERT encodes the user's intent and matches it to GPT-generated trait descriptions (e.g., "pen = used for writing, contains ink"). LLaVA provides scene context to support reasoning. The system integrates all components via visual-semantic matching to identify the most relevant object.

visual features into a general reasoning engine [3], [4]. In our framework, we use LLaVA to generate high-level scene descriptions (captions and object lists) to support object detection. LLMs have also been used to provide commonsense priors for tasks like VQA and robotics [10]. We extend these ideas by combining GPT-4's offline-generated trait knowledge with CLIP's efficient online matching. Since querying GPT-4V per image is costly and slow, we pre-compute traits offline to enable fast and scalable open-world detection.

III. Proposed Method

Our system consists of four main components (Figure 1): (1) YOLOv8 for region proposals, (2) CLIP for visual-textual matching, (3) LLaVA for scene-level context, and (4) a GPT-4-generated trait knowledge base for commonsense reasoning.

A. YOLOv8 Candidate Proposals

We use YOLOv8 (pre-trained on COCO) as a class-agnostic region proposal network by selecting all high-confidence boxes and ignoring predicted labels. This efficiently narrows the search space to a few likely object regions, even when YOLO misclassifies items like pencils, while maintaining real-time speeds [1].

B. GPT-4 Trait Description Knowledge base

We curated over 100 object concepts, including all 80 COCO classes and various everyday items (e.g., writing tools, utensils, devices). GPT-4 generated detailed traits for each object, describing attributes, functions, and typical contexts (e.g., a "pen" is "used for writing on paper," "contains ink," "found on office desks"). These structured traits are stored in a JSON knowledge base and enable commonsense-driven matching without requiring real-time LLM queries. At inference, user prompts are semantically matched to trait descriptions. For flexible prompts like "something to write with," both the prompt and traits are encoded using Sentence-BERT, and the best match is selected via cosine similarity. This offline strategy avoids the latency and cost of live GPT queries.

C. CLIP Visual-Semantic Matching

CLIP matches YOLO proposals to the user's described object. The target trait description (e.g., "made of plastic, used for writing") is encoded by CLIP's text encoder (512-dimensions). Each YOLO region is cropped and encoded using CLIP's image encoder. We compute cosine similarity between the text and image embeddings:

$$s_i = \frac{f_{\text{img}}(B_i) \cdot f_{\text{text}}(\text{query})}{|f_{\text{img}}(B_i)||f_{\text{text}}(\text{query})|} \tag{1}$$

The region with the highest score above a threshold is selected; otherwise, the system outputs "object not found."

When multiple trait sentences exist (e.g., synonyms), we average the similarity scores:

$$s_i^{\text{traits}} = \frac{1}{T} \sum_{t \in \mathcal{T}} \frac{f_{\text{img}}(B_i) \cdot f_{\text{text}}(t)}{|f_{\text{img}}(B_i)||f_{\text{text}}(t)|} \tag{2}$$

This approach leverages CLIP's zero-shot capability [2], while reducing computation compared to naive sliding window scanning.

D. Sentence-BERT Embedding for Flexible Queries

For flexible prompts, we use Sentence-BERT to encode the user query (384-dimensional embeddings) and match it against pre-generated trait embeddings. This allows the system to understand queries even when the object name is not explicitly stated.

E. LLaVA Scene Understanding

LLaVA provides global scene descriptions (e.g., listing desks, whiteboards, stationery in a classroom) [4]. We use LLaVA outputs to adjust the CLIP matching threshold dynamically:

If the queried object is mentioned, the threshold is lowered to increase sensitivity.

If unrelated, the threshold is raised to reduce false positives.

This soft, rule-based adjustment improves robustness without hard-coding object lists. More advanced LLM-driven integration is left for future work.

IV. Experiments

A. Dataset

For evaluation, we use three sources: (1) the COCO 2017 validation set for known class detection, (2) a curated subset of images from the Open Images dataset, and (3) a custom benchmark of 30 novel objects not included in YOLOv8's label space (e.g., pens, whiteboards, toothpaste, coffee cups).

Each image is paired with a natural language query—ranging from explicit object names to high-level intent-based prompts (e.g., "I need something to write with", "a travel bag"). A detection is considered correct if the predicted bounding box achieves an Intersection-over-Union (IoU) ≥ 0.5 with the ground truth.

B. Our System

Our system combines YOLOv8 candidate proposals, CLIP-based visual-semantic matching, GPT-4-generated trait descriptions, and optional LLaVA scene context [4]. We also perform ablation by removing trait expansion and scene context to assess their impact.

C. Implementation Details

The settings are as follows:

- **YOLOv8 (small variant):** Confidence threshold set to 0.25.
- **CLIP:** ViT-B/32 backbone.
- **Trait Matching:** Sentence-BERT embeddings.
- **Trait Generation:** GPT-4 (March 2024 version).
- **Scene Context:** LLaVA (Vicuna-13B v1.5).

V. Results

We evaluated our commonsense-guided detection system in two stages:

- **(1) Standard Class Detection (Known Classes)**
- **(2) Flexible Commonsense Prompt Detection**

Table I shows precision and recall for known COCO classes, while Table II evaluates detection on 30 novel objects outside YOLOv8's label set. Table III further assesses the system's ability to interpret flexible, intent-driven prompts using commonsense reasoning.

TABLE I
Detection Performance for Known Classes (e.g., 'cat', 'dog', 'car') at IoU ≥ 0.5

Method	Precision (P)	Recall (R)
YOLO-only (COCO classes: cat, dog, car)	0.91	0.52
CLIP-only (sliding window)	0.45	0.78
Ours	0.84	0.82

YOLO-only performs well on known classes but fails on unseen objects due to its fixed vocabulary. CLIP-only generalizes better but lacks precise localization. Our system combines YOLO, CLIP, and GPT-based reasoning to achieve strong performance on both known and novel objects.

TABLE II
Detection Performance on Novel Commonsense Benchmark (30 unseen objects)

Method	Precision (Novel)	Recall (Novel)
YOLO-only (COCO trained)	0.40	0.20
CLIP-only (sliding window)	0.52	0.65
Ours (Commonsense-guided)	**0.74**	**0.71**

These results reveal the limits of traditional detectors on unseen objects and highlight the effectiveness of commonsense-guided reasoning for real-world intent. Our benchmark offers a strong foundation for future open-world detection research.

TABLE III
Performance on Commonsense Prompts (Flexible Intent Understanding)

Prompt	Method	Result
I need a travel bag	YOLO-only	Failed
I need a travel bag	CLIP-only	Partial match
I need a travel bag	Ours	Correct (suitcase detected)
A furry animal	YOLO-only	Failed
A furry animal	CLIP-only	Partial match
A furry animal	Ours	Correct (cat detected)

Table III illustrates how the system interprets flexible prompts using commonsense reasoning—such as detecting a cat from "a furry animal" or a suitcase from "a travel bag"—without being explicitly told the object name.

VI. Conclusion

We proposed an open-world object detection system that combines YOLOv8, CLIP, LLaVA, and GPT-4 to integrate visual grounding with commonsense reasoning. By leveraging pre-generated trait knowledge, our method bridges user intent and perception, significantly improving recall on novel objects while maintaining high precision. Future work includes scaling trait coverage, enabling dynamic LLM updates, and optimizing scene understanding with lighter models.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2020-II201789), and the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2025-RS-2023-00254592) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation).

References

[1] Ultralytics, "Yolov8: Next-generation object detection," https://github.com/ultralytics/ultralytics, 2023, gitHub Repository.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," https://arxiv.org/abs/2103.00020, 2021, arXiv preprint.

[3] OpenAI, "Gpt-4v(ision): Expanding gpt-4 into visual modalities," https://openai.com/research/gpt-4v, 2023, openAI Research.

[4] H. Liu, C. Zhang, Y. Xu, H.-Y. Lee, Y. Wang, J. Zhang, Z. Liu, J. Wang, Y. Wang, J. Chen *et al.*, "Llava: Large language and vision assistant," https://llava-vl.github.io/, 2023, gitHub Project.

[5] F. Fang, J. Zhang, Y. Wang, Y. Wu, C. Xu, H. Duan, and D. Lin, "Yolo-world: Open-vocabulary object detection," https://arxiv.org/abs/2401.00497, 2024, arXiv preprint.
[6] Ultralytics, "Yolo-world documentation," https://docs.ultralytics.com/tasks/open-vocabulary/, 2024, ultralytics Docs.
[7] X. Hu, X. Gu, J. Yang, Z. Zhang, X. Wang, J. Luo, and J. Gao, "Regionclip: Region-based language-image pretraining," https://openaccess.thecvf.com/content/CVPR2022/html/Hu_RegionCLIP_Region-Based_Language-Image_Pretraining_CVPR_2022_paper.html, 2022, cVPR 2022.
[8] L. H. Li, J. Zhang, X. Yang, X. Hu, L. Wang, T. Darrell, and J. Gao, "Glip: Grounded language-image pretraining," https://openreview.net/forum?id=HLlA-MykP7, 2022, openReview.
[9] X. Wang, X. Hu, D. Lin, and P. Luo, "Desco: Descriptive context modeling for open-vocabulary object detection," https://arxiv.org/abs/2308.14709, 2023, arXiv preprint.
[10] Y.-C. Chen, L. Yu, T. Shu, Z. Wang, K.-W. Chang, L. Zettlemoyer *et al.*, "Language models for commonsense reasoning in visual question answering," in *Springer Computer Vision and Pattern Recognition*, 2021.