Scene Localization in Dense Images via Natural Language Queries

1. Introduction

This project addresses the problem of localizing specific interactions within dense images using natural language queries. In scenarios like surveillance, autonomous navigation, or contextual scene analysis, images often depict multiple simultaneous activities. The goal is to ground a textual query such as *“a vendor selling vegetables”* into the correct sub-scene region of a high-resolution image.

The system should:

- Understand dense visual contexts.
- Parse free-form natural language queries.
- Output a bounding box or cropped image region that corresponds to the query.

---

2. Problem Statement

Input:

- A single dense image (H × W × 3).
- A natural language query (e.g., *“a man snatching a chain”*).

Output:

- Bounding box coordinates `(x1, y1, x2, y2)` or a cropped image containing the described interaction.

Deliverables:

- A working prototype capable of processing dense images and queries.
- Documentation and a short demo video.

---

3. Approach and Methodology

Step 1: Grounding DINO + SAM

- Tried text-conditioned detection (Grounding DINO) + segmentation (SAM).
- **Challenge:** Integration issues with updated Hugging Face versions caused constant errors.
- **Result:** Pipeline not usable.

Step 2: Custom Hugging Face Model

- Built a dataset and trained a model.
- **Challenge:** Dataset mismatch, Colab crashes, runtime disconnects.
- **Result:** Poor and inconsistent performance.

Step 3: YOLO + CLIP Integration (Final)

- Combined YOLO for detection with CLIP for text-image similarity.
- **Process:**
    - YOLO detects candidate bounding boxes.
    - CLIP encodes each crop + query, computes cosine similarity.
    - Highest similarity score → returned as the final match.
- **Result:** Acceptable performance; became the final working pipeline.

Step 4: Fine-Tuning Attempts on Custom CVAT Dataset

- Collected market-like images (Khan Market, Central Market, Mumbai) and annotated with CVAT.
- Fine-tuned YOLO on this dataset.
- **Challenge:** Accuracy dropped, likely due to dataset mismatch and limited size.
- **Result:** Failed to improve performance.

Step 5: Testing with YOLO Darknet Dataset (Online)

- Used an existing dataset (YOLO Darknet format) found online.
- **Result:** Achieved **good results** when queries matched objects in this dataset.
- This highlighted the importance of high-quality, well-aligned datasets.

Step 6: Owlv2

- Tried open-vocabulary detection.
- **Challenge:** Very slow on Colab, poor accuracy in dense scenes.
- **Result:** Dropped.

---

4. Final Pipeline (YOLO + CLIP)

1. Input: Dense image + natural language query.
2. Detection: YOLO generates candidate bounding boxes.
3. Matching: CLIP compares query with each crop.
4. Output: Bounding box or cropped region that best matches the query.

---

5. Challenges Faced

- **Data Collection:** Lack of scene-level interaction datasets.
- **Colab Limitations:** Frequent crashes, temporary storage loss.
- **Training:** Custom CVAT dataset failed to improve model performance.
- **Inference:** Owlv2 was too slow.

---

6. Results

- Queries like *“a person with a dog”* or *“a vendor selling vegetables”* returned relevant crops.
- Failed on the custom CVAT dataset.
- **Worked well on the YOLO Darknet dataset** found online, which had clean annotations and object classes.

---

7. Future Improvements

- Collect or build a large-scale, **high-quality dataset** for interaction-level tasks.
- Explore fine-tuned CLIP variants for better text understanding.
- Migrate to stable GPU platforms (Paperspace, Kaggle) instead of Colab.

---

8. Conclusion

After testing Grounding DINO, SAM, Hugging Face models, fine-tuned YOLO, and Owlv2, the **YOLO + CLIP integration** emerged as the most practical. It offered reasonable speed and accuracy.

Although fine-tuning with the custom CVAT dataset failed, **the pretrained YOLO + CLIP pipeline performed well on the YOLO Darknet dataset**. This demonstrated that model success is strongly tied to dataset quality and alignment.

---

9. Personal Reflection

Working on this project was a transformative experience. It was my first time building a vision-language pipeline from scratch, and every step taught me something new. I experimented with multiple approaches—Grounding DINO + SAM, custom Hugging Face models, Owlv2, and finally YOLO + CLIP. Along the way, I encountered setbacks like Colab crashes, integration errors, and disappointing results from fine-tuning, but each challenge deepened my understanding.

Even though the fine-tuned YOLO model failed on my custom dataset (which I collected and annotated using CVAT with real-world market scenes), it still gave encouraging results on an online YOLO Darknet dataset. This showed me that data quality and alignment are crucial, and that even state-of-the-art models depend heavily on the right dataset.

Beyond technical skills, this project taught me resilience, persistence, and problem-solving under constraints. I am genuinely proud of the working prototype I built in just a week, and I feel motivated to keep pushing myself further in multimodal AI.

---

10. Acknowledgements

I sincerely thank **AIMS** for giving me this project and for pushing me to explore multimodal AI deeply. I am grateful to the mentors, organizers, and reviewers who designed such a challenging and thought-provoking task.

Your guidance didn’t just help me build a prototype—it helped me learn more than I could have imagined in such a short time. I truly appreciate the opportunity and the trust you placed in me.