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

Given:

- **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: Initial Exploration with Grounding DINO + SAM

- **Tried:** Grounding DINO for text-conditioned object detection + SAM for segmentation.
- **Challenge:** Integration issues with updated Hugging Face versions; constant errors despite tutorials.
- **Result:** Pipeline could not be completed.

Step 2: Attempt at a Custom Model

- **Tried:** Built a model using Hugging Face datasets.
- **Challenge:** Dataset mismatch, Colab runtime disconnects and crashes.
- **Result:** Performance was poor, no consistent improvements.

Step 3: YOLO + CLIP Integration

- **Tried:** Combined YOLO (object detection) with CLIP (text-image similarity).
- **Implementation:**
    1. YOLO detects candidate bounding boxes.
    2. CLIP encodes crops + query and computes cosine similarity.
    3. The best match returned as the final crop.
- **Result:** Gave acceptable results. Became the final working pipeline.

Step 4: Fine-Tuning Attempts

- **Tried:** Fine-tuned YOLO with a custom dataset I collected from various online sources to mimic scenes like Khan Market, Central Market, Mumbai markets, etc., annotated using CVAT.
- **Challenge:** Accuracy dropped in many cases, possibly due to overfitting or dataset mismatch.
- **Result:** Did not give satisfactory improvements; I regretfully had to revert to the pretrained YOLO + CLIP model.

Step 5: Exploring Owlv2

- **Tried:** Tested **Owlv2** (open-vocabulary detection).
- **Challenge:** Very slow on Colab, poor results on dense scenes.
- **Result:** Dropped in favor of YOLO + CLIP.

---

4. Final Pipeline (YOLO + CLIP)

- **Input:** Dense image + natural language query.
- **Detection:** YOLO detects candidate regions.
- **Matching:** CLIP compares query with each crop.
- **Output:** Bounding box or cropped region that best matches the query.

---

5. Challenges Faced

- **Data Collection:** Lack of publicly available datasets for scene-level interaction localization.
- **Colab Limitations:** Frequent restarts, temporary storage loss, GPU crashes.
- **Training:** Fine-tuning often worsened results instead of improving them.
- **Inference:** Owlv2 too slow, not practical for dense images.

---

6. Results

- Prototype can process queries like *“a person with a dog”* or *“a vendor selling vegetables”* and return relevant cropped regions.
- Performance is moderate but usable. Achieved within a short timeframe.
- Demo video submitted separately.

---

7. Future Improvements

- Build a larger, high-quality custom dataset for scene-level interactions.
- Explore advanced multimodal models on stable GPU platforms.
- Improve query understanding with fine-tuned CLIP variants.
- Use persistent cloud platforms (Kaggle, Paperspace) instead of Colab.

---

8. Conclusion

After experimenting with Grounding DINO, SAM, custom models, fine-tuned YOLO, and Owlv2, the most practical solution was YOLO + CLIP integration. This offered a balance of speed, reliability, and reasonable accuracy. The project highlights the importance of data quality, resource constraints, and model trade-offs in multimodal vision-language research.

---

9. Personal Reflection

Working on this project was an exciting learning experience. Since it was my first time building a deep learning model from scratch, every step was a challenge and opportunity to grow.

I explored multiple approaches—Grounding DINO + SAM, custom Hugging Face models, Owlv2, and finally YOLO + CLIP. Frustrations included Colab crashes, integration issues, and fine-tuning challenges.

Collecting and annotating my own dataset of real-world markets (Khan Market, Central Market, Mumbai markets) with CVAT was rewarding, but fine-tuning did not improve results. Ultimately, I relied on the pretrained YOLO + CLIP pipeline.

Through this project, I gained technical knowledge in vision-language models, as well as resilience, patience, and problem-solving skills. Even though results weren't perfect, the journey of experimenting and learning was incredibly valuable.

I am grateful to AIMS for providing this opportunity, which motivated me to explore multimodal AI intensively in just one week.

---

10. Acknowledgements

I sincerely thank AIMS for providing this project and a platform to learn, explore, and grow.

I am grateful to mentors, organizers, and reviewers for their effort in designing a thought-provoking, practical problem. This assignment not only improved my technical skills but also strengthened my confidence in handling real-world AI challenges.