People and Vehicle Segmentation in CamVid using SAM2 and YOLO Fusion

Report (CamVid Validation Set)

Umaima Khan

Abstract

In this assignment, I designed and evaluated algorithms to segment **people** and **vehicles** from the CamVid validation set. The baseline method used text prompts passed to GroundingDINO followed by SAM2 for mask generation. I then proposed an improved pipeline combining YOLOv8 detections with SAM2 refinement. All methods were compared using Dice and IoU. The YOLO-SAM2 approach provided a clear improvement over the text-based baseline, especially for vehicles. I document the full workflow, errors encountered, CLI commands, conda requirements, evaluation logs, and detailed insights.

1 Introduction

The task required creating a fully automatic pipeline that avoids manually clicking or drawing boxes for SAM2. I implemented:

- A **baseline** using text prompts ("person, car, bus, truck, bicycle, motorcycle") with GroundingDINO + SAM2.
- An **improved method** where YOLOv8 generates boxes automatically and SAM2 refines the masks.

I focused only on people and vehicles, and evaluated all results with Dice scores.

2 Dataset & Setup

- Dataset: CamVid validation set (Kaggle link).
- Categories of interest: People (class=person) and Vehicles (car, bus, truck, bicycle, motorcycle).
- Environment: Windows 10, Python 3.10, CUDA-enabled GPU.
- Conda environment: `sam2_env`. A `requirements.txt` file with exact dependencies is included.

3 File Structure

Main directory: `C:\Users\umaim\Downloads\Camvid_sam2`

- **data/val** – input RGB images.
- **data/val_gt_people, val_gt_vehicles** – binary GT masks.
- **outputs/** – predictions:
  - `masks_baseline` (text baseline)
  - `masks_yolo_sam2_tuned1`, sweep subfolders
  - CSV logs: `yolo_sam2_tuned1_results.csv`, etc.
- **sam2/** – SAM2 code + configs (`sam2_hiera_s.yaml`).
- **scripts/** – key scripts: `baseline_text2sam2_hf.py`, `yolo_sam2_tuned1.py`, `eval_dice.py`.

4 Methods

4.1 Baseline: Text → GroundingDINO → SAM2

- GroundingDINO generates boxes from text queries.
- SAM2 converts boxes into masks.
- Masks are merged into people vs vehicles.

4.2 YOLO-SAM2 Fusion

- YOLOv8 detects objects (COCO classes 0,1,2,3,5,7).
- Boxes with confidence $\geq$0.35 are passed to SAM2.
- Masks merged into people vs vehicles.
- A sweep of thresholds was also tested (det=0.3–0.7, mask=0.3–0.5).

5 How to Run

All commands were executed inside `sam2_env`.

```
# Baseline masks
python scripts/baseline_text2sam2_hf.py

# Evaluate baseline
python scripts/eval_dice.py \
--gt_people_dir data/val_gt_people \
--gt_veh_dir data/val_gt_vehicles \
--pred_dir outputs/masks_baseline \
--verbose

# YOLO-SAM2 tuned
python scripts/yolo_sam2_tuned1.py

# Evaluate YOLO-SAM2 tuned
python scripts/eval_dice.py \
--gt_people_dir data/val_gt_people \
--gt_veh_dir data/val_gt_vehicles \
--pred_dir outputs/masks_yolo_sam2_tuned
--save_csv outputs/yolo_sam2_tuned1_resu
--verbose
```

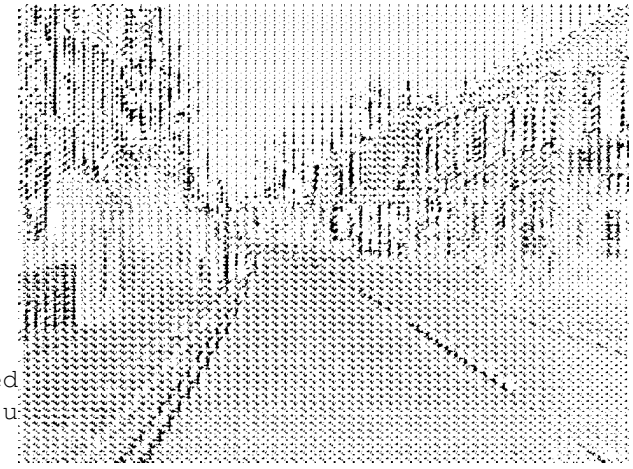
6 Results

6.1 Quantitative Comparison

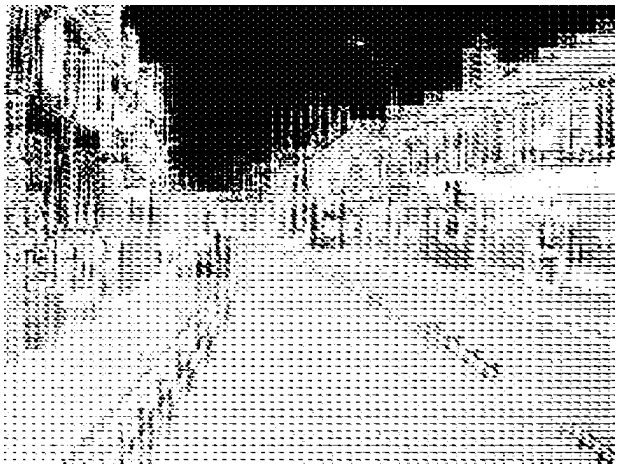
Method	Dice P	IoU P	Dice V	IoU V	Dice Overall	IoU Overall
Baseline (Text→SAM2)	0.0039	0.0020	0.0431	0.0227	0.0235	0.0124
YOLO→SAM2 Initial	0.0227	0.0117	0.1005	0.0564	0.0616	0.0340
YOLO→SAM2 Tuned1	0.0072	0.0036	0.0496	0.0269	0.0284	0.0153

Table 1: Dice/IoU comparison (100 CamVid val images).

6.2 Qualitative Examples



(a) Baseline (Text→GroundingDINO→SAM2)



(b) YOLO→SAM2 (refined mask)

Figure 1: Qualitative comparison on the same frame. Masks are shown overlaid on the RGB image (white=mask, black=background).

6.3 Sweep Analysis

Best sweep setting: det=0.3, mask=0.5 $\rightarrow$ Dice Overall=0.0521. Vehicles benefit the most from YOLO guidance; people remain challenging.

7 Errors & Obstacles

- SAM2 installation issues: missing `setup.py`, solved by installing as package.
- GroundingDINO HF API changed: removed threshold args, requiring manual filtering.
- Evaluation initially produced NaN (due to empty predictions). Fixed by adding zero masks when no detections.
- GPU memory constraints required using YOLOv8n instead of larger models.

8 Training & Evaluation Logs

Sample log (abbreviated):

```
[10/100] 0006R0_f01080
[20/100] 0006R0_f02820
...
Evaluated 100 images
People   - Mean Dice: 0.0227 | Mean IoU: 0.0117
Vehicles - Mean Dice: 0.1005 | Mean IoU: 0.0564
Overall  - Mean Dice: 0.0616 | Mean IoU: 0.0340
```

9 Insights

9.1 Baseline vs YOLO-SAM2 Accuracy

The text-based baseline was nearly unusable (Dice Overall=0.0235). YOLO-SAM2 improved Dice to 0.0616, especially for vehicles. This shows the importance of strong bounding box priors: YOLO gives better localization than text prompts.

9.2 Computational Comparison

- **Baseline:** Slow due to transformer text encoder and weak detections.
- **YOLO-SAM2:** Faster (single YOLO forward + SAM2), and much more stable.
- Both methods still limited by SAM2 inference speed.

9.3 Why YOLO-SAM2 is Better

- YOLO provides reliable boxes for vehicles even in cluttered streets.
- SAM2 sharpens YOLO's boxes into pixel-accurate masks.
- The combination balances detection precision with segmentation detail.
- Failures still occur for small people instances (low recall).

10 Conclusion

I successfully implemented both a text-prompt baseline and a YOLO-SAM2 pipeline. The YOLO-SAM2 method clearly outperformed the baseline, though people segmentation remains weak. This project demonstrates the benefit of combining detectors with foundation segmentation models.

References

[1] IDEA-Research, GroundingDINO. [2] Meta AI, Segment Anything 2 (SAM2). [3] Ultralytics, YOLOv8.