Samprith Kalakata (srk9068)
N16994838

CS-GY 6613 - Artificial Intelligence INET
Assignment 2 Report

---

1. **Data Sources and Prep**
    a. Corpus video: YouTube ID YcvECxtXoxQ (Toyota RAV4 review). Downloaded as input_video.mp4.
    b. Frame sampling: ffmpeg -vf "fps=1/5" → 1 frame every 5 seconds. Produced 559 frames (frame_0001.jpg … frame_0559.jpg) covering 0–2,790 s. Stored in assignments/assignment-2/frames/.
    c. Query set: Hugging Face dataset aegean-ai/rav4-exterior-images (65 exterior frames at 5 s spacing, stored remotely; loaded on the fly in Part C). Each row includes image, timestamp, and metadata.

2. **Detector Choice & Training (Part A) – part_a_finetune_seg_model.py**
    a. Model: Ultralytics YOLO v26 segmentation (yolo26n-seg.pt) fine-tuned on the Carparts segmentation dataset.
        i. Rationale: Pretrained part-level labels, fast inference, and segmentation masks (though downstream we keep boxes).
    b. Training configs:
        i. Local notebook-friendly run: 1 epochs, 512 img size, batch 8, workers 4, AMP off on Apple M-series (device = "mps" fallback to CPU) (Used AI help to get memory safe defaults, see the script for details)
        ii. GPU run for final model:
            1. 20 and 50 epochs runs on Lightning.AI A100 (script variant in comments). Resulting checkpoints saved as part_a_best.pt and part_a_best_50_epochs.pt respectively. Both copied locally for inference and uploaded to Github as well (Stable artifacts reside in assignments/assignment-2/models/)

3. **Detection Corpus (Part B) – part_b_retrieval_corpus.py**
    a. Inputs: sampled frames + part_a_best_50_epochs.pt.
    b. Inference settings: confidence threshold 0.25; uses YOLO segmentation but stores bounding boxes; records frame index and timestamp (derived from 5 s stride).
    c. Output schema (per detection): video_id, frame_index, timestamp_sec, class_id, class_label, confidence_score, bounding_box ([x_min, y_min, x_max, y_max]), individual box coords, frame_file, detector_name.
        i. See README for description of schema
    d. Main artifact: video_detections_50_epochs.parquet

i. 1,100 rows; 370 unique timestamps; frame range 1–559; time range 0–2,790 s.
ii. Top classes (counts): trunk 179, wheel 170, hood 117, front_glass 88, right_mirror 76, front_bumper 71, front_left_light 60, back_glass 56, back_bumper 55, back_left_light 53.
iii. Confidence: mean 0.47, median 0.40.
iv. Intervalization (5 s gap rule) by class (count / avg duration in seconds): trunk 79 / 4.7; wheel 41 / 8.3; hood 47 / 5.7; front_glass 23 / 13.3; front_bumper 21 / 11.9; right_mirror 23 / 8.7; others mostly short (many single-frame, 0–2 s).

e. Secondary artifact (20-epoch model): video_detections.parquet with 911 rows; kept for comparison but not used downstream.

4. Retrieval Logic (Part C) – part_c_semantic_search.py

a. Steps:
i. Load part_a_best_50_epochs.pt.
ii. Read corpus detections from video_detections_50_epochs.parquet.
iii. Build a class→interval index: for each class, unique timestamps are merged into contiguous intervals if the gap ≤ 5 s (frame stride). Supporting detection count tallied per interval.
iv. For each query image (65 total), run detector (threshold 0.25) to obtain labels.
v. For each detected label, return all matching intervals from the index; if no matching intervals exist, emit a null interval; if no labels are detected, emit a fully null record.

b. Output: query_retrieval_results.parquet
i. Rows: 7,550 (one row per query×interval).
ii. Coverage: 65 queries; 4 queries produced no intervals (indices 30, 36, 38, 58).
iii. Interval rows: 7,546 with timestamps; average interval duration 5.48 s (median 0.0 s because many single-frame hits), max 130 s; supporting detections mean 2.58, max 32.
iv. Label usage (rows): wheel 1,599; trunk 948; back_bumper 920; hood 846; back_glass 490; front_glass 460; front_bumper 399; others tail off.
v. Intervals per query: mean 124, median 122, max 235 — reflects the label-only matching strategy.

c. Conclusion is retrieval is class-label exact-match with no visual embeddings or spatial reasoning and temporal grouping follows frame stride.

5. Steps to reproduce

a. (Optional - I uploaded the model weights I used to GitHub) Re-train detector using part_a_finetune_seg_model.py

i. For full 50-epoch quality, run the commented Lightning variant on a CUDA box and place the resulting part_a_best_50_epochs.pt into assignments/assignment-2/models/.

b. Build detection corpus
   i. Download & sample video as in README and ensure frames/ exists
   ii. Run part_b_retrieval_corpus.py to build detection corpus

c. Run semantic search
   i. Run part_c_semantic_search.py to create a car part (class_label) to intervals from detection corpus mapping and use that to run semantic search

6. **Failure cases and limitations**
   a. Temporal resolution limited to 5 s sampling; brief appearances can be missed or collapse to 0 s intervals
   b. Retrieval is label-only: no visual similarity check within the class; results can include every interval containing that label, leading to many intervals per query and potential false positives.
   c. Some classes underrepresented (doors, tailgate) and some intervals are single-frame, giving zero-duration windows.
   d. Confidence threshold fixed at 0.25; no calibration or per-class tuning; masks unused (only boxes preserved).
   e. Dataset/domain gap: query set and corpus both from the same video family but still show slight distribution shift; four queries produced no detections.