Patrick Durzynski

CS 370-Intro to Artificial Intelligence

02 February, 2026

CS 370 Assignment 2 Report

This assignment required us to use an object detector on images of cars extracted from a given video. We then detect various car parts in those images. After that, we are given a new dataset of query images, and we must perform a semantic search on our part 1 database to find matching clips for every image and its timestamp.

Section 1: Detector choice and configuration

I used YOLOv26 segmentation as provided by the Ultralytics segmentation Colab template. For the training, I worked wth 15 epochs with batches of 16 and 64 workers. This was also given by the Ultralytics colab template. However, I reduced batches from 32 to 16, as during runtime, it couldn't handle 32 and changed it to 16 automatically. I decided to raise the confidence level from 0.2 in the template to 0.3. I believed it's the right balance of getting as many frames as possible while still having some quality control.

Section 2: Video sampling strategy

I went with the strategy of sampling the entire video at 1 frame per 5 seconds. This was done to reduce time and excessive computation, while still getting a variety of angles of the car. I used the trained object detector on these frames and stored the results in the detections.paquet file.

Section 3: Image-to-Video Matching Logic

We were given a dataset consisting of various images. For each query image, we would have to do the following:

First, we use the same YOLOv26 model I used on the video frames for the query image. I extract the class labels and search for the detection.parquet file for frames containing the same class(es). I used a function to extract clips with a 10-second threshold between detections. The function would store a clip variable containing the start time, end time, and number of detections. That clip would be appended to a list, which would be saved into a datframe, which itself would be saved as ‘semantic_results.csv.’

Section 4: Failure cases and limitations

I had a lot of failures during this assignment. Some were solved by reordering my code or fixing types, like a case where my semantic dataset was empty, or the array holding all my detections only had 1 entry. I also had an issue when it came to each matching clip’s start and end time being the same, so I

increased the gap between detections in clips and placed a minimum duration for each detection of 1 second.

However, my first big issue was the fact that the timestamps and detection were not accurate. For example, one of my entries was the timestamp 14-19 that showed the back_glass. However, in the video, that timeframe shows two shots, neither of them with back_glass. I get how the second shot could have been an error. The front hood was up, so it could have confused the front glass with the back. However, the first shot was underneath the car, so there was no possible way it should have confused any part as the back glass! By increasing the confidence level initially from 0.2 to 0.3, I managed to clear a few of those errors, but still had some inaccurate images/semantic matches.

The problem seems to stem from the training. The packet file consistently only caught around 500-600 detections. When I messed around with the colab template, I noticed the testing wasn't very accurate there, too. Maybe the reduced batches decreased the effectiveness of the training. I decided not to increase the confidence level any higher, as the example images from the paquet showed that some images were wrong but still showed high confidence! Also, around half of the images had a confidence of 0.4 or lower.

Overall, this was a challenging test for me. I know the program isn't perfect. It has some mistakes and limitations, but I managed to fine-tune it to run with a decent level of accuracy.