

# Report

April 6, 2025

## 1 Homework 0 Report

### 1.1 Code Explanation

The solution loads video frames from a directory, converts them to grayscale, and computes a per-pixel average to obtain a background image that effectively suppresses transient objects like cars. An absolute difference between this background and each frame is computed, with Otsu's method automatically determining a threshold for foreground extraction. The binary masks are then used to detect connected components, with bounding boxes drawn around these regions to highlight the cars. Interactive visualizations were added to facilitate frame-by-frame inspection and analysis. I consulted ChatGPT for help with debugging.

### 1.2 Discussion Questions

1. When you compute the average image (background image), the cars have disappeared – why?
  - The cars disappear in the average image because they are transient objects that only occupy a pixel for a short duration relative to the entire video. When you average many frames, the static background consistently contributes to each pixel, while the moving cars contribute only sporadically. Thus, their effect is diluted, effectively removing them from the computed background.
2. How well does the technique work to separate the cars from the background? Where does it fail and why?
  - The technique works well here since the background remains constant and the cars are moving, since averaging the frames eliminates the transient objects. It would fail if a car stops in the same position and may become part of the background. Also another thing would be variation in lighting since parts of the background can be mistakenly classified as foreground. Another thing would be when cars overlap or are too close, since they would be considered as a single connected component, making it difficult to detect individual cars.