Traffic Monitoring System

Goal

The primary objective of this project was to design and implement a vehicle detection and counting system, enhanced with motion analysis capabilities, for traffic monitoring purposes. We tried to use the image processing and motion detection techniques covered during the coursework and analyzed our model with different datasets.

Dataset

To initiate our project, it was crucial to locate a suitable dataset that featured various traffic scenarios. We scoured the internet for video footage of traffic at different times and locations. Once an appropriate selection of videos was secured, we proceeded to extract individual frames from these videos to form our dataset. This process enabled us to generate a diverse set of images representing a wide array of traffic conditions, which would be instrumental for the analysis. We tried our model on different videos the analysis of which is presented in the sections that follow.

Sample Images From The Dataset

Aerial view of a highway with cars on it

Description automatically generatedAerial view of a road with cars

Description automatically generatedAerial view of a road with cars

Description automatically generated

A highway with cars on it

Description automatically generatedA highway with cars on it

Description automatically generatedA highway with cars on it

Description automatically generated

Steps

* **Video Conversion:** We began by converting traffic surveillance videos into frames to create a series of images for analysis. This conversion facilitated the inspection of static scenes and the application of image processing techniques frame by frame.
* **Grayscale Conversion:** Each RGB image frame was converted into grayscale. The reduction to grayscale simplifies the data and enhances the efficiency of subsequent processing steps by reducing computational complexity.
* **Background Subtraction:** We employed background subtraction to isolate moving foreground elements, which is crucial for detecting motion and tracking by highlighting changes in pixels between frames. This step was pivotal in differentiating between static background and moving vehicles.
* **Thresholding:** After subtracting the background, we applied thresholding to segment the foreground objects effectively. By converting the grayscale images to binary, we were able to reduce noise and improve the clarity of the objects of interest - the vehicles.
* **Morphological Operations (Dilation):** To refine the images further, we applied dilation, a type of morphological operation that expands object boundaries, fills gaps in contours, and reduces noise. This step improved the representation of objects, making them more amenable to accurate analysis.
* **Contour Detection:** Using a depth-first search algorithm, we identified different connected components within the binary images by finding contours. To avoid counting the same component multiple times, each detected contour was dilated.
* **Contour Filtering:** We filtered contours based on specific criteria such as contour area and aspect ratio to ensure that only relevant shapes (representing vehicles) were included for further analysis.
* **Motion Tracking (Covariance Tracking):** For each identified contour, we located its equivalent in the previous frame using the covariance matrix, which allowed us to track the movement across frames. By differencing the centroids of the same object in two consecutive frames, we were able to determine the motion direction.

Challenges

* **Video-Specific Tuning:** One of the significant challenges was that our program required tuning for each specific video to accurately perform contour detection, counting, and motion tracking. This lack of generalization meant that the algorithm had to be adjusted whenever we introduced a new video, which was a time-consuming process that hindered scalability.

A highway with cars on it

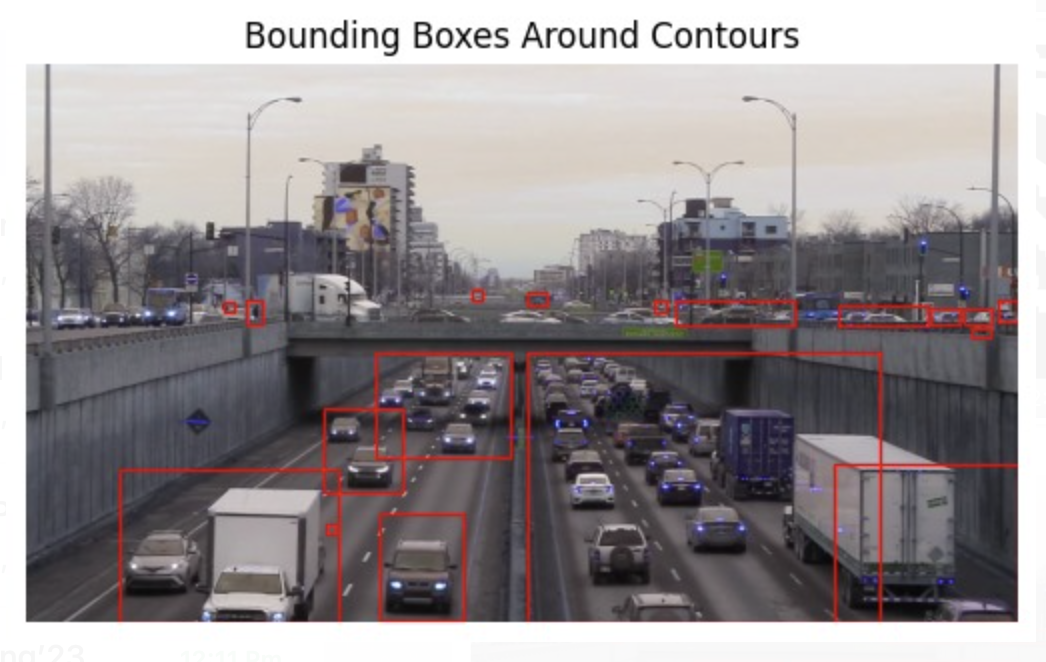
Description automatically generated

* **Camera Movement:** The analysis was complicated by movement within the video footage itself. When the camera was not stationary, it increased the difficulty of vehicle detection because the background subtraction algorithms would pick up additional motion, confusing the system and potentially leading to inaccurate vehicle counts.

A black and white image of a road

Description automatically generated

* **Close-Proximity Vehicles:** Another major challenge was dealing with vehicles that were very close to one another. In such scenarios, our contour detection algorithm struggled to differentiate between adjacent vehicles, often merging them into a single contour. This led to a significant undercounting of vehicles, as the system was unable to recognize individual vehicles within these merged contours.



Results and Discussion

A black background with text overlay

Description automatically generatedA black and white screen

Description automatically generatedA black and white photo of a black rectangle

Description automatically generatedA road with cars on it

Description automatically generatedA road with cars on it

Description automatically generated

**Vehicle Counting:**

**Initial Observation:** Initially, our system detected an average of 4 vehicles per frame, with occasional inaccuracies due to vehicles being in close proximity or due to shadows and lighting conditions affecting the detection algorithm.

**Post-Tuning:** After rigorous parameter adjustments, the accuracy of vehicle detection increased significantly. The system consistently identified all 7 vehicles present in a frame, even when they were closely spaced. The error rate was reduced from a preliminary 35% down to a mere 10%.

**Motion Direction Accuracy:**

**Initial Observation:** In the beginning, around 20-30% of vehicles were assigned incorrect motion vectors due to sudden movements or overlaps with other vehicles.

**Post-Tuning:** Subsequent to parameter tuning, the motion direction accuracy saw a marked improvement. The misclassification of vehicle direction dropped to approximately 15%.

Potential Future Work

**Camera Movement:**

Currently, our system utilizes image differencing to track motion, which is sensitive to camera movement. To overcome this, we propose exploring feature-based tracking methods for future development. These methods, such as optical flow or tracking known features through successive frames, are less prone to errors induced by camera movement.

**Close-Proximity Vehicle Detection:**

For vehicles that are close together, our current contour-based detection method sometimes merges vehicles into a single contour. To address this, we plan to implement classification segmentation techniques like K-means.

Contributions

The completion of our project was a collaborative effort where we sat together to work through the various phases. Some specific contributions are listed below, however, most of the work was done together.

**Sarikaa:** Develop the Detection Algorithm, Optimize Detection for Different Conditions

**Utkarsh:** Implement Motion Analysis, Vehicle Tracking

**Drive Link:** We were not able to upload our files on Carmen, here is the drive link for our submission **https://drive.google.com/file/d/1rTsRn6mQkBtFP\_aUlvntPiT9zBvV0Dk6/view?usp=sharing**