International Institute
Of Information
Technology Hyderabad

Vehicle Detection and Tracking System:

Enhancing Traffic Flow Analysis with Efficient Vehicle Detection

Guide: Dr. Anoop Namboodiri

Team: Visionary Coders

Submitted by

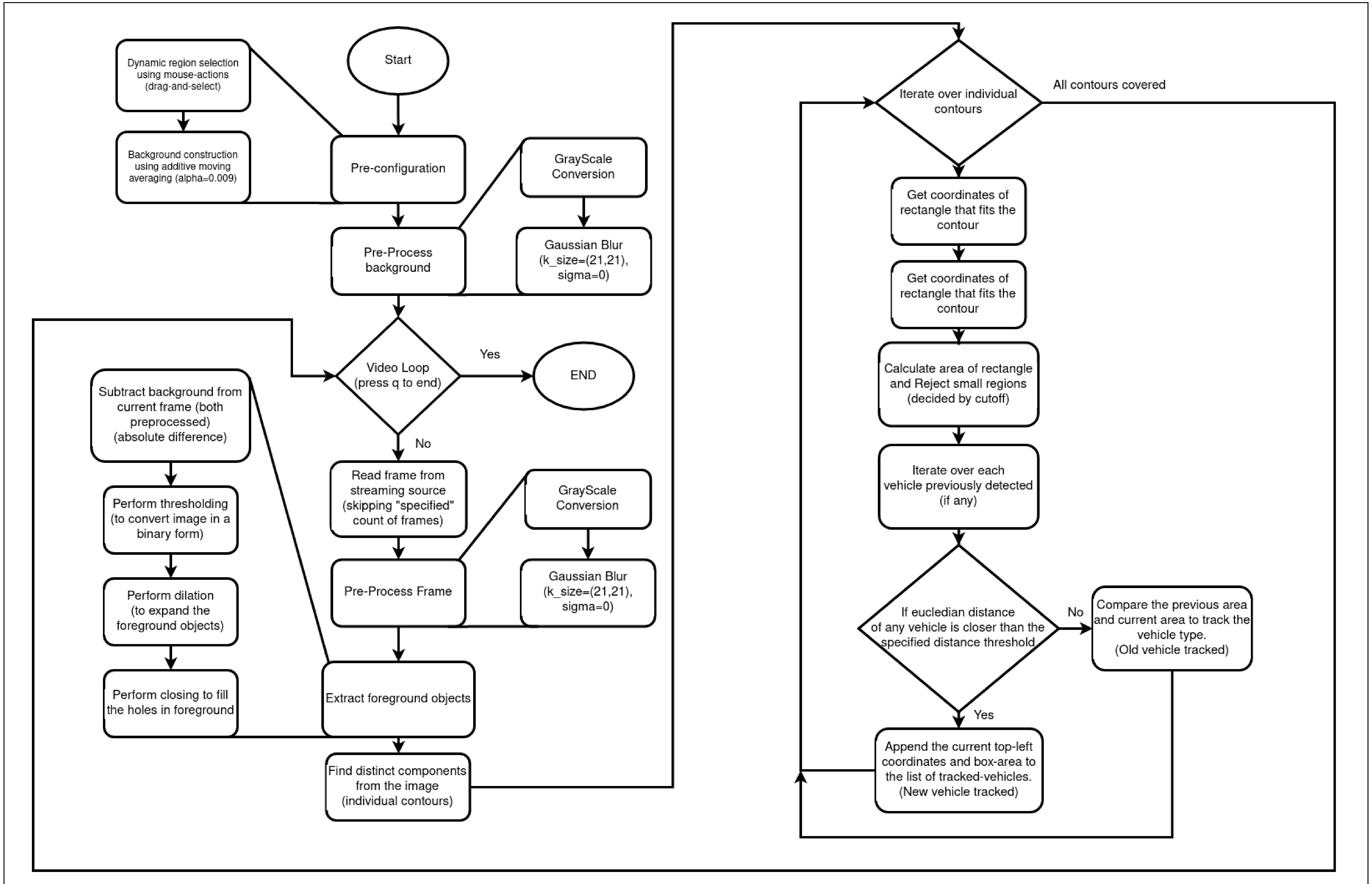
Ashish Chokhani, 2021102016
Ishit Bansal, 2021101083

2nd December, 2024

Introduction

- Vehicle detection plays a key role in traffic management and flow analysis.
- The goal of this project is to improve accuracy in vehicle detection and tracking for real-time traffic monitoring.
- Utilized digital image processing techniques to address issues like vehicle tracking and classification.

Flow Chart



Start
Dynamic region selection using mouse-actions (drag-and-select)
Background construction using additive moving averaging (alpha=0.009)
Pre-configuration
GrayScale Conversion
Gaussian Blur (k_size=(21,21), sigma=0)
Pre-Process background
Video Loop (press q to end)
Yes
END
No
Read frame from streaming source (skipping "specified" count of frames)
GrayScale Conversion
Gaussian Blur (k_size=(21,21), sigma=0)
Pre-Process Frame
Subtract background from current frame (both preprocessed) (absolute difference)
Perform thresholding (to convert image in a binary form)
Perform dilation (to expand the foreground objects)
Perform closing to fill the holes in foreground
Extract foreground objects
Find distinct components from the image (individual contours)
Iterate over individual contours
All contours covered
Get coordinates of rectangle that fits the contour
Get coordinates of rectangle that fits the contour
Calculate area of rectangle and Reject small regions (decided by cutoff)
Iterate over each vehicle previously detected (if any)
If eucledian distance of any vehicle is closer than the specified distance threshold
No
Compare the previous area and current area to track the vehicle type. (Old vehicle tracked)
Yes
Append the current top-left coordinates and box-area to the list of tracked-vehicles. (New vehicle tracked)

Workflow

Image Processing

- Conversion of frames to grayscale for simplified analysis.
- Noise reduction using Gaussian blur (kernel size = 21x21).
- Binary thresholding to detect object edges and shapes.

Motion Analysis

- Background construction using a moving average method (alpha = 0.009).
- Subtraction of current frame from background to extract motion (absolute difference).
- Foreground objects are enhanced using dilation and closing operations.
- Contour extraction for vehicle detection.

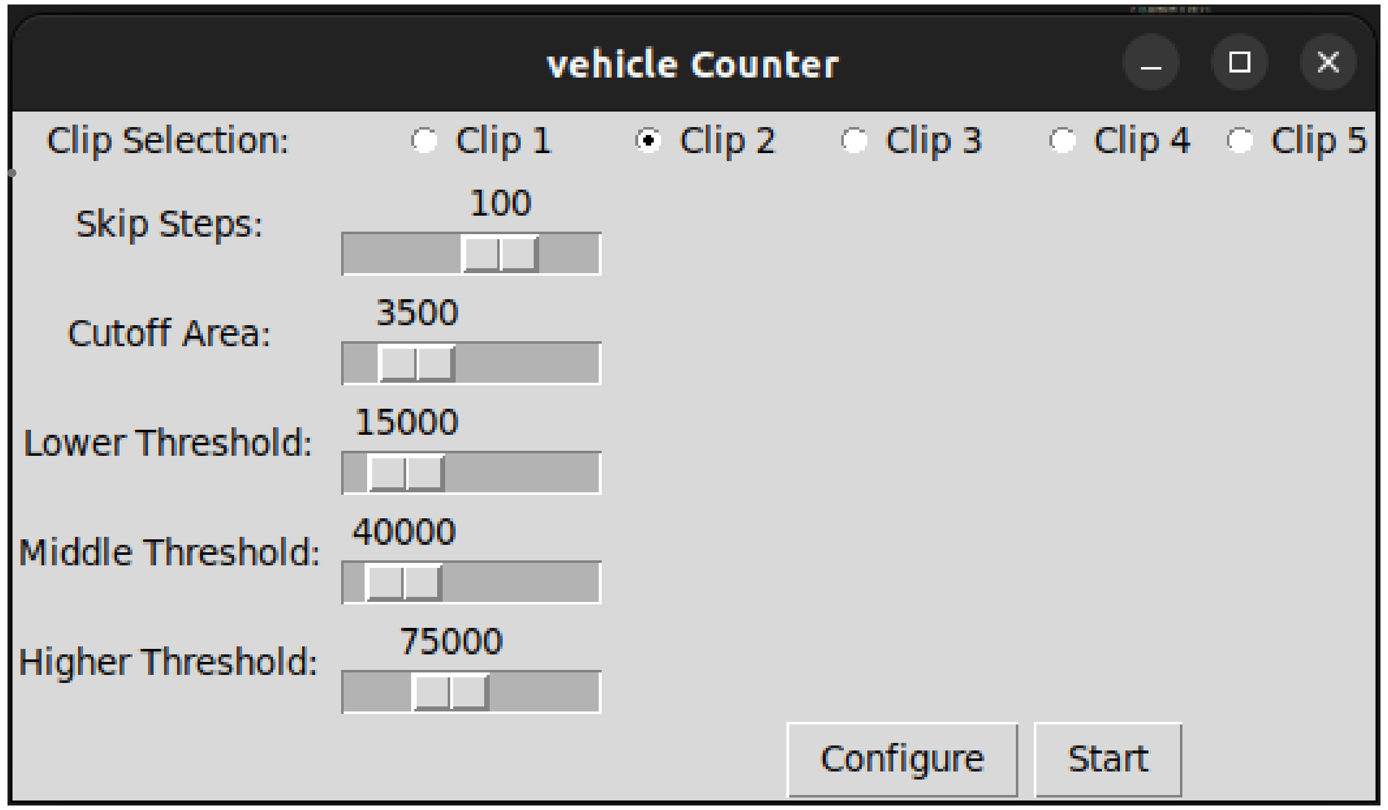
Vehicle Tracking and Contour Analysis

- Bounding boxes are drawn around detected vehicles using contours.
- Small regions (noise) are filtered out based on the area cutoff threshold.
- Tracking vehicles using their bounding box areas and positions.
- Vehicles are tracked using Euclidean distance to prevent misidentification.

GUI

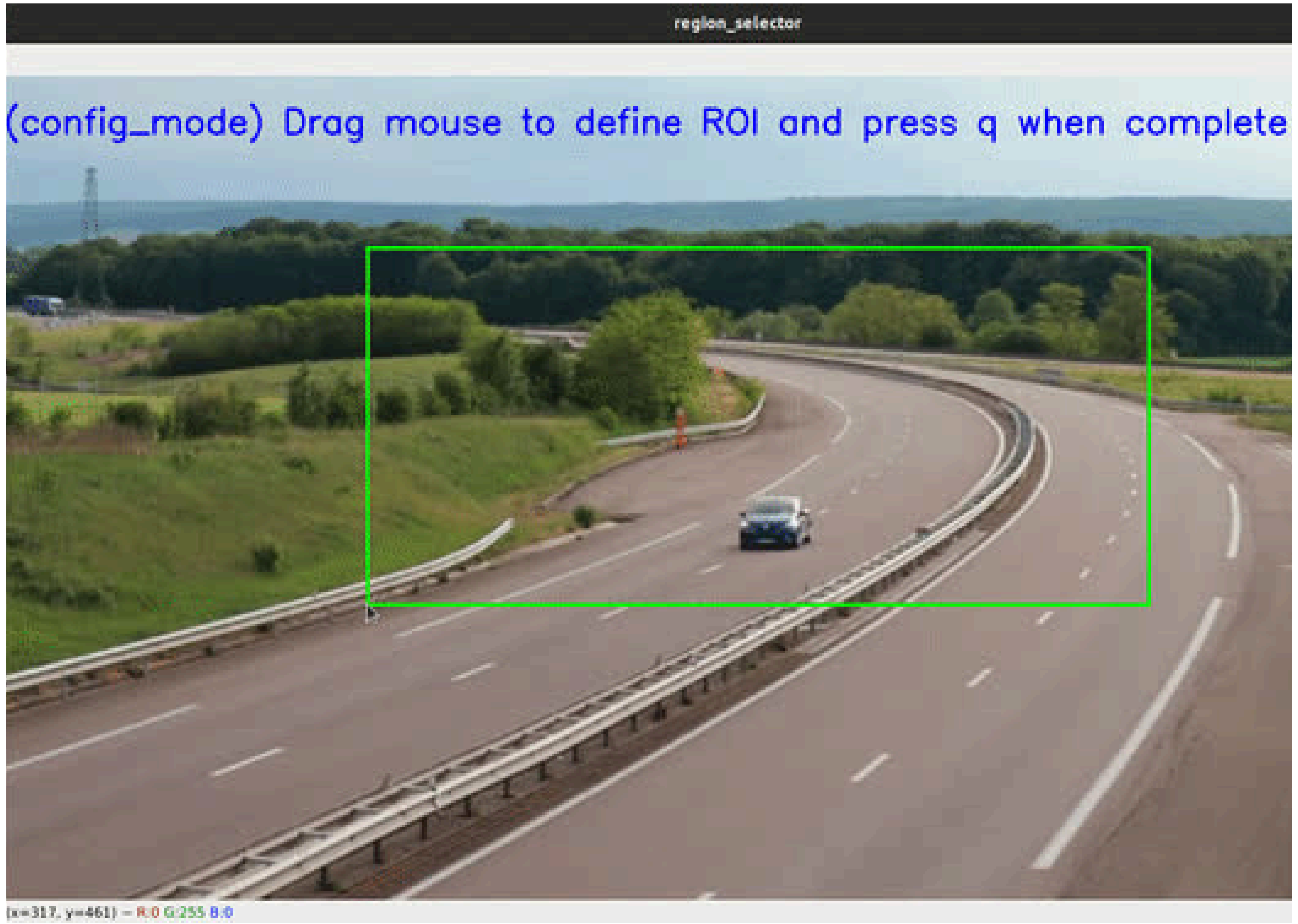
This GUI for the "Vehicle Counter" allows users to configure and start vehicle detection. Key features include:

- **Clip Selection:** Choose from five video clips.
- **Skip Steps**: Adjust the number of frames skipped during processing.
- **Cutoff Area**: Set the minimum object size to detect.
- **Thresholds:** Define size ranges for small, medium, and large vehicles.
- **Configure button:** Select the region of interest
- **Start button:** Start vehicle detection using the chosen parameters



Region Selection

- **Dynamic ROI:** Manually select ROI during configuration.
- **Relevance:** Focus detection on key areas like traffic lanes.
- **Interactive:** Adjust ROI based on the video scene.
- **Flexibility:** Adapt ROI for different traffic environments.
- **Impact:** Improves detection accuracy by focusing on relevant areas.



Background Creation

- **Moving Average Method:** The background is built by gradually incorporating new frames into the existing model.
- **Weighting:** Recent frames are given more weight, while older frames are less influential.
- **Cumulative Process:** Frames are added over time to update the background model.
- **Smoothing:** After enough frames, techniques like Gaussian blur are applied to smooth the background.

Issues from Proposed method

Sobel Edge Detection

Issues:

- Excessive noise in the Sobel edge detection output.
- Unclear vehicle boundaries due to inconsistent lighting or high background texture.
- This creates difficulty in isolating vehicles for accurate detection and classification.

How we addressed it:

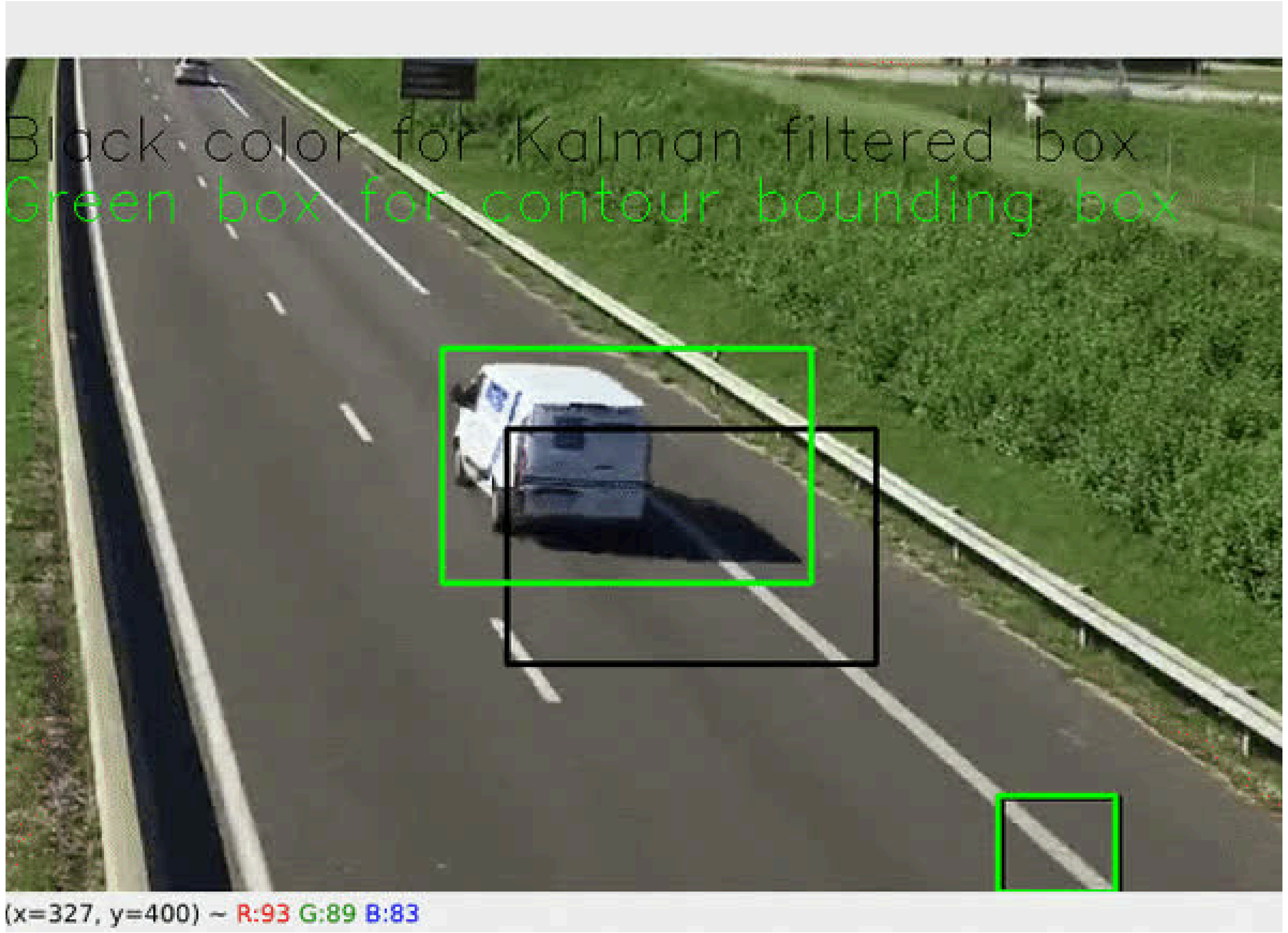
- Applied Gaussian smoothing on grayscale frames to reduce noise.
- This helped suppress background noise and clarified vehicle edges, improving detection accuracy.

Issues from Proposed method

Kalman Filter

Issues:

- Kalman filters struggle with unpredictable vehicle movements and occlusions.
- Tracking becomes inaccurate as Kalman filters fail to predict bounding box positions over time.
- Multiple Kalman filters are needed, increasing computational complexity.



How we addressed it:

- Replaced Kalman filters with a **Nearest Neighbor trick:**
- Track vehicles by connecting bounding boxes between consecutive frames.
- **Exploit the continuity of moving vehicles:** the bounding box at time-step t+1 must move by a small distance from time-step t.

Classification Criteria

On the basis of vehicle size, we classify them into 4 types:
(There thresholds can be modified in the GUI)

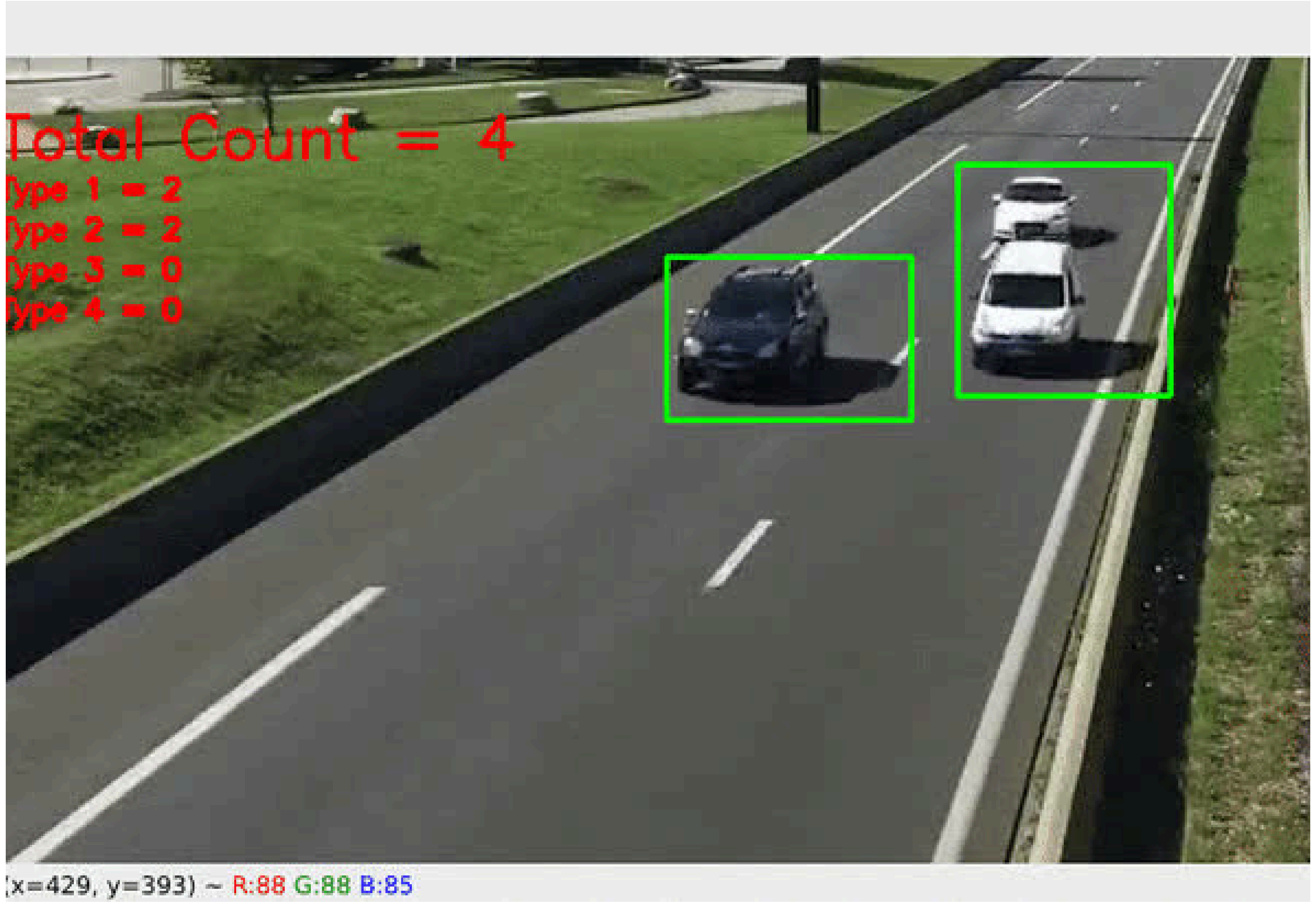
- **Type 1** (Bicycle, MotorCycle)
- **Type 2** (Car)
- **Type 3** (Pickup, Minibus)
- **Type 4** (Buses, Trucks, Trailers)

Extreme case: For a partially captured vehicle, initially it maybe mis-classified but it is handled later as we consider the largest area till the vehicle is in the frame.

Counting

- Ground Truth values for each group of vehicles are “manually” counted from a given video.

- Further, when an untracked vehicle enters the “detection zone” it is added to the total count as well as classified into one of the mentioned groups by updating the count of that particular category.



Results

Confusion Matrix

Prediction

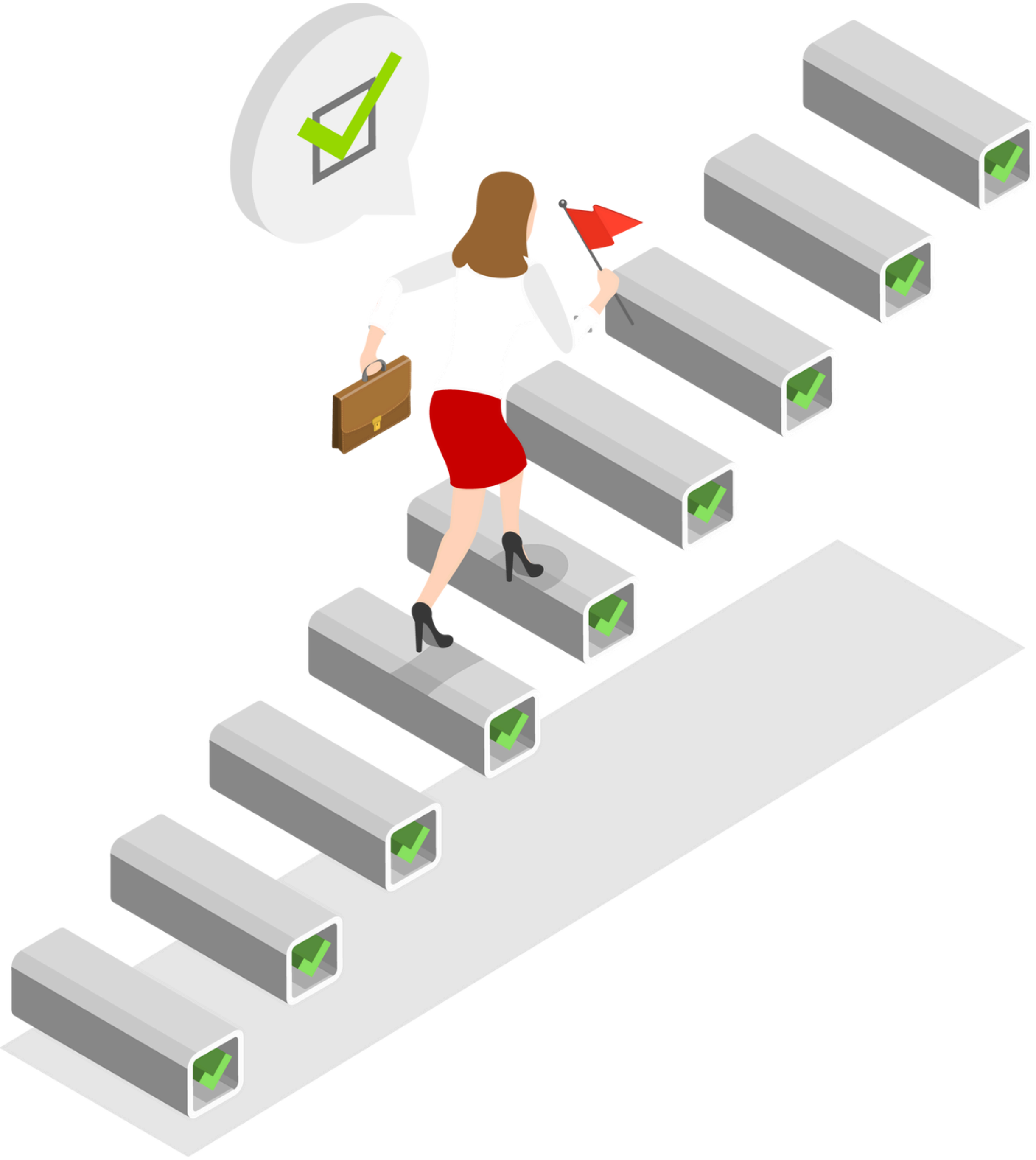
Ground Truth

	Type 1	Type 2	Type 3	Type 4
Type 1	2	1	0	0
Type 2	1	25	0	0
Type 3	0	0	10	0
Type 4	0	0	0	3

Accuracy :

$$\left(\frac{40}{42}\right) \times 100 = 95.23\%$$

Challenges faced



- Difficulty in tracking vehicles moving across lanes
- Shadows and lighting conditions affecting background subtraction accuracy.
- Misclassification of closely moving vehicles.
- Processing lag with high frame rates

Conclusion

- This enhanced vehicle detection system is efficient and provides accurate tracking.
- Suitable for real-time traffic monitoring with improvements in classification accuracy.
- Offers a scalable solution for traffic analysis at low cost

Questions?

References And Literature Survey

01 Tourani, A., & Shahbahrami, A. (2015). Vehicle counting method based on digital image processing algorithms. 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)

02 Pancharatnam, M., & Sonnadara, U. (2008). Vehicle Counting and Classification from a Traffic Scene.

03 Jamiya S, S., & P., Esther. (2019). A Survey On Vehicle Detection And Tracking Algorithms In Real Time Video Surveillance.

04 Data: Video Clips Used-

- Video 1. Video 2. Video 3

THANKS

FOR YOUR ATTENTION

Submitted by
Ashish Chokhani, 2021102016
Ishit Bansal, 2021101083

2nd December 2024