CSE 4127: Image Processing and Computer Vision Laboratory

Vehicle Detection and Counting Using Median Frame Analysis in Video Streams

By

Md. Nasim Ridoy

Roll: 1907104

Submitted To:

Dr. Sk. Md. Masudul Ahsan

Professor

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Dipannita Biswas

Lecturer

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

July, 2024

Objectives

The primary objectives of this project are:

- To develop a robust system that can accurately detect vehicles in a video stream using advanced image processing techniques.
- To reliably count the number of vehicles passing through a predefined region of interest in the video frame.
- To implement efficient background subtraction and frame differencing techniques for identifying moving objects in the video.
- To apply morphological operations to remove noise and refine the detection of vehicles.
- To allow users to interactively select regions of interest in the video frame for vehicle counting.
- To visually display detected vehicles, bounding boxes, and vehicle counts on the video frames.
- To design the system in a way that it can be easily scaled and adapted to different camera setups and traffic scenarios.

Introduction

Vehicle detection and counting is a critical task in modern traffic management and urban planning. With the continuous growth in vehicle numbers, cities face significant challenges related to traffic congestion, road safety, and pollution. Accurate and efficient methods for monitoring traffic flow are essential for mitigating these issues and making informed decisions about infrastructure development.

Traditional approaches to vehicle detection and counting often involve physical sensors like inductive loop detectors, infrared sensors, and radar systems. While these methods can be effective, they come with several drawbacks. Installing and maintaining physical sensors can be costly and labor-intensive. Additionally, these systems are often limited to specific points on the road and may not provide comprehensive data across broader areas.

In contrast, video-based vehicle detection and counting offer a more flexible and scalable solution. By using video data from traffic cameras, it is possible to monitor traffic flow across extensive road networks without the need for extensive physical infrastructure. Recent advancements in image processing and computer vision have made it feasible to develop systems that can automatically detect and count vehicles with high accuracy.

The implementation of this project involves several steps, including selecting key frames from the video, calculating a median frame for background subtraction, detecting moving objects, and applying morphological operations to refine the detection results. Additionally,

a user interface is provided to allow the selection of regions of interest, enhancing the system's flexibility and usability.

Theory

The foundation of this project is based on several key concepts in image processing and computer vision:

Background Subtraction

This technique involves creating a model of the scene's background and identifying moving objects by comparing each frame to this background. Any significant changes are considered to be moving objects.

Frame Differencing

This method calculates the difference between consecutive frames to detect motion. Significant differences indicate the presence of moving objects.

Median Frame Calculation

By calculating the median of a series of frames, a robust background model can be created. This model is less sensitive to changes in lighting and other transient variations.

Contour Detection

This technique involves identifying the boundaries of objects in a binary image. Contours are useful for extracting shape and positional information about detected objects.

Morphological Operations

These operations, such as dilation and erosion, are used to refine binary images by removing noise and closing gaps in detected objects.

Object Tracking

By identifying and tracking the centroids of detected objects across frames, it is possible to count the number of objects passing through a specified region.

Methodology

The development of the vehicle detection and counting system involves several key steps, each contributing to the overall functionality and accuracy of the system. The following methodology outlines these steps in detail:

1. Video Input

The process begins with loading the video file that contains the traffic footage. This video serves as the primary input for the system. The video is read using OpenCV's 'VideoCapture' class, which allows for frame-by-frame processing.

Figure 1: A frame of Input Footage

2. Frame Selection for Background Modeling

To effectively detect moving vehicles, a background model is needed to distinguish between static and dynamic elements in the video. This is achieved by selecting a set of key frames from the video, which are used to calculate a median frame representing the background.

Steps:

- **Total Frame Count**: Determine the total number of frames in the video using `cap.get(cv2.CAP_PROP_FRAME_COUNT)`.
- **Random Frame Selection**: Randomly select a set of frames (e.g., 50 frames) from the video to capture a variety of background scenes. This helps in creating a robust background model that can handle variations in lighting and transient objects.
- **Frame Validation**: Ensure that each selected frame is valid by checking if it can be read successfully.

3. Median Frame Calculation

Once the key frames are selected, a median frame is calculated. This frame serves as the background model, representing the static elements of the scene.

Steps:

- **Frame Array Creation**: Store the selected frames in an array.
- **Median Calculation**: Compute the median of the frames along the time axis using `np.median(frames_array, axis=0)`. This median frame effectively reduces the impact of moving objects and transient changes in the scene.

Figure 2: Median Frame

4. Selection of two points

To count vehicles, a specific region of interest (ROI) within the video frame is defined. This is done interactively, allowing the user to select two points on the median frame to define the ROI.

Steps:

- **Display Median Frame:** Show the median frame to the user using cv2.imshow().
- **Mouse Callback Function:** Implement a mouse callback function that captures the coordinates of the points clicked by the user. This function also displays the selected points on the median frame.

- **Collect Points:** Ensure exactly two points are collected to define the ROI.

Figure 3: Median frame with two points selected

5. Frame Processing

Each frame of the video is processed to detect moving vehicles. This involves several sub-steps:

5.1 Grayscale Conversion

Convert each frame to grayscale to simplify the processing and reduce computational complexity.

5.2 Absolute Difference Calculation

Calculate the absolute difference between the current grayscale frame and the median frame. This highlights the moving objects by subtracting the static background.

Figure 4: Absolute difference of a frame

5.3 Thresholding

Apply a binary threshold to the absolute difference frame to create a binary image where moving objects are highlighted. The threshold value is 30 here.

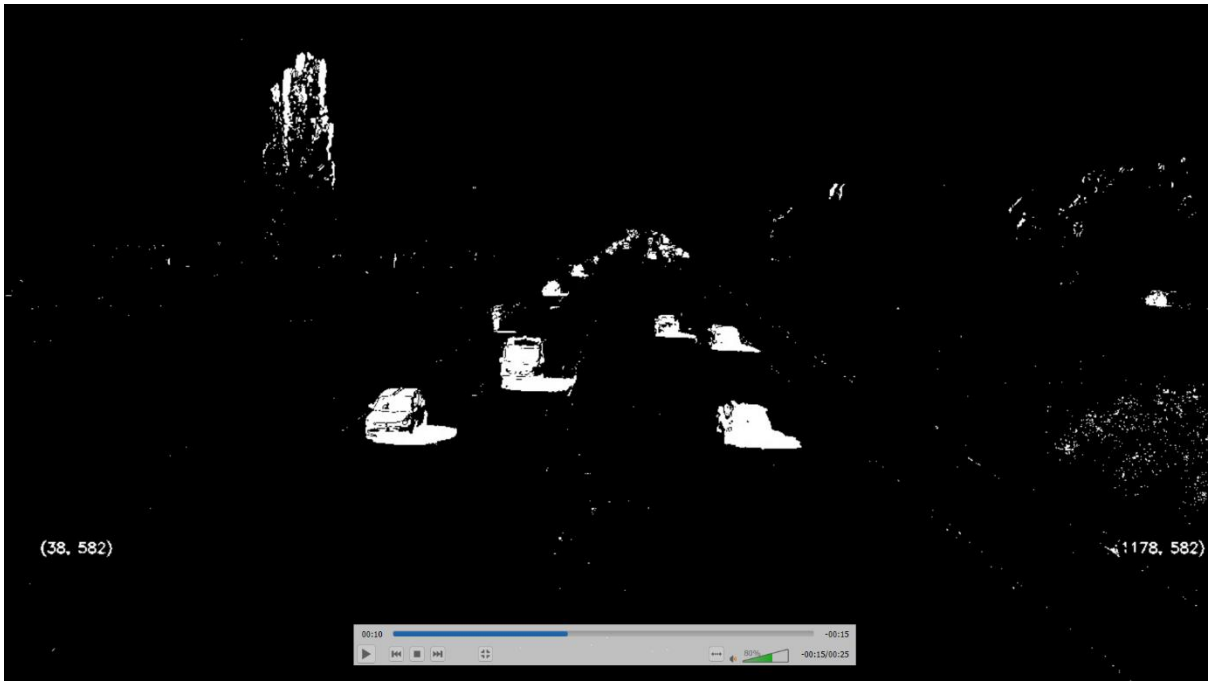

Figure 5: binary image after thresholding

5.4 Morphological Operations

Use morphological operations to remove noise and refine the detection of moving objects. This includes operations such as opening (erosion followed by dilation) and dilation. After that further dilation was done for more connectivity of the vehicle as a whole for better contour finding.

Figure 6:After Morphological Open

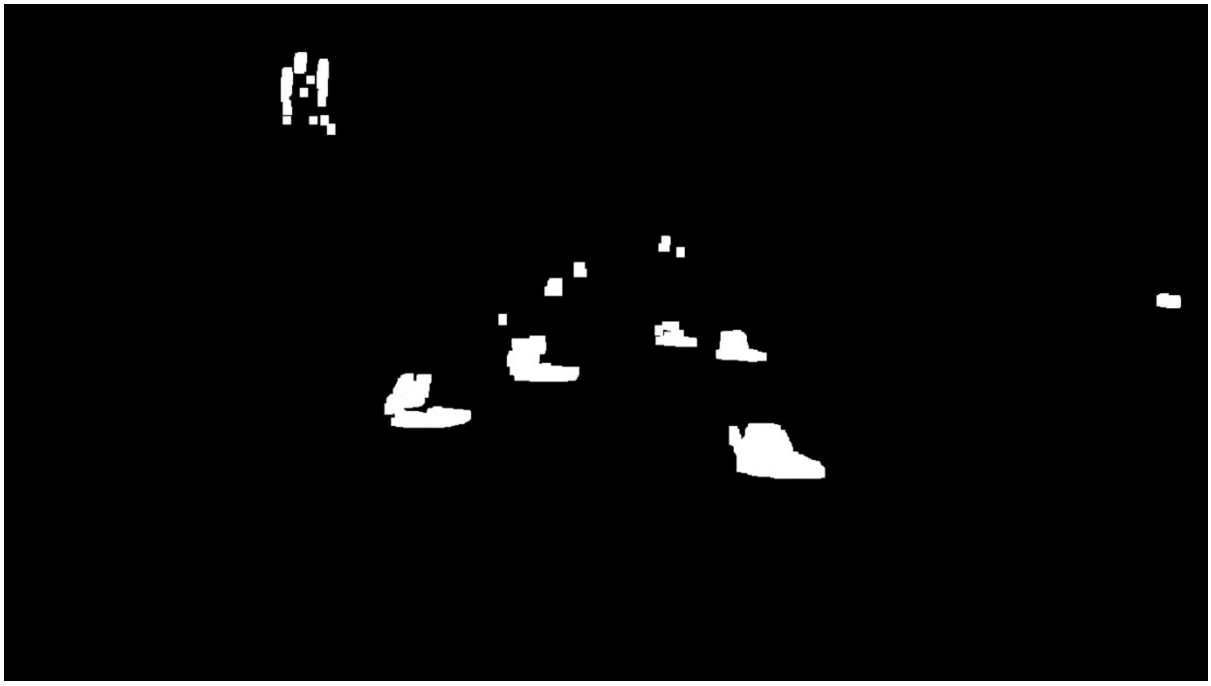

Figure 7: After Dilation on previous frame

5.5 Object Detection and Tracking

Detect contours in the binary image to identify moving objects and track their positions.

Steps:

- **Contour Detection:** Use ***cv2.findContours()*** to find contours in the binary image.
- **Bounding Box Calculation:** For each contour, calculate the bounding box and draw it on the original frame.
- **Center Calculation:** Compute the centroid of each bounding box to track the position of the detected objects.

5.6 Vehicle Counting

Count the number of vehicles passing through the defined ROI by tracking the detected objects' centroids and checking if they intersect with the line defined by the two points selected earlier.

Steps:

- **Line Equation Calculation:** Calculate the coefficients of the line equation defined by the two selected points. The line equation can be represented as:

$$\mathbf{Ax + By + C = 0}$$

where A = y2-y1, B = x1-x2 and C = x2y1-x1y2

- **Distance Calculation:** For each detected object, calculate the perpendicular distance from its centroid to the line using the formula:

$$\mathbf{distance} = \frac{|\boldsymbol{Ax + By + C}|}{\sqrt{(\boldsymbol{A^2} + \boldsymbol{B^2})}}$$

- **Vehicle Count Update:** If the distance is within a specified threshold (offset), increment the vehicle count.

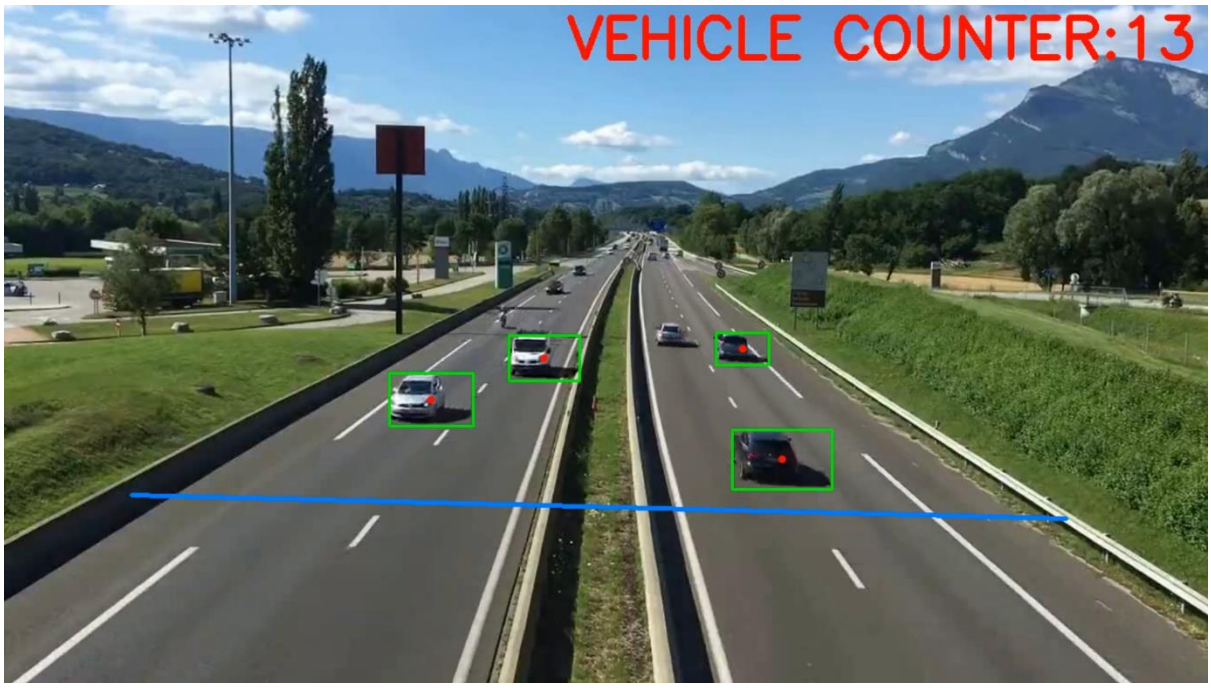

Figure 8: final output frame

Pseudocode

 # Video Input
open video file
initialize frame index to 0

 # Frame Selection for Background Modeling
total_frames = get total number of frames in video
initialize empty list frame_ids
while length of frame_ids < 50:
    randomly select frame_id
    if frame is valid:
        add frame_id to frame_ids

 # Median Frame Calculation
initialize empty list frames
for each frame_id in frame_ids:
    set video to frame_id
    read frame and add to frames
calculate median_frame from frames

 # Region of Interest (ROI) Selection
display median_frame to user
define mouse callback function to capture coordinates of points clicked
wait for user to select two points
if less than two points selected:
    print error and exit

 # Frame Processing
set video to first frame
convert median_frame to grayscale
initialize video writers for different stages of processing
initialize progress bar

while frames are available:
    read current frame
    convert current frame to grayscale
    calculate absolute difference between current frame and median frame
    threshold absolute difference to create binary image
    apply morphological operations to remove noise
    dilate binary image

    # Object Detection and Tracking
    find contours in binary image
    for each contour:
        calculate bounding box
        if bounding box meets size criteria:
            draw bounding box on frame
            calculate center of bounding box
            add center to list of detected points

    # Vehicle Counting
    for each detected point:
        if point is within ROI and close to the line:
            increment vehicle counter
            remove point from detected points

    # Visualization and Output
    overlay vehicle count on frame
    draw ROI line on frame
    display frame
    write frame to output video

release video resources
close all OpenCV windows

Result:

Example 1:

Figure 9: Input frame

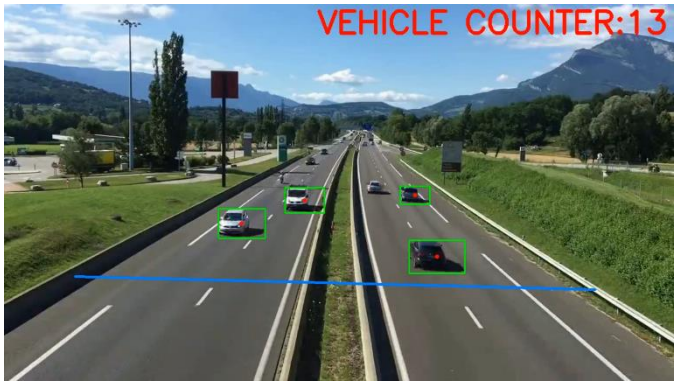

Figure 10: output frame

Example 2:

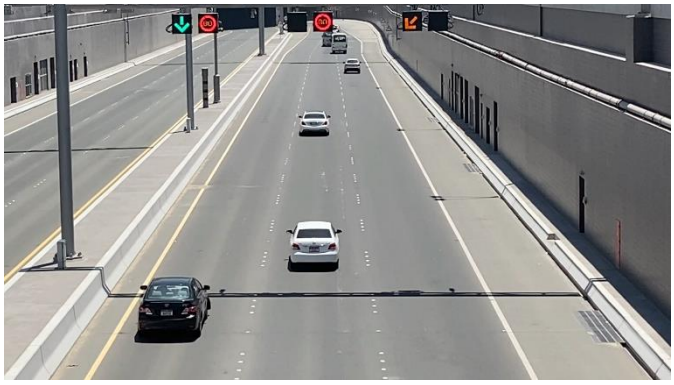

Figure 12: input frame

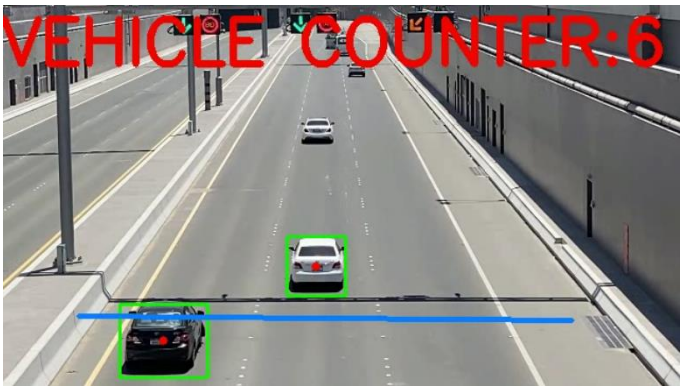

Figure 11: output frame

Discussion

The system designed for vehicle detection and counting leverages fundamental image processing and computer vision techniques. By using a median frame as the background model, the system effectively distinguishes between moving vehicles and static background elements. The random frame selection for background modeling ensures robustness against transient changes and varying lighting conditions.

The methodology of converting frames to grayscale, calculating absolute differences, and applying binary thresholding provides a reliable mechanism to detect motion. Morphological operations further enhance the detection accuracy by removing noise and refining object shapes. The system efficiently processes each frame in real-time, making it suitable for practical applications such as traffic monitoring and management.

Limitations:
Lighting Conditions: Variations in lighting, such as shadows or reflections, can affect the accuracy of vehicle detection. While the median frame helps mitigate some of these issues, sudden lighting changes can still pose challenges.

Frame Selection: The random selection of frames for background modeling assumes that the selected frames will adequately represent the background. In scenarios with significant background changes, this assumption may not hold, leading to inaccuracies.

Occlusions: Vehicles that are partially or fully occluded by other objects may not be accurately detected or counted. This is a common challenge in crowded traffic scenes.

Conclusion

The project successfully demonstrates a system for vehicle detection and counting using basic image processing and computer vision techniques. The methodology of background modeling, motion detection, and object tracking provides a solid foundation for traffic monitoring applications. Despite some challenges related to lighting conditions and occlusions, the system performs reliably under various scenarios.

By addressing the identified challenges and exploring potential improvements, the system can be further enhanced to provide more accurate and robust vehicle detection and counting capabilities. This project serves as a valuable exercise in applying theoretical concepts to practical problems, highlighting the potential of computer vision technologies in real-world applications.

Overall, the system achieves its objectives, demonstrating the feasibility and effectiveness of using image processing techniques for vehicle detection and counting in traffic monitoring systems.

References

- OpenCV Library Documentation. Available at: https://docs.opencv.org/
- Bradski, G., Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer Science & Business Media
- Research papers
- Youtube videos