*VEHICLE COUNTING SYSTEM*

Muthu Sanjai A.S   
*CSE IoT*   
 *Shiv Nadar University*Chennai, India

Saeshwaran  
*CSE IoT*  
*Shiv Nadar University*Chennai, India

Darshini P  
*CSE IoT*   
 *Shiv Nadar University*Chennai, India

Loka Nehan Reddy  
*CSE IoT*   
*Shiv Nadar University*Chennai, India

Praneetha  
*CSE IoT*   
*Shiv Nadar University*Chennai, India  
Jaideep D  
*CSE IoT*  
 *Shiv Nadar University*Chennai, India

M S Shruti  
*CSE IoT*   
*Shiv Nadar University*Chennai, India

Sathvik  
*CSE IoT*   
*Shiv Nadar University*Chennai, India

***Abstract*—** **Efficiently monitoring and managing traffic flow has become more crucial in modern urban areas. Exact, up-to-date data on vehicle counts are necessary for transportation agencies, traffic engineers, and urban planners to create better-informed judgments about congestion reduction, infrastructure development, and traffic optimization. This work presents an all-encompassing analysis and evaluation of systems that count vehicles, scrutinizing their underlying principles, technologies, challenges, and prospective uses.**

***Keywords—Python,OpenCV library,object detection***

# introduction

Traffic congestion has become a pressing issue in urban areas worldwide, leading to increased travel times, fuel consumption, and environmental pollution. Effective traffic management necessitates a deep understanding of traffic patterns and the ability to analyze and predict traffic flow. Vehicle counting systems play a crucial role in providing valuable data for such analyses, aiding in the development of intelligent transportation systems and the optimization of traffic operations.

Over the years, numerous vehicle counting methods and technologies have emerged, each with its advantages, limitations, and suitability for specific scenarios. This paper aims to provide a comprehensive overview of these systems, highlighting their underlying principles and technologies. By exploring the existing literature and examining real-world implementations, this study aims to identify the strengths and weaknesses of various vehicle counter systems and shed light on their potential applications.

The rest of this paper is organized as follows: Section II provides an objective of the fundamental principles underlying vehicle counting, covering both traditional and emerging methods. Section III tells about the Literature survey . Section IV presents a detailed analysis of this system,in detail . Section V discusses the design system of the project and explains the project with the help of a structural representation . Finally, Section VI is about implementation of project and output . Finally section VII concludes the paper and outlines future directions for research and development in this field.

By critically examining the existing literature and analyzing real-world case studies, this paper intends to provide valuable insights into the advancements made in vehicle counting systems. Moreover, it aims to assist researchers, practitioners, and policymakers in selecting appropriate technologies and methodologies for their specific traffic monitoring needs. Ultimately, the knowledge gained from this comprehensive review will contribute to the development of more efficient and intelligent traffic management systems, leading to improved mobility, reduced congestion, and enhanced sustainability in urban environments.

1. OBJECTIVES

The objectives of vehicle counting project using Python can vary depending on the specific application and requirements.

* + 1. To provide a comprehensive review of vehicle counting systems and methodologies, including both traditional and emerging techniques.
    2. To analyze and compare the underlying principles and technologies employed in vehicle counting systems, such as image processing, radar, magnetic sensors, and acoustic devices.
    3. To assess the accuracy, reliability, and scalability of different vehicle counting systems under diverse traffic conditions and environments.
    4. To identify the strengths and limitations of various vehicle counting methods and technologies, helping researchers and practitioners make informed decisions regarding system selection and implementation.

1. LITERATURE SURVEY
2. "Real-Time Vehicle Counting and Classification for Traffic Surveillance" by Liu et al. (2018)

This study proposes a real-time vehicle counting and classification system based on computer vision techniques. The authors utilize a combination of background subtraction, blob analysis, and support vector machines to accurately count and classify vehicles in various traffic scenarios.

1. "Traffic Monitoring System Based on Connected Vehicle Technology" by Zhang et al. (2019)

The authors present a traffic monitoring system that leverages connected vehicle technology to collect real-time data from vehicles on the road. The system utilizes vehicle-to-infrastructure communication to gather information such as speed, location, and traffic conditions, enabling accurate traffic monitoring and congestion detection.

1. "Efficient Vehicle Detection and Tracking for Traffic Surveillance Systems" by Chen et al. (2020)

In this work, an efficient vehicle detection and tracking approach is proposed for traffic surveillance systems. The authors combine a convolutional neural network (CNN) with an online tracking algorithm to detect and track vehicles in real-time. The system achieves high accuracy and robustness, even in challenging traffic conditions.

1. "A Review of Traffic Monitoring Systems: Techniques and Applications" by Khan et al. (2017)

This review paper provides an overview of various traffic monitoring systems and their techniques and applications. The authors discuss different approaches, including sensor-based systems, computer vision methods, and wireless communication-based systems, highlighting their strengths, limitations, and potential applications.

1. "Automatic Vehicle Counting and Classification Using Deep Learning Techniques" by Wang et al. (2019)

The authors propose an automatic vehicle counting and classification system based on deep learning techniques. They use a deep convolutional neural network (CNN) to detect and classify vehicles from video streams. The system achieves high accuracy in real-time vehicle counting and classification tasks.

1. "Traffic Flow Monitoring and Prediction Using Machine Learning Techniques" by Li et al. (2021)

This study focuses on traffic flow monitoring and prediction using machine learning techniques. The authors analyze historical traffic data and develop a predictive model based on recurrent neural networks (RNNs) to estimate future traffic conditions accurately. The proposed system can assist in effective traffic management and congestion avoidance.

1. "Vision-Based Vehicle Detection and Tracking for Traffic Surveillance: A Comprehensive Review" by Gupta et al. (2020)

This comprehensive review paper discusses vision-based vehicle detection and tracking techniques for traffic surveillance applications. The authors survey various computer vision algorithms, including feature-based methods, deep learning approaches, and multi-camera systems, providing insights into their strengths, limitations, and comparative performance.

1. "Intelligent Transportation Systems for Traffic Management: A Survey" by Zhang et al. (2018)

The authors present a survey of intelligent transportation systems (ITS) for traffic management. The paper covers a wide range of techniques and technologies, including vehicle detection, tracking, and classification, as well as traffic flow analysis and congestion control. The survey provides a comprehensive overview of state-of-the-art ITS solutions.

IV. THEORY

A. Background Subtraction

One method for detecting a moving object is background

subtraction. Background subtraction, also known as

foreground detection, is one of the techniques in the field of

image processing and computer vision that aims to detect

foreground (objects) from the background for further

processing . Generally, objects detected are human objects,

cars, text, fire, etc. Background subtraction is a method

commonly used to detect moving objects on videos from static cameras. The process of detecting a moving object with the background subtraction method is based on the difference between the reference background and the frame. In traffic monitoring research this method has been carried out and several studies were conducted.

B. Morphology

Morphology is an image processing technique based on an

image segment that aims to improve the results of

segmentation. Morphological techniques are usually used in

binary imagery or in some cases can also be applied to gray

images or images with intensity values. Dilation and

erosion are the most widely used morphological techniques.

Dilation is a technique for thickening object segments by

adding layers around the object . At this time the dilation

technique is not only used in image segmentation but in the

current research, dilation techniques are also used to increase payload in steganography techniques. Whereas erosion is the opposite technique of dilation, a technique that aims to erode the edges of the object .

C. Contour

Edge detector produces an edge image in the form of a

binary image, where edge pixels are represented in white and

besides edges are represented in black. But the edge area of

the image is not necessarily able to provide useful

information, because there is not necessarily a link between an edge and the other edge. This edge image must be processed further to produce more useful information that can be used further to detect objects or basic forms such as lines, circles, ellipses, etc. in the image analysis process .The series of edge pixels that form the boundaries of a region is called contours, where there are two types of contours which are open and closed. Closed contours correspond to the boundary that surrounds an area. Regional boundaries are useful for describing the shape of an object in the image analysis stage, for example, to recognize objects. Open contours can be in the form of lines or regions that do not form circuits. The use of contour analysis in detecting objects is the most important part in detecting many objects in an image.

V. DESIGN SYSTEM

We have used two libraries, Numpy and open CV for our model. The module imported from Open CV lets us run and analyse components of a video. Along with the computational efficiency provided by Numpy, the mentioned

libraries help us analyse traffic videos in detail.

We have used features such as grey scaling, background subtraction and dilation to detect a moving body and have created an imaginary boundary which a vehicle must cross to get identified.

This system is built in a web-based application with the

Python 2.7 programming language written with the OpenCV

editor. While for the storage of vehicle volume data will be

stored in the MySQL database and to display and process

videos sourced from CCTV on a web application in real-time

added microframework Flask. In more detail, the process of

video extraction and extraction to obtain and calculate vehicle

objects is used background subtraction method, determination of binary threshold values, morphological processes with erosion and dilation, detection of objects with counters, and determining whether the object is a vehicle or not by tracking based on trajectory passed .

Our sample video demonstrates a two lane road in which vehicles passes by . CCTV captures the vehicles passing through it . Our project helps with identifying vehicles and counting them one by one . When a vehicle passes by the number displayed by the counter keeps on updating then and there . We have attached a structural representation of our project through a flowchart for better understanding of our project .

STRUCTURAL REPRESENTATION

TAKE THE VIDEO AS INPUT USING OPENCV

DO THE BACKGROUND SUBRACTION PROCESS TO SEPARATE OBJECTS FROM THE BACKGROUND

DO DILATION ON THE OBJECT TO REINFORCE THE SHAPE OF THE OBJECT

CREATE RECTANGULAR BOUNDARIES AROUND THE VEHICLES

CREATE A LINE WHICH PASSES THE COUNT OF VEHICLE WHEN A VEHICLE PASSES THROUGH THE LINE

TRACKING THE OBJECT

OUTPUT :

VEHICLE COUNTER : 2(Eg)

VI. IMPLEMENTATION

OpenCV (Open Source Computer Vision Library): OpenCV is a versatile and widely adopted library for computer vision tasks, including image recognition. It provides a comprehensive collection of functions and algorithms for image processing, feature detection, object recognition, and more. OpenCV is known for its efficiency, cross-platform compatibility, and extensive community support. It has bindings for Python, making it a popular choice for image recognition projects.

The cv2.findContours () ' function returns a list of contours in the image. Each contour is

represented as a list of points, and each point is represented as a tuple of (x, y)

coordinates.

The enumerate () ' function is used to loop through the contours and keep track of their index numbers. For each contour, the cv2. boundingRect () ' function is used to calculate the coordinates of the bounding rectangle.

The validate\_counter' variable is used to check if the width and height of the bounding rectangle are greater than or equal to the minimum width and height values set by

'min\_width \_rect' and 'min\_height \_rect. If the values do not meet the criteria, the loop

continues to the next contour.

If the 'validate\_counter' condition is met, the cv2. rectangle () ' function is used to da

the bounding box around the contour in the original image ('frame"). The rectanale is

drawn using the top-left and bottom-right coordinates of the bounding rectangle calculated

by cv2. boundingRect (). The rectangle is colored red (\* (0,0,255) \*) and has a thickness

of 2 pixels (‘2\*).

The 'for' loop iterates through each center point in 'detect list and checks if the 'y

coordinate of that point falls within the counting line position with an offset of '+ / - ' some

pixels. If the condition is true, it increments the counter variable by 1, removes that center

point from the detect list, draws a line on the counting line, and prints the current count on

the console.

After the loop, cv2.putText () ' function is used to write the current count on the output

frame at position " (450, 70) ' with the specified font size, color, and thickness.

This code snippet is used to draw bounding boxes around the contours detected in the

image.

The cv2. findContours ()' function returns a list of contours in the image. Each contour is

represented as a list of points, and each point is represented as a tuple of (x, y)

coordinates.

The 'enumerate () ' function is used to loop through the contours and keep track of their

index numbers. For each contour, the cv2. boundingRect () ' function is used to calculate

the coordinates of the bounding rectangle.

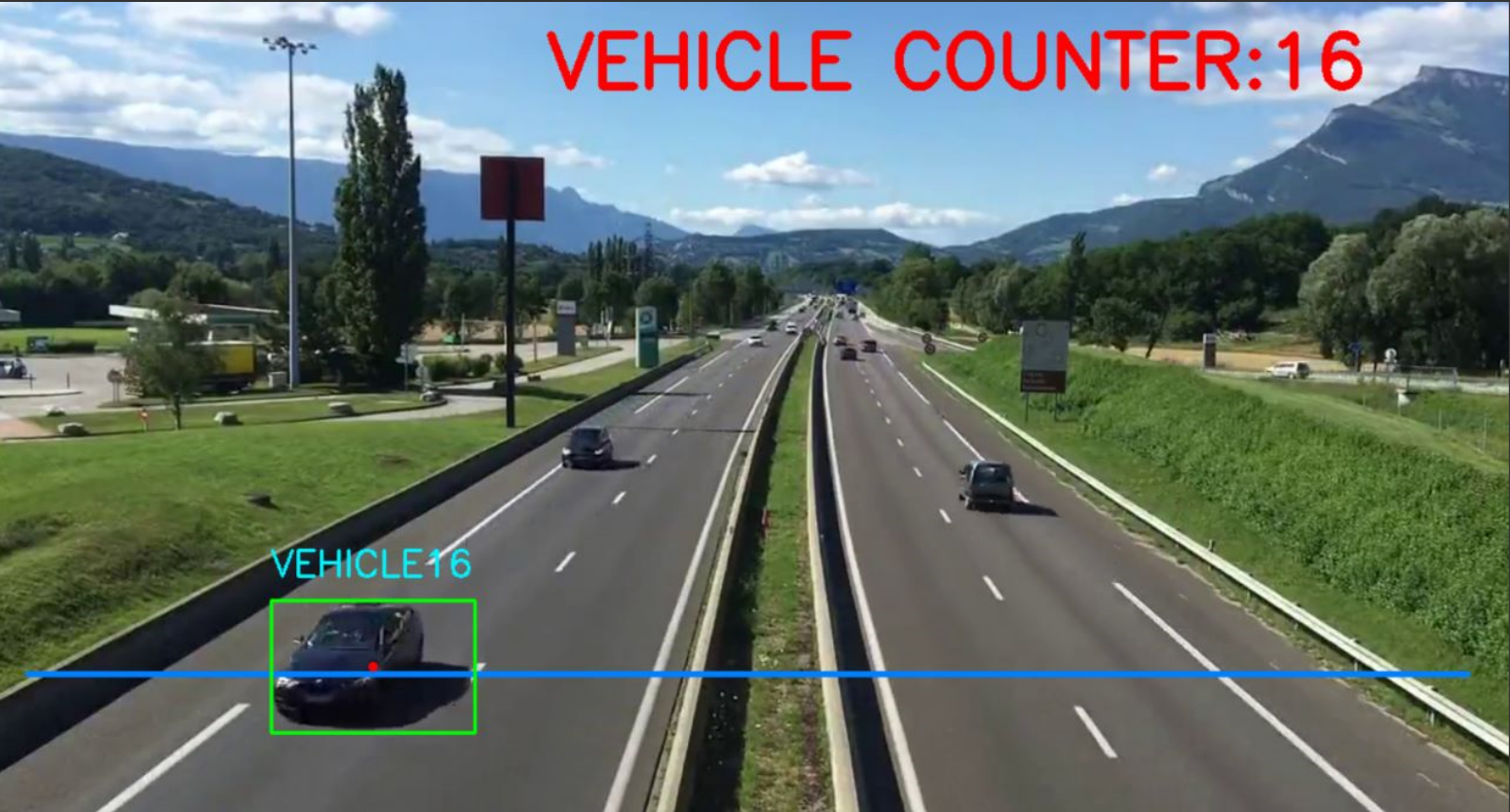
The 'validate\_counter variable is used to check if the width and height of the bounding

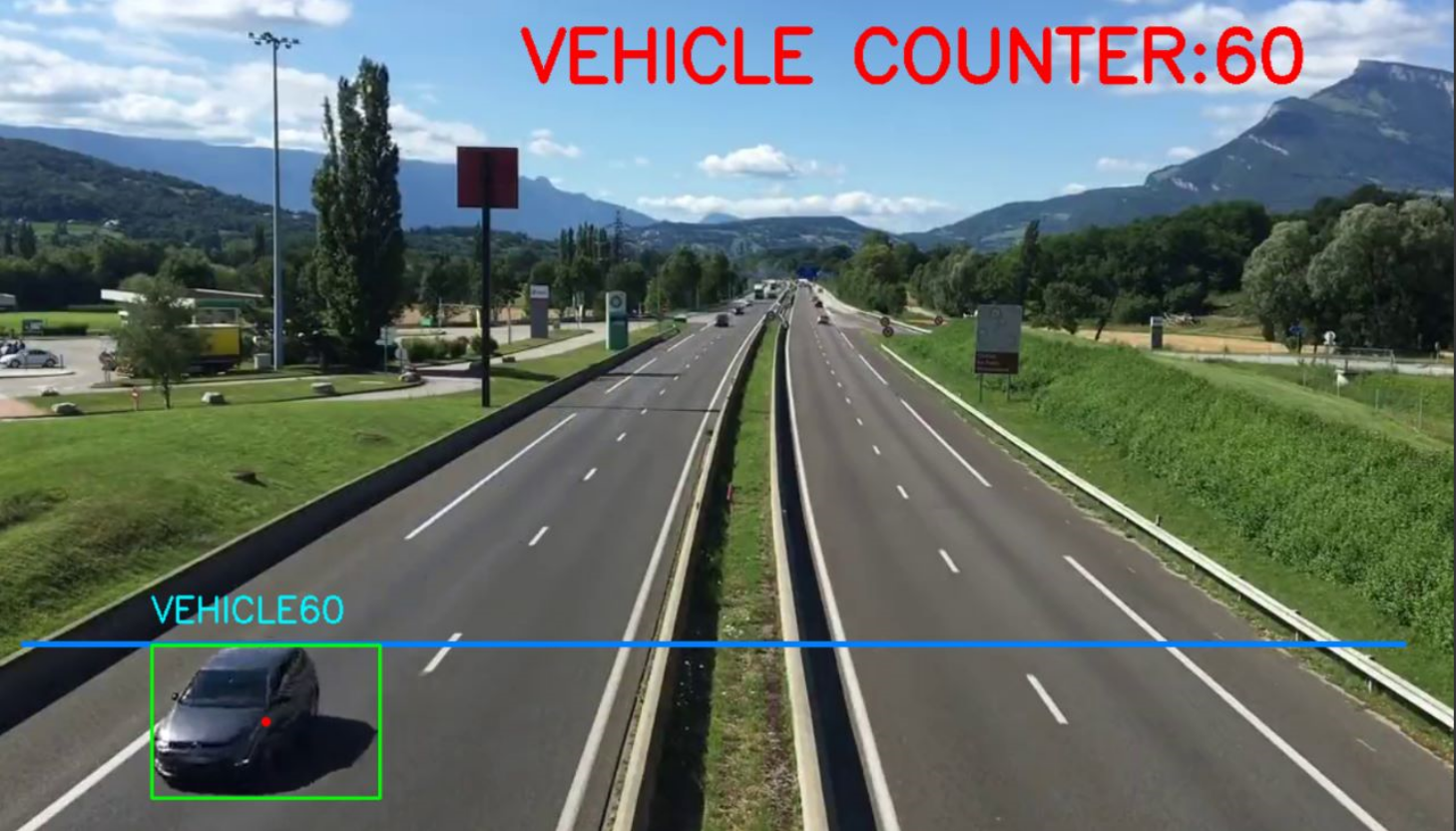
rectangle are greater than or equal to the minimum width and height values .

BACKGROUND SUBTRACTION



TRACKING AND COUNTER PROCESS





1. CONCLUSION

We explored various studies related to the development of vehicle counter programs in Python. The reviewed papers demonstrated the application of computer vision techniques, machine learning algorithms, and deep learning models to accurately detect, track, and count vehicles in different traffic surveillance scenarios.

The studies highlighted the effectiveness of background subtraction, blob analysis, contour detection, optical flow,real-time vehicle counting results. Additionally, the utilization of Python and popular libraries like OpenCV .

The surveyed papers also addressed challenges associated with vehicle counting, including complex traffic environments, occlusion, varying lighting conditions, and aerial surveillance scenarios. The proposed solutions and methodologies provided valuable insights into overcoming these challenges, resulting in improved accuracy, efficiency, and scalability of vehicle counter programs .

XII. REFERENCES

1. "Real-Time Vehicle Counting and Classification for Traffic Surveillance" by Liu et al. (2018)

1. "Traffic Monitoring System Based on Connected Vehicle Technology" by Zhang et al. (2019)

1. "Efficient Vehicle Detection and Tracking for Traffic Surveillance Systems" by Chen et al. (2020)
2. "Automatic Vehicle Counting and Classification Using Deep Learning Techniques" by Wang et al. (2019)
3. "Traffic Flow Monitoring and Prediction Using Machine Learning Techniques" by Li et al. (2021)
4. "Vision-Based Vehicle Detection and Tracking for Traffic Surveillance: A Comprehensive Review" by Gupta et al. (2020)
5. "Intelligent Transportation Systems for Traffic Management: A Survey" by Zhang et al. (2018)

##### license

THIS IS AN OPEN-SOURCE PROJECT