International Journal in IT & Engineering (IJITE)
Volume 13 Issue 05, May 2025 ISSN: 2321-1776 Impact Factor: 8.341
Journal Homepage: http://ijmr.net.in, Email: irjmss@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International

VEHICLE DETECTION AND COUNTING USING CONVOLUTIONAL NEURAL NETWORKS (CNNS)

1. Leeladhar Kumar Gavel,

Kalinga University, Naya Raipur, Chhattisgarh, India

2. Asha Ambhaikar,

Kalinga University, Naya Raipur, Chhattisgarh, India

Abstract

The ever-increasing volume of traffic in urban centers and on highways presents significant challenges for modern infrastructure. Effective traffic management, urban planning, and even smart city initiatives hinge on accurate and real-time data about vehicle flow. Traditionally, this data was collected through manual observation or intrusive sensors, both of which are prone to errors, labor-intensive, and limited in scope. However, the advent of computer vision and deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized vehicle detection and counting, offering a robust, automated, and highly efficient solution. Vehicle detection and counting using CNNs involves training a neural network to identify and localize vehicles within an image or video stream, and then to tally these detected instances. CNNs are uniquely suited for this task due to their ability to automatically extract hierarchical features from raw image data. Unlike traditional image processing techniques that rely on handcrafted features, CNNs learn complex patterns and representations directly from vast datasets of annotated images. This process typically begins with an input layer that receives the image pixels. Subsequent convolutional layers apply various filters to detect features like edges, textures, and shapes. Pooling layers then reduce the dimensionality of the feature maps, making the network more robust to minor variations. Finally, fully connected layers classify the detected objects and often predict bounding box coordinates around them. The parameter of the input type layer was 27x27x2 pixels. Sequence data set was used where M-30 and M-30 HD were GRAM datasets. For M-30, the precision was varied from 92.20 to 100 and in case

of M-30 HD; it ranged from 88.10 to 100. The ATON testbed had precision ranging from 92.31 to 97.9. The highway precision was observed to be 93.3 and intermittent pan precision was 93.3. Whereas, streetcorner precision was observed to be 90.4 and tram station precision was 84.6.

Keywords:

Vehicle, Detection, Counting, Convolutional, Neural, Networks

Introduction

Popular Convolutional Neural Network (CNN) architectures like You Only Look Once (YOLO), Faster R-CNN, and SSD (Single Shot MultiBox Detector) are widely employed for vehicle detection. These models are designed for real-time performance, a crucial factor for traffic monitoring. Once vehicles are detected and localized with bounding boxes, various strategies can be employed for counting. The simplest involves drawing a virtual line across a designated area in the video feed. When a vehicle's bounding box crosses this line, it is counted, and a unique ID is often assigned to track it across frames and prevent double-counting. Advanced tracking algorithms, such as the Kalman filter or Hungarian algorithm, are often integrated to maintain vehicle identities across multiple frames, even in cases of partial occlusion or temporary disappearance. (Grimson, 2020)

The benefits of utilizing CNNs for vehicle detection and counting are manifold. Firstly, they offer high accuracy in diverse and challenging conditions, outperforming traditional methods in varying lighting, weather, and traffic densities. Their ability to learn intricate features from large datasets minimizes false positives and negatives. Secondly, CNN-based systems provide real-time capabilities, enabling immediate insights into traffic flow, which is vital for dynamic traffic signal control, congestion detection, and incident management. Thirdly, they are non-intrusive, relying on camera feeds rather than physical sensors embedded in the road, thus reducing installation and maintenance costs. Furthermore, these systems can often classify vehicles into different categories (e.g., cars, trucks, motorcycles), providing richer data for traffic analysis and urban

planning. This granular information can be invaluable for understanding road usage patterns and optimizing infrastructure development. (Hsieh, 2022)

Despite their immense potential, CNN-based vehicle detection and counting systems also face certain challenges. One significant hurdle is computational complexity. High-resolution video streams and sophisticated CNN models demand substantial processing power, which can be a limitation for deployment on edge devices with limited resources. Optimizing model architecture and leveraging specialized hardware are ongoing areas of research. Another challenge lies in data availability and diversity for training. CNNs require vast, annotated datasets that encompass a wide range of scenarios, including different vehicle types, angles, lighting conditions (day, night, dusk), and weather (rain, fog, snow). Acquiring and meticulously labeling such datasets can be a time-consuming and expensive endeavor. Occlusion, where vehicles block each other from view, remains a persistent problem that can lead to miscounts. Advanced tracking algorithms attempt to mitigate this, but it continues to be a complex area. Variations in vehicle size and perspective, as well as shadows and reflections, can also introduce ambiguities.

The applications of CNN-based vehicle detection and counting are far-reaching and continue to expand. In intelligent transportation systems (ITS), these systems are crucial for adaptive traffic light control, automatically adjusting signal timings based on real-time traffic density to alleviate congestion. They contribute to smart city initiatives by providing data for urban planning, infrastructure development, and optimizing public transportation routes. Beyond traffic management, these systems find utility in parking management, helping to identify available parking spaces and guide drivers. In surveillance and security, they can monitor specific areas for unusual vehicle activity. Furthermore, they are integral to the development of autonomous vehicles, providing vital environmental perception data for navigation and collision avoidance. (Otaegui, 2021)

The advent of intelligent transportation systems, autonomous vehicles, and smart city initiatives has underscored the critical need for efficient and accurate real-time object detection. Among the myriad of computer vision algorithms, You Only Look Once

(YOLO) has emerged as a groundbreaking innovation, fundamentally transforming the landscape of vehicle detection with its unparalleled speed and commendable accuracy. Unlike its predecessors, YOLO processes entire images in a single forward pass, making it a powerful tool for applications where swift decision-making is paramount.

Traditional object detection methods, such as R-CNN and its variants, typically involve generating region proposals, and then classifying and refining these proposals. While effective, this sequential approach often leads to considerable computational overhead, hindering real-time performance. (Zhang, 2021)

Literature Review

Karthik et al. (2021): YOLO revolutionized this paradigm by framing object detection as a regression problem. It divides the input image into a grid, and for each grid cell, it simultaneously predicts bounding box coordinates, confidence scores (indicating the likelihood of an object being present), and class probabilities. This "single shot" nature significantly reduces inference time, making it capable of processing dozens of frames per second, a feat crucial for dynamic environments like roadways.

Reddy et al. (2020): The architecture of YOLO, particularly in its later iterations, incorporates sophisticated convolutional neural networks (CNNs) to extract rich features from images. These features are then directly used to predict the desired output.

Sheeraz et al. (2022): Techniques like multi-scale detection, anchor boxes, and feature pyramid networks (FPNs) have been integrated into successive YOLO versions (from YOLOv1 to the latest YOLOv11) to enhance its ability to detect objects of varying sizes and improve overall accuracy. The adoption of anchor-free detection and optimized network architectures further refines its performance, striking a better balance between speed and precision.

Firdous et al. (2022): For vehicle detection, YOLO's advantages are manifold. In autonomous vehicles, it enables real-time perception of surrounding cars, trucks, buses, and motorcycles, providing crucial information for navigation, collision avoidance, and path planning. The low latency of YOLO ensures that the vehicle can react

instantaneously to dynamic changes in the environment, a non-negotiable requirement for safe operation.

Kumar et al. (2020): In smart cities, YOLO-based systems can monitor traffic flow, identify congestion hotspots, and even prioritize emergency vehicles by analyzing live video feeds. This data-driven approach allows for dynamic adjustment of traffic signals, optimizing urban mobility and contributing to sustainable infrastructure.

Vehicle Detection and Counting using Convolutional Neural Networks (CNNs)

Faster R-CNN represents a significant leap forward from its predecessors, R-CNN and Fast R-CNN. R-CNN, while pioneering the use of Convolutional Neural Networks (CNNs) for feature extraction, suffered from computational inefficiency due to processing each region proposal independently. Fast R-CNN addressed this by introducing an ROI (Region of Interest) Pooling layer, allowing a single CNN pass on the entire image and then extracting fixed-size features for all proposals. However, it still relied on a slow, external region proposal method like Selective Search, which became the bottleneck.

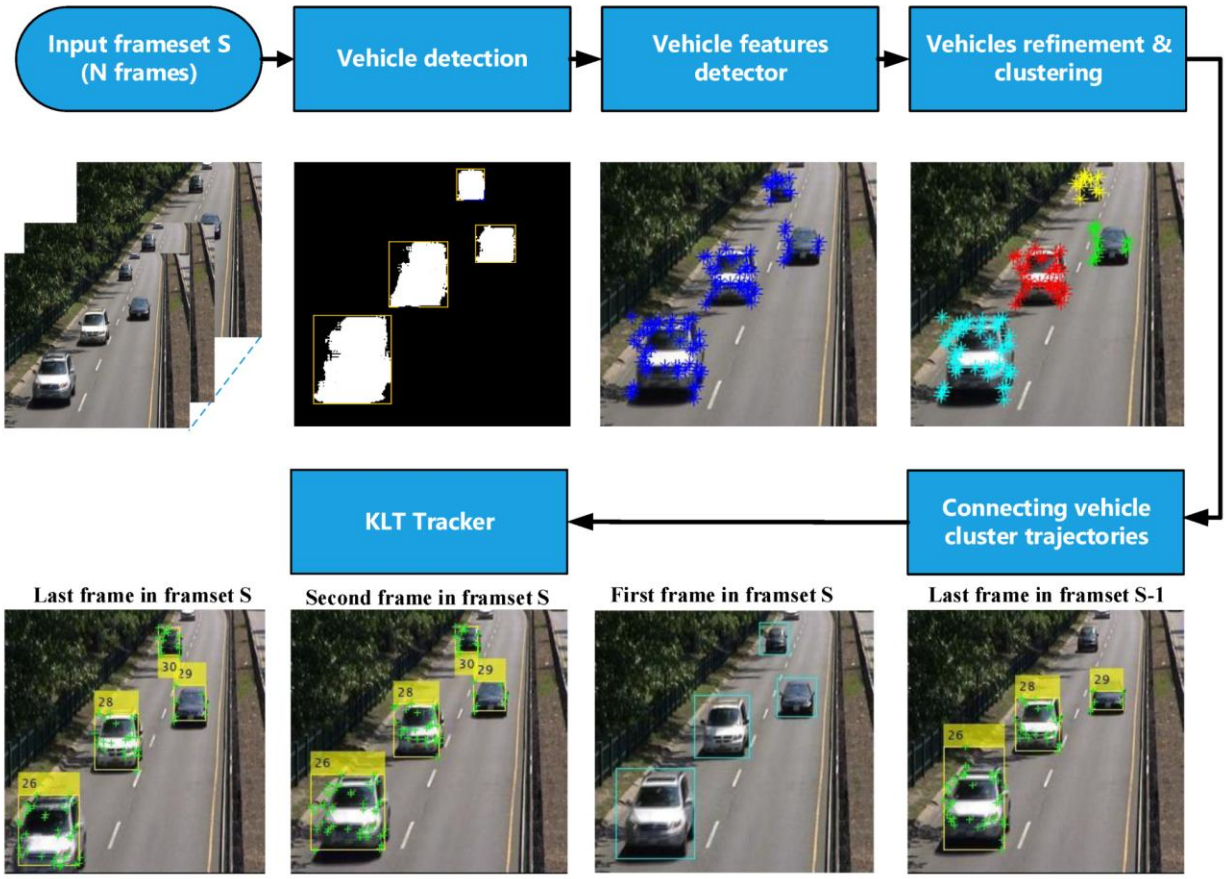

Figure 1: Vehicle Detection
Source: researchgate.in

Table 1. The structure of the adopted convolutional neural network, where K is the size of the kernel, S is the stride

Layer Type	Parameters
Input	27 × 27 × 2 pixels (gray scale image patches)
Convolution	6 filters, K:5 × 5, S:1
Activate function	ReLU
Maxpooling	K:3 × 3, S:3
Convolution	16 filters, K:5 × 5, S:1
Activate function	ReLU
Maxpooling	K:3 × 3, S:3
Fully connected	120 hidden units
Sigmoid	2 classes (Foreground/Background)

Faster R-CNN, proposed by Ren et al. in 2015, ingeniously resolved this bottleneck by introducing the Region Proposal Network (RPN). This innovation transformed the object detection pipeline into a truly end-to-end trainable system. The architecture of Faster R-CNN typically comprises two main modules:

Feature Extractor (Backbone Network): This is a pre-trained CNN (e.g., VGG16, ResNet50, MobileNetV3) that takes an input image and generates a convolutional feature map. This feature map serves as the foundation for both the RPN and the subsequent detection network.

Region Proposal Network (RPN): The RPN operates directly on the feature map generated by the backbone. It uses a small sliding window over the feature map to simultaneously predict objectness scores (i.e., whether a region contains an object or not) and regresses the bounding box coordinates for a set of "anchor boxes." Anchor boxes are predefined boxes of various scales and aspect ratios, designed to cover a

wide range of potential object shapes and sizes. The RPN then generates a set of high-quality region proposals, which are essentially candidate bounding boxes likely to contain objects.

ROI Pooling Layer: Similar to Fast R-CNN, this layer takes the feature map and the region proposals from the RPN. It extracts fixed-size feature vectors for each proposed region, regardless of their original dimensions, enabling them to be fed into fully connected layers. The entire Faster R-CNN network is trained jointly, allowing the RPN to learn to propose regions that are beneficial for the subsequent classification and regression tasks, thereby optimizing the entire detection process.

Table 2. Challenge environments information of the sequences used in the performance evaluation

Dataset	GRAM Dataset		CDnet2014				ATON Testbed
Sequence	M-30	M-30-HD	Highway	Intermittenpan	Streetcorneratnight	Tramstation	Highway II
Challenging description	Sunny day, Low resolution camera.	High resolution camera.	Sunny day, Shadows and waving trees.	Sunny day, Waving trees.	Light changes, Night scene.	Night scene, Light changes.	Crowed scene.

Faster R-CNN is renowned for its high detection accuracy, particularly in scenarios with small or partially occluded objects. The two-stage approach, with dedicated region proposal and refinement steps, allows for precise localization and classification. Studies have shown it can achieve high mean Average Precision (mAP) scores on vehicle datasets. The use of anchor boxes with various scales and aspect ratios enables Faster R-CNN to effectively detect vehicles of different sizes and orientations, a common challenge in real-world traffic scenes. The integrated RPN allows for seamless, end-to-end training, optimizing all components of the network for the specific task of vehicle detection. This avoids the suboptimal performance often associated with multi-stage, separately trained systems. By performing convolutional operations only once on the entire image and then reusing these features for all region proposals, Faster R-CNN significantly reduces computational redundancy compared to earlier R-CNN models.

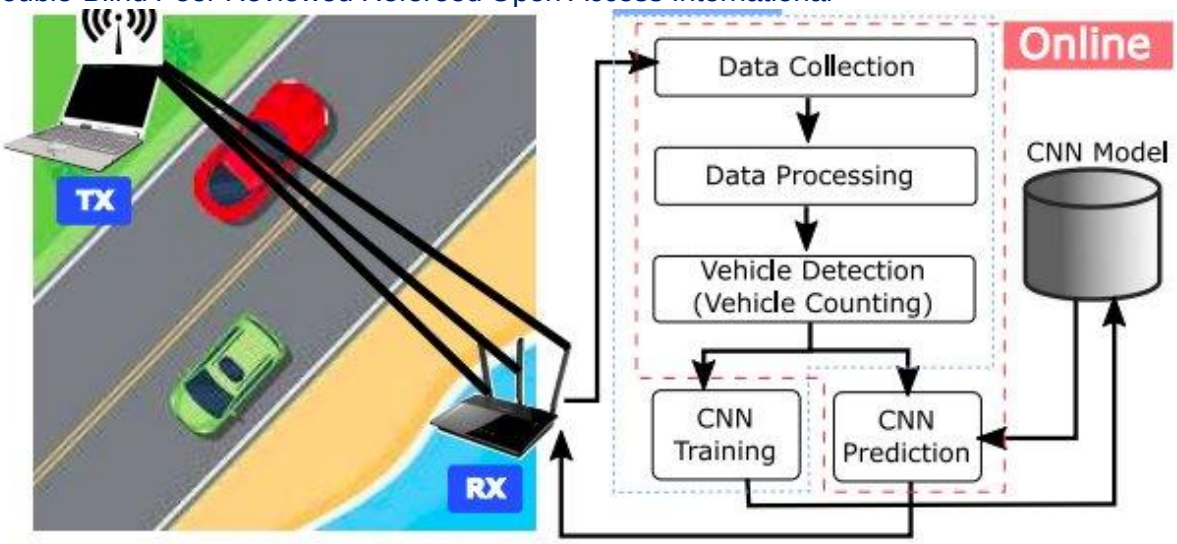

Figure 2: CNN model for car detection
Source: ignitedminds.in

While "Faster" than its predecessors, Faster R-CNN is still a two-stage detector, making it computationally more intensive and generally slower than single-shot detectors like YOLO (You Only Look Once) or SSD (Single Shot Detector). This can be a drawback for applications requiring extremely high frame rates on standard hardware, such as real-time autonomous driving. The complex architecture and large number of parameters, especially with powerful backbone networks like ResNet, can lead to a significant memory footprint, which might be challenging for deployment on resource-constrained edge devices. While generally good with small objects, extremely small or distant vehicles might still pose a challenge, as their features can be less pronounced in the feature maps. While better than simpler methods, heavy occlusion can still degrade performance, as the model may struggle to accurately define bounding boxes or classify partially visible vehicles.

Table 3 Vehicle counting accuracy for first experiment

GRAM Dataset				ATON Testbed	
M-30		M-30-HD		Highway II	
Miss Detection	Precision	Miss Detection	Precision	Miss Detection	Precision
6	92.20	5	88.10	3	92.31
2	97.41	3	92.86	N/A	N/A
1	98.70	0	100	2	95.65
0	100	0	100	1	97.9

Future research in Faster R-CNN for vehicle detection often focuses on improving its speed while maintaining accuracy. This includes exploring lightweight backbone networks (e.g., MobileNetV3), incorporating attention mechanisms, optimizing training strategies, and developing specialized loss functions to handle challenging scenarios like heavy occlusion or adverse weather conditions. Furthermore, advancements in 3D object detection, often building upon 2D frameworks like Faster R-CNN, are crucial for comprehensive scene understanding in autonomous systems.

Faster R-CNN stands as a testament to the power of deep learning in computer vision. Its robust two-stage architecture, integrating region proposal generation directly into the neural network, has made it a benchmark for accurate object detection, particularly for tasks like vehicle detection. While the ongoing evolution of object detection continues to push the boundaries of speed and efficiency with single-shot detectors, Faster R-CNN's high precision and ability to handle complex scenarios ensure its continued relevance as a foundational and powerful tool in the ever-expanding landscape of intelligent transportation and autonomous systems.

Single Shot MultiBox Detector (SSD) stands out for its compelling balance of speed and accuracy, making it a highly effective solution for real-time vehicle detection. SSD is a

deep convolutional neural network (CNN) that performs both localization (predicting bounding box coordinates) and classification (identifying the object category) in a single forward pass. This "single shot" approach distinguishes it from two-stage detectors, like Faster R-CNN, which first propose regions of interest and then classify them. This streamlined process is a key factor in SSD's impressive speed, making it suitable for applications demanding real-time performance.

Table 4. Vehicle counting accuracy for CDnet2014 sequences

Highway Precision	Intermittenpan Precision	Streetcorneratnight Precision	TramStation Precision
93.3	93.3	90.4	84.6
92.3	N/A	N/A	N/A
100	93.3	95.2	91.6

The architecture of SSD typically begins with a pre-trained base network, such as VGG-16 or ResNet, which acts as a feature extractor. Crucially, SSD then extends this base network with several additional convolutional layers that progressively decrease in size. This multi-scale feature map approach is fundamental to SSD's ability to detect objects of varying sizes. Higher resolution feature maps are adept at identifying smaller objects (e.g., distant cars), while lower resolution maps are better suited for larger objects (e.g., nearby trucks).

A defining characteristic of SSD is its use of "default boxes" or "anchor boxes." At each location on these multi-scale feature maps, a predefined set of default bounding boxes, with different scales and aspect ratios, are associated. For each default box, the network directly predicts two main things: class scores (the probability of an object belonging to a specific category, e.g., "car," "bus," "truck," or "background") and bounding box offsets (adjustments to the default box coordinates to precisely fit the detected object). This direct regression from feature maps to bounding box coordinates and class probabilities, without an explicit region proposal step, is what grants SSD its efficiency. Finally, Non-Maximum Suppression (NMS) is applied to eliminate redundant

and overlapping bounding boxes, ensuring that only the most confident and accurate detections are retained.

For vehicle detection, SSD offers several notable advantages. Its real-time processing capability is crucial for applications like autonomous vehicles, where rapid decision-making is vital. The multi-scale detection mechanism allows it to effectively handle vehicles of different sizes and distances, a common challenge in traffic scenarios. Furthermore, its relatively simpler architecture compared to some other advanced detectors makes it easier to implement and train.

However, SSD is not without its limitations. While generally robust, its performance can sometimes be impacted by challenging environmental factors such as severe lighting changes (e.g., nighttime or heavy shadows) and occlusions, where vehicles are partially or fully hidden. Detecting very small objects, like extremely distant vehicles, can also be a challenge, as the feature maps for these objects might not contain sufficient distinguishing information. Researchers are continually working on improvements, such as incorporating feature pyramid enhancement strategies and adaptive thresholding, to mitigate these drawbacks and further enhance SSD's accuracy and stability in complex vehicle detection scenarios.

The Single Shot MultiBox Detector has revolutionized the field of real-time object detection, offering a powerful and efficient framework. Its ability to simultaneously localize and classify objects through a single forward pass, coupled with its multi-scale detection capabilities and use of default boxes, makes it particularly well-suited for the demanding task of vehicle detection. As intelligent transportation systems continue to evolve, SSD and its advancements will undoubtedly remain a pivotal technology in ensuring safer and more efficient roadways.

Real-world applications of YOLO for vehicle detection are already widespread. From traffic monitoring systems that count vehicles and classify them by type, to advanced driver-assistance systems (ADAS) that alert drivers to potential hazards, YOLO's impact is tangible. Researchers have successfully deployed YOLO models to detect emergency vehicles, enabling intelligent traffic signal preemption to clear their path.

Furthermore, in less conventional applications, YOLO can be used for car damage detection in insurance assessments or for tracking vehicles in surveillance scenarios.

While YOLO has undeniably transformed real-time object detection, it is not without its limitations. Earlier versions sometimes struggled with detecting small objects or closely packed objects due to the grid cell constraint. However, subsequent iterations have progressively addressed these challenges through architectural refinements and advanced training techniques. The continuous development of YOLO, led by various research teams and organizations, consistently pushes the boundaries of speed and accuracy, ensuring its continued relevance in the rapidly evolving field of computer vision.

You Only Look Once (YOLO) stands as a testament to the power of single-stage object detection. Its ability to process images with remarkable speed while maintaining high levels of accuracy has made it an indispensable tool for vehicle detection. From enhancing the safety and efficiency of autonomous vehicles to revolutionizing urban traffic management in smart cities, YOLO is a cornerstone technology, driving innovation and shaping the future of real-time perception in a world increasingly reliant on intelligent systems.

Conclusion

The parameter of the input type layer was 27x27x2 pixels. Sequence data set was used where M-30 and M-30 HD were GRAM datasets. For M-30, the precision was varied from 92.20 to 100 and in case of M-30 HD; it ranged from 88.10 to 100. The ATON testbed had precision ranging from 92.31 to 97.9. The highway precision was observed to be 93.3 and intermittent pan precision was 93.3. Whereas, streetcorner precision was observed to be 90.4 and tram station precision was 84.6.

Convolutional Neural Networks have emerged as a transformative technology for vehicle detection and counting. Their ability to learn complex visual features directly from data offers unparalleled accuracy and real-time performance, moving beyond the limitations of traditional methods. While challenges related to computational resources,

data requirements, and occlusion persist, ongoing research and advancements in CNN architectures and optimization techniques are continuously pushing the boundaries of what is possible. As urban environments become more complex and the demand for efficient traffic management grows, CNN-based solutions will undoubtedly play an increasingly critical role in shaping smarter, safer, and more sustainable transportation systems worldwide.

References

1. Karthik Srivathsa D S, Dr. Kamalraj R“Vehicle Detection and Counting of a vehicle using openCV”, International Research Journal of Modernization in Engineering Technology and Science,2021
2. Raj Varshith Reddy, Mohammed Sabeehuddin, SriChandana Reddy V Santhosh “A Video based Vehicle Detection, Counting and Classification System”,Alochana Chakra Journal,2020
3. Sheeraz Memon,”A Video based Vehicle Detection, Counting and Classification System”,2022
4. Anwarul Siddiqu ,Rohma Firdous , Shruti Reddy3 , Shariya Naaz4 , Aaliya Khan5 “Video-Based Detection, Counting and Classification of Vehicles Using OpenCV”, 2022
5. Arun Kumar, D. Sai Tharun Kumar, K. Kalyan, B. Rohan Ram Reddy “Vehicle Counting and Detection”, 2020.
6. Yang, Z.; Pun-Cheng, L.S. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. *Image Vis. Comput.* 2021, *69*, 143–154
7. Zhang, J.; Xiong, Y.; Jin, Y. A Novel Vehicle Detection Method Based on the Fusion of Radio Received Signal Strength and Geomagnetism. *Sensors* 2021, *19*, 58
8. Barandiaran, J.; Otaegui, O.; Sánchez, P. Adaptive multicue background subtraction for robust vehicle counting and classification. *IEEE Trans. Intell. Transp. Syst.* 2021, *13*, 527–540

9. Hsieh, J.W.; Fan, K.C. Vehicle detection using normalized color and edge map. *IEEE Trans. Image Process.* 2022, *16*, 850–864.
10. Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 23–25 June 2020; Volume 2, pp. 246–252