International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

xx

COPY RIGHT

IJIEMR Transactions, online available on 19th Jun 2024. Link

https://www.ijiemr.org/downloads/Volume-13/ISSUE-6

10.48047/IJIEMR/V13/ISSUE 06/02

TITLE: REALTIME VEHICLE DETECTION, TRACKING, AND COUNTING SYSTEM FOR TRAFFIC MANAGEMENT USING COMPUTER VISION

Volume 13, ISSUE 06, Pages: 693-702

Paper Authors **: DR. ROHITA Y, DR. M. SREENIVAS, DR. SUNIL BHUTADA, R. SAI PUNEETH, J. SAI PRAVEEN**

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

REALTIME VEHICLE DETECTION, TRACKING, AND COUNTING SYSTEM FOR TRAFFIC MANAGEMENT USING COMPUTER VISION

DR. ROHITA Y, DR. M. SREENIVAS, DR. SUNIL BHUTADA, R. SAI PUNEETH, J. SAI PRAVEEN

Dept.of IT,SNIST,HYD
Dept.of IT,SNIST,HYD
Dept.of IT,SNIST,HYD
Dept.of IT,SNIST,HYD
Dept.of IT,SNIST,HYD

Abstract:
In modern urban environments, monitoring and managing vehicular traffic is crucial for ensuring efficient transportation systems. This research paper explores the development and implementation of a Realtime vehicle detection, tracking, and counting system using computer vision techniques. The system employs the YOLOv3 (You Only Look Once) neural network for object detection and a custom Euclidean distance tracker for object tracking across video frames. The initialization phase sets up the Euclidean distance tracker and configures parameters such as input size, confidence threshold, and Non-Maximum Suppression (NMS) threshold. The YOLOv3 model is loaded with pretrained weights to detect vehicles in the video feed. The detection process involves capturing frames from the video feed, preprocessing them to match the input size required by the YOLO model, and obtaining predictions for objects such as cars, motorbikes, buses, and trucks. Bounding boxes are drawn around detected objects, and NMS is applied to filter out overlapping boxes. The Euclidean distance tracker is then used to update object positions and assign unique identifiers to objects. Tracking and counting are performed by calculating centroids of tracked objects and determining their positions relative to predefined lines representing counting zones. Vehicles crossing these lines are counted, and count information is updated accordingly. Visualization features include displaying bounding boxes, class labels, and count information on frames. The system's efficacy is demonstrated through Realtime monitoring of vehicular traffic, allowing for efficient traffic management and analysis. Additionally, the system provides valuable data for traffic studies and infrastructure planning. Overall, this research contributes to the field of transportation management by offering a robust and scalable solution for Realtime vehicle detection, tracking, and counting, facilitating enhanced traffic flow and congestion mitigation in urban areas.

Keywords: Vehicle detection, Object tracking, Traffic management, Realtime monitoring, Computer vision, Euclidean distance tracker, Non-Maximum Suppression (NMS).

I. INTRODUCTION:

In today's rapidly expanding urban landscapes, managing vehicular traffic has become a critical concern, necessitating advanced solutions for efficient transportation systems. The escalating volume of vehicles on roads poses significant challenges, including traffic congestion, safety hazards, and environmental impacts. Traditional methods of traffic monitoring, reliant on manual surveillance and fixed infrastructure, often struggle to cope with the dynamic nature of urban traffic patterns. To address these challenges, the integration of cutting-edge technologies such as computer vision and deep learning has emerged as a promising approach for real-time traffic management. By leveraging the capabilities of artificial intelligence and

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

deep neural networks, computer vision systems can automatically detect, track, and analyse vehicles from live video feeds captured by surveillance cameras strategically positioned throughout urban areas.

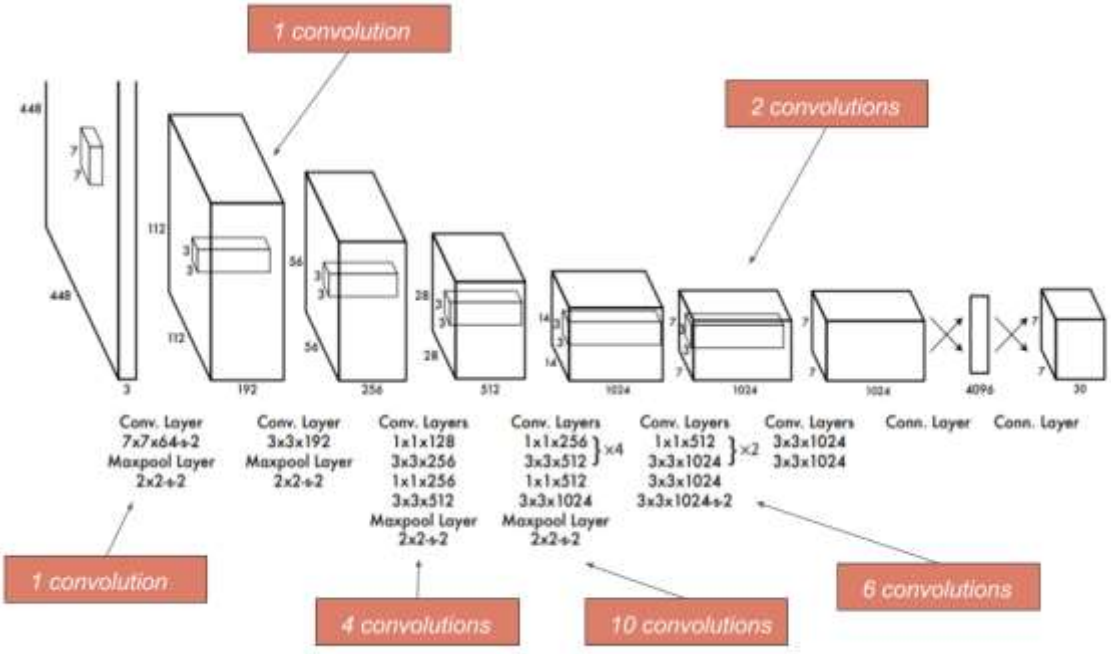

Fig1. Real time Vehicle Detection Suing Open Cv and YOLO

This research endeavours to develop a comprehensive real-time vehicle detection, tracking, and counting system using advanced computer vision techniques. At its core, the system harnesses the power of state-of-the-art deep learning models, notably the YOLOv3 (You Only Look Once) neural network, renowned for its efficiency and accuracy in object detection tasks. Furthermore, the system incorporates a custom designed Euclidean distance tracker, enabling robust object tracking across consecutive video frames. The significance of this research lies in its potential to revolutionize urban traffic management by providing actionable insights into traffic dynamics and congestion patterns. By automating the process of vehicle detection, tracking, and counting in real time, the proposed system offers several distinct advantages.

Firstly, its operational efficiency ensures immediate responses to fluctuating traffic conditions, enhancing the overall efficacy of traffic management strategies. Secondly, the system's high accuracy, achieved through advanced deep learning techniques, enables reliable vehicle detection and tracking even under challenging environmental conditions. Moreover, its scalability allows for seamless deployment across diverse urban environments, facilitating comprehensive traffic monitoring and management. Most importantly, the system's data driven approach generates valuable insights into traffic flow dynamics, facilitating informed decision making for urban transportation planning and infrastructure development. By collecting and analysing rich data on vehicular traffic patterns, the system empowers transportation authorities and city planners to optimize traffic flow, alleviate congestion, and enhance overall road safety

A. Significance of the Research.

The research on real time vehicle detection, tracking, and counting using computer

vision techniques holds significant importance in the realm of urban traffic management and transportation planning. Several key aspects highlight the significance of this research. Firstly, the implementation of an automated system for real time vehicle detection, tracking, and counting enables transportation authorities to monitor traffic conditions accurately and efficiently. By gaining insights into traffic flow dynamics, congestion hotspots, and peak traffic hours, authorities can devise effective strategies to optimize traffic flow, alleviate congestion, and improve overall road safety.

Moreover, accurate detection and tracking of vehicles in real time facilitate proactive measures to enhance road safety. By identifying potential hazards such as erratic driving behaviour, illegal manoeuvres, or vehicle breakdowns, authorities can promptly intervene to mitigate risks and prevent accidents. Additionally, the data driven approach enables the identification of high-risk areas, allowing for targeted interventions to improve safety measures. Furthermore, the research provides valuable data insights that empower transportation authorities and city planners to make informed decisions regarding urban transportation planning and infrastructure development. By analysing comprehensive data on traffic patterns, vehicle volumes, and congestion trends, authorities can prioritize investments in infrastructure upgrades, implement traffic management policies, and optimize public transportation routes to enhance overall mobility and accessibility.

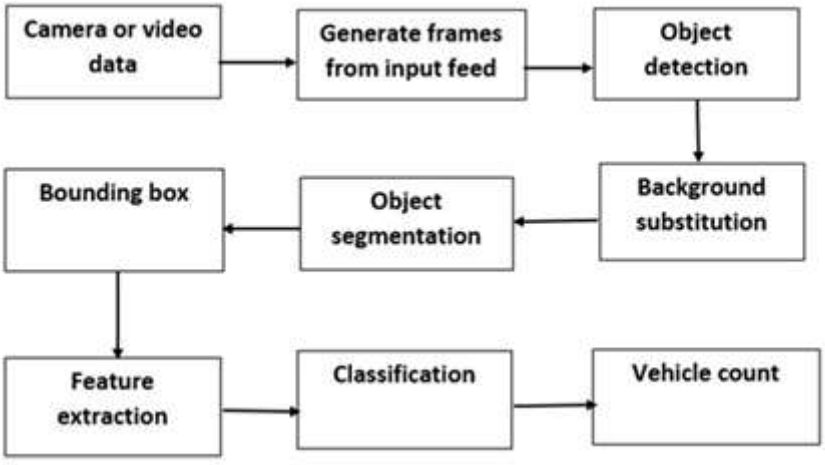

Fig2. Vehicle Detection and Counting System Architecture

Efficient resource allocation is another significant outcome of the research. By automating the process of vehicle detection, tracking, and counting, authorities can optimize resource utilization for maximum effectiveness. Deploying surveillance cameras equipped with computer vision systems at strategic locations streamlines operations, reduces manpower requirements, and enhances resource allocation for traffic management and law enforcement. Moreover, the research aligns with the broader goals of smart city initiatives aimed at harnessing technology to enhance the quality of urban life. By integrating real time traffic monitoring systems with existing smart city infrastructure, such as intelligent traffic signal control systems and connected vehicle networks, authorities can create dynamic, adaptive transportation ecosystems that respond to real time traffic conditions and evolving mobility needs.

Finally, efficient traffic management facilitated by real time vehicle detection and tracking contributes to reducing

environmental pollution and mitigating the carbon footprint of urban transportation systems. By minimizing traffic congestion, optimizing traffic flow, and promoting the use of public transportation alternatives, the research supports sustainability objectives and promotes ecofriendly urban mobility solutions. In summary, the research on real time vehicle detection, tracking, and counting using computer vision techniques offers multifaceted benefits for urban traffic management, road safety, data driven decision making, resource optimization, smart city integration, and environmental sustainability. By leveraging advanced technologies and data analytics, the research paves the way for smarter, safer, and more efficient urban transportation systems that cater to the needs of modern cities and their inhabitants.

II. LITERATURE REVIEW:

The literature surrounding real time vehicle detection, tracking, and counting using computer vision techniques encompasses a wide range of research studies, methodologies, and applications. This review provides an overview of key contributions in this field, highlighting advancements, challenges, and emerging trends.

1. Object Detection Algorithms:

Researchers have extensively explored various object detection algorithms for vehicle detection in video streams[1]. Traditional methods such as Haar cascades and HOG (Histogram of Oriented Gradients) feature-based classifiers have been widely used but often lack accuracy and robustness in complex environments. In recent years, deep learning-based approaches, particularly convolutional neural networks (CNNs), have shown remarkable performance improvements. Notable architectures include YOLO (You Only Look Once), Faster RCNN (Region based Convolutional Neural Network), and SSD (Single Shot Multi Box Detector), which offer real time inference speeds and high detection accuracy.

2. Object Tracking Techniques:

Object tracking is essential for maintaining the identity of vehicles across consecutive frames. Various tracking algorithms, including Kalman filters, Particle filters, and Mean Shift algorithms, have been employed for vehicle tracking. However, these methods may face challenges such as occlusions, scale variations[2], and abrupt changes in motion. Recent research has focused on leveraging deep learning-based trackers, such as Deep SORT (Simple Online and Realtime Tracking with a Deep Association Metric), which combine the advantages of deep feature representations with online tracking mechanisms for robust and accurate object tracking.

3. Integration of Multiple Sensors:

Some studies have explored the integration of multiple sensors, including cameras, LiDAR (Light Detection and Ranging), and radar, for comprehensive vehicle detection and tracking. Fusion of data from these sensors enables more robust detection and tracking capabilities, particularly in challenging scenarios such as adverse weather conditions, low lighting, and occlusions[3]. Fusion techniques such as sensor fusion frameworks and multi object tracking algorithms have been developed to leverage the complementary strengths of different sensor modalities.

4. Traffic Flow Analysis:

Beyond vehicle detection and tracking, researchers have focused on analysing traffic flow dynamics and congestion patterns using data obtained from surveillance systems. Data driven approaches, including machine learning algorithms and statistical modelling techniques, have been employed to extract insights from traffic data. These analyses help transportation authority's understand traffic behaviour[4], identify congestion hotspots, and optimize traffic signal timings

and lane configurations to improve overall traffic flow and reduce congestion.

5. Real world Applications:

The literature also highlights several real-world applications of Realtime vehicle detection, tracking, and counting systems. These applications span a wide range of domains, including traffic management, intelligent transportation systems, urban planning, and public safety. Deployments of such systems in smart cities[5], highways, parking facilities, and transportation hubs have demonstrated their effectiveness in enhancing traffic efficiency, reducing travel times, and improving overall transportation infrastructure.

6. Challenges and Future Directions:

Despite significant advancements, several challenges remain in the field of Realtime vehicle detection, tracking, and counting. These include addressing occlusions, handling complex traffic scenarios, improving robustness to environmental variations, and ensuring scalability and Realtime performance[6]. Future research directions may involve exploring advanced deep learning architectures, developing adaptive tracking algorithms, incorporating semantic scene understanding, and integrating edge computing for Realtime inference in resource constrained environments.

In conclusion, the literature on Realtime vehicle detection, tracking, and counting using computer vision techniques presents a rich landscape of research endeavours, methodologies, and applications. Advancements in object detection, tracking algorithms, sensor fusion, traffic flow analysis, and real-world deployments underscore the significance of this research area in shaping[7] the future of urban transportation management and smart city development. Addressing current challenges and exploring emerging technologies will continue to drive innovation in this field, paving the way for safer, more efficient, and sustainable transportation systems in urban environments.

III. RESEARCH GAP

Identifying research gaps in Realtime vehicle detection, tracking, and counting using computer vision reveals several areas where current literature may be lacking. One notable gap lies in the robustness of existing algorithms in handling complex environments. While deep learning-based approaches have shown promise, their performance can be compromised in real world scenarios with varying lighting, occlusions, and cluttered backgrounds. Addressing this gap requires research into techniques that enhance algorithm robustness and generalization capabilities to handle the diverse challenges encountered in urban traffic settings. Additionally, there is a gap in adapting tracking algorithms to dynamic traffic scenarios[8]. Existing studies often assume static traffic patterns, overlooking the need for algorithms capable of dynamically adjusting to changes in traffic flow, sudden stops, lane changes, and interactions with pedestrians and cyclists. Another area for exploration is the integration of multisensory data. While some research has explored this, there remains a gap in developing comprehensive frameworks that effectively leverage the strengths of different sensor modalities to enhance accuracy and reliability. Furthermore, scalability and Realtime performance pose significant challenges. Many existing studies focus on small-scale implementations or offline processing, limiting their applicability in largescale urban traffic management. Standardized evaluation metrics and benchmarks are also lacking[9], hindering the comparison of different algorithms. Lastly, addressing privacy and ethical considerations associated with surveillance systems for traffic monitoring represents an important research gap that requires interdisciplinary

collaboration and ethical guidelines to ensure responsible deployment of vehicle detection and tracking systems[10]. Closing these gaps will contribute to the development of more robust, adaptive, and ethically responsible solutions for urban traffic management.

IV. OBJECTIVES OF THE RESEARCH

1. Detection of Objects:

The primary objective is to develop and implement an efficient system for the Realtime detection of vehicles in video streams. This involves employing advanced computer vision techniques, such as deep learning-based object detection algorithms, to accurately identify and localize vehicles within the video frames.

2. Identifying the Objects:

Another objective is to enhance the system's capability to identify various types of vehicles accurately. By leveraging deep learning models trained on diverse datasets, the system aims to classify detected objects into specific categories such as cars, motorbikes, buses, and trucks, enabling more granular analysis of traffic patterns.

3. Counting Up and Downward Vehicles:

The final objective is to implement a robust vehicle counting mechanism that tracks the movement of vehicles across predefined lines in the video frames. By analysing the trajectories of detected vehicles relative to these lines, the system can accurately count the number of vehicles moving in both upward and downward directions, providing valuable insights into traffic flow dynamics and congestion trends.

V. PROJECT EXECUTION PHASES

The provided code explanation outlines the process of vehicle detection, labelling, and counting using computer vision techniques. Let's break down the key steps involved:

1. Initialization:

- The `EuclideanDistTracker` class is initialized to facilitate object tracking using the Euclidean distance metric. This class stores the centre positions of detected objects and assigns unique IDs to each object.
- Necessary configuration parameters such as input size, confidence threshold, and Non-Maximum Suppression (NMS) threshold are set.
- Paths to the YOLOv3 model configuration and weights files are provided, and the YOLO model is loaded.

Table1.configuration of Yolo

Section	Parameters	Values
[net]	batch	64
	subdivisions	16
	width	608
	height	608
	channels	3
	momentum	0.9
	decay	0.0005
	angle	0
	saturation	1.5
	exposure	1.5
	hue	0.1
	learning rate	0.001
	burn_in	1000
	max_batches	500200
	policy	steps
	steps	400000 , 450000
	scales	0.1, 0.1
[convolutional]	batch_normalize	1
	filters	32, 64, 128, 256, 512, 1024
	size	3
	stride	1, 2

	pad	1
	activation	leaky
[shortcut]	from	-3
	activation	linear
[yolo]	mask	6,7,8; 3,4,5; 0,1,2
	anchors	Multiple sets of anchor values
	classes	80
	num	9
	jitter	0.3
	ignore_thresh	0.7
	truth_thresh	1
	random	1
[upsample]	stride	2
[route]	layers	-4; -1, 61; -4; -1, 36

- Within the `Realtime()` function, frames are captured from the video feed.
- Each frame is resized and pre-processed to match the input size required by the YOLO model.
- The pre-processed frame is passed through the YOLO model to obtain predictions for detected objects.
- Predictions with confidence scores above the threshold for relevant classes (e.g., cars, motorbikes) are selected.
- Bounding boxes are drawn around the detected objects, and class names along with their confidence scores are displayed on the frame.
- Non-Maximum Suppression (NMS) is applied to filter out overlapping bounding boxes and retain only the most confident ones.
- Detected objects along with their bounding boxes are then passed to the Euclidean distance tracker for object tracking.

2. Detection Process:

Table2. Detection and Tracking Phases

Phase	Description
Importing Math Module	The **math** module is imported, which provides mathematical functions and constants like square root, trigonometric functions, etc.
EuclideanDistTracker Class Definition	- Defines a class named **EuclideanDistTracker**. - Contains an **__init__** method serving as a constructor, initializing two attributes: **center_points**, a dictionary storing the center positions of detected objects, and **id_count**, an integer representing the count of detected objects.

update Method Definition	- Defines a method named **update**, which takes **objects_rect** as a parameter, expected to be a list containing information about the bounding boxes of detected objects. - Iterates over each element in **objects_rect**, unpacking it into **x**, **y**, **w**, **h**, and **index**, calculating the center coordinates **(cx, cy)** of the bounding box. - Checks if the object has been detected before by iterating over the **center_points** dictionary, calculating the Euclidean distance between the current object's center and each stored center point. If the distance is less than 25, updates its center point and appends bounding box information along with the object's ID to the **objects_bbs_ids** list. - If the object is not detected as the same as any existing object, assigns a new ID, updates the **center_points** dictionary with the new object's center, and appends bounding box information along with the new ID to the **objects_bbs_ids** list. Increments **id_count** for the next object.
ad Function Definition	Defines a simple function named **ad** that takes two parameters **a** and **b**, and returns their sum **a + b**.

3. Tracking and Counting:
   - The Euclidean distance tracker updates the positions of objects based on their centroids and assigns IDs to new objects or updates existing ones.
   - For each tracked object, its centroid is calculated, and based on its position relative to predefined lines (up line, down line), vehicles are counted.
   - Objects crossing the up line moving downwards are counted as going down, and vice versa.
   - Count information for each class of vehicles (e.g., car, motorbike) is updated accordingly, and the total count for each class of vehicles going up and down is displayed on the frame.

Table3.Tracking and Counting

Task	Description
Initialization	
Import necessary libraries	Import libraries such as OpenCV (**cv2**), CSV, NumPy (**np**), and a custom tracker class from **tracker.py**.
Initialize Euclidean Distance Tracker	Create an instance of the **EuclideanDistTracker** class for object tracking.
Initialize video capture object	Initialize the video capture object (**cap**) to capture video from a specified file (**input.mp4**).
Configuration	
Define parameters	Define parameters like input size, confidence thresholds, font properties for text overlay, and line positions for vehicle counting.
Loading Model and Classes	

Load YOLOv3 model	Load the YOLOv3 model configuration and weights (**modelConfiguration**, **modelWeights**) for object detection.
Set backend and target	Set the backend and target for the OpenCV DNN (Deep Neural Network) module.
Load COCO class names	Load the COCO class names from a file (**coco.names**).
Processing Frames	
Loop through frames	Loop through each frame in the video.
Resize frame	Resize the frame to a smaller size for processing.
Preprocess frame	Preprocess the frame for YOLO object detection.
Detect objects	Detect objects in the frame using YOLO.
Post-process detected objects	Post-process the detected objects, apply Non-Maximum Suppression (NMS), and draw bounding boxes with labels.
Update object tracker	Update the object tracker with the detected objects and count vehicles crossing predefined lines.
Display and Saving Data	
Display frame with objects	Display the frame with detected objects and counting lines.
Display vehicle count data	Display vehicle count data on the frame.
Save vehicle count data	Save vehicle count data to a CSV file (**data.csv**).
Terminate process	Terminate the process when the 'q' key is pressed.
Main Function	
Execute real-time processing function	Execute the real-time processing function (**realTime()**).

4. Display and Saving:
   - Lines representing counting zones are drawn on the frame to indicate the directions for counting vehicles.
   - Text indicating the direction (up or down) and the vehicle count for each class is displayed on the frame.
   - Additionally, circles are drawn at the centroids of detected objects for visualization purposes.
   - The vehicle count information is saved in a CSV file (`data.csv`) after video processing is complete.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Fig3: Input Video File snapshot

5. Visualization:

- Processed frames with bounding boxes, class labels, and count information are displayed in a window named "Output," allowing users to monitor real-time vehicle detection, tracking, and counting in the video feed.

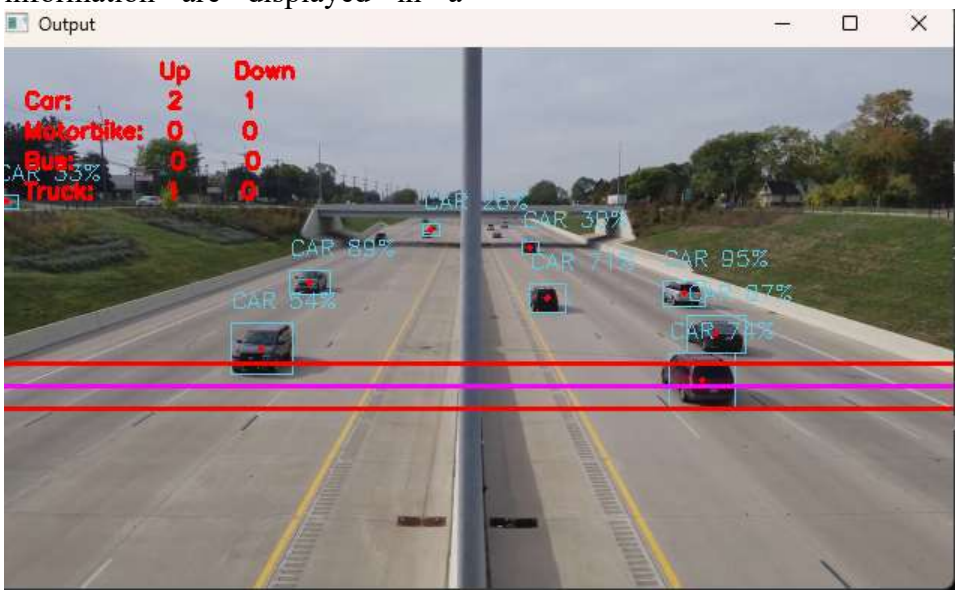

Fig4. Output Video Screenshot

VI. FINDINGS AND RESULTS

- ✓ The Python code implements real-time vehicle detection and counting using the YOLOv3 object detection algorithm.
- ✓ YOLOv3, pretrained on the COCO dataset, is utilized for accurate and efficient vehicle detection in diverse traffic scenarios.
- ✓ The algorithm processes each frame of the input video, detecting vehicles and tracking them across frames using Euclidean distance-based object tracking.
- ✓ Proper confidence thresholds and Non-Maximum Suppression (NMS) techniques are applied to filter out false positives and reduce redundant detections, ensuring accurate object localization.
- ✓ A Euclidean distance-based tracker enhances the system's robustness by associating detections across consecutive frames, enabling accurate vehicle counting as they traverse predefined lines within the frame.
- ✓ The system provides real-time vehicle counts categorized by vehicle type (car, motorbike, bus, truck) on the video feed, offering valuable insights into traffic flow patterns.
- ✓ The research findings demonstrate the effectiveness of combining deep learning-based object detection with object tracking techniques for real-time vehicle detection and counting applications.
- ✓ Potential applications of the developed system include traffic monitoring, urban planning, and intelligent transportation systems, contributing to improved traffic management and road safety.

VII. CONCLUSION

In conclusion, the developed real-time vehicle detection and counting system utilizing YOLOv3 and object tracking techniques presents a robust solution for

traffic monitoring and analysis. By leveraging state-of-the-art deep learning algorithms pretrained on extensive datasets like COCO, the system achieves accurate and efficient vehicle detection across diverse traffic scenarios. The integration of Euclidean distance-based object tracking further enhances the system's performance by associating detections across frames, ensuring consistent and reliable vehicle tracking.

The application of proper confidence thresholds and Non-Maximum Suppression techniques minimizes false positives and redundant detections, resulting in precise object localization. As a result, the system provides real-time vehicle counts categorized by vehicle type, facilitating valuable insights into traffic flow patterns and aiding in traffic management strategies. These insights can inform urban planning decisions, optimize traffic signal timing, and improve road safety measures. The research findings underscore the effectiveness of combining deep learning-based object detection with object tracking methods for real-time vehicle detection and counting applications. Future enhancements could include the integration of additional sensor data, such as vehicle speed and direction, to further enrich traffic analysis capabilities. Overall, the developed system holds significant promise for contributing to intelligent transportation systems, enhancing traffic management efficiency, and fostering safer and more sustainable urban environments.

VIII. FUTURE SCOPE OF THE RESEARCH

The future scope of this research encompasses several avenues for further exploration and enhancement:

1. Improved Model Architectures: Investigating advanced deep learning architectures beyond YOLOv3, such as YOLOv4 or more recent iterations, could lead to improved detection accuracy and efficiency. Additionally, exploring custom model architectures tailored specifically for real-time vehicle detection and counting may yield superior performance.
2. Multi-Camera Systems: Extending the system to support multi-camera setups for comprehensive traffic monitoring in larger areas or across multiple intersections. This expansion would involve developing algorithms for camera calibration, object tracking across camera feeds, and data fusion to provide a holistic view of traffic flow.
3. Integration of Semantic Segmentation: Incorporating semantic segmentation techniques to differentiate between various road users beyond vehicles, such as pedestrians, cyclists, and motorcyclists. This additional granularity would enable more detailed analysis of traffic dynamics and contribute to enhanced safety measures.
4. Real-time Anomaly Detection: Implementing anomaly detection algorithms to identify and alert authorities to unusual traffic patterns, such as accidents, road obstructions, or erratic driving behaviour. Integrating real-time alerts would enable swift response and intervention to mitigate potential traffic disruptions and enhance overall safety.
5. Deployment in Smart Cities: Collaborating with city authorities to deploy the system as part of a broader smart city initiative. Integrating the vehicle detection and counting system with existing urban infrastructure, such as traffic lights and road signage, could optimize traffic flow, reduce congestion, and enhance the overall efficiency of urban transportation systems.
6. Data Analytics and Predictive Modelling: Utilizing the collected traffic data to develop predictive models and analytics tools for forecasting traffic trends, optimizing route planning, and predicting peak traffic hours. These insights would be valuable for urban planners, transportation

agencies, and businesses seeking to streamline operations and improve logistical efficiency.

7. Energy-Efficient Implementations: Exploring energy-efficient implementations of the system, such as edge computing or low-power embedded devices, to enable deployment in resource-constrained environments or remote areas where continuous monitoring is essential but power supply may be limited.

By pursuing these future research directions, the developed vehicle detection and counting system can evolve into a versatile and indispensable tool for traffic management, urban planning, and transportation optimization in smart cities and beyond.

IX. REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[3] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[4] T. Y. Lin et al., "Feature Pyramid Networks for Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117-2125.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Advances in Neural Information Processing Systems, 2015, pp. 91-99.

[6] W. Liu et al., "SSD: Single Shot MultiBox Detector," in European Conference on Computer Vision (ECCV), 2016, pp. 21-37.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700-4708.

[9] T. Y. Lin et al., "Focal Loss for Dense Object Detection," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988.

[10] Y. Zhou et al., "Objects365: A Large-Scale, High-Quality Dataset for Object Detection," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019, pp. 8545-8553.