Grenze International Journal of Engineering and Technology, January Issue

Speed Estimation and Vehicle Tracking using YoloV8

Sahana K J[1] and Dr. Bhat Geetalaxmi Jairam[2]
[1-2]Department of Information Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India
Email: 2023scn_sahanakj_a@nie.ac.in, bgj@nie.ac.in

***Abstract*— Vehicle tracking and speed detection play a vital role in enhancing road safety and managing traffic flow efficiently. Traditional systems like RADAR and LIDAR can be expensive and complex to deploy. In this work, authors investigate and leverage the YOLOv8 deep learning model, which is based on a convolutional neural network (CNN) architecture, combined with centroid tracking and FPS-based speed estimation, to introduce a real-time, cost- effective solution for accurately detecting and monitoring vehicles in video streams. YOLOv8 is a powerful object detection algorithm that offers fast and accurate detection of vehicles in video frames, making it ideal for real-time applications like traffic monitoring. Through thorough evaluation, our system delivers high accuracy in speed estimation. The proposed model is versatile, non-intrusive, and adaptable for various applications such as traffic law enforcement, smart city infrastructure, and autonomous vehicle systems.**

***Index Terms*— YOLOv8, Vehicle Detection, Speed Estimation, Object Tracking, Real-Time Traffic Monitoring, Computer Vision, Deep Learning, Smart Transportation, Traffic Surveillance.**

I. INTRODUCTION

Vehicle detection and speed estimation using YOLO is a smart way to handle traffic. YOLO is a deep learning method that quickly finds vehicles in video footage by dividing the image into sections and identifying objects in each one. It not only spots vehicles but also tracks how far they move across frames to estimate their speed. This helps in understanding traffic flow, improving road safety, and can even support systems like automatic toll collection. It's a fast and efficient tool for modern traffic analysis [1]. Speed estimation systems in the traffic monitoring domain usually utilize traffic monitoring cameras such as CCTV cameras designed to monitor traffic without any specific platform or setting for estimating vehicle speed. The vehicle speed estimation for speed enforcement is used for road safety measures that cause accidents [3].

TrackNCount uses YOLOv8, a powerful real-time object detection model, to identify and classify vehicles in video footage. It outputs bounding boxes, confidence scores, and labels, enabling accurate tracking and analysis. Its ability to handle complex traffic scenes makes it suitable for monitoring highways and urban intersections alike [4]. DeepSORT enhances YOLOv8 by enabling object tracking through the assignment of unique IDs to each detected vehicle. This allows the system to consistently follow vehicles, even if they temporarily leave the camera's field of view [4].

Smart traffic systems are key to building safer cities, and vehicle speed monitoring plays a major role [9]. Speeding in restricted zones— like curves, crossings, or crowded areas—is a leading cause of accidents. While speed limits are set to reduce such risks, many drivers ignore them. Estimating vehicle speed from video footage offers a modern solution, though it can be challenging due to changing environments [5].

Grenze ID: 01.GIJET.12.1.535_2

Fig. 1. AI of Detecting Vehicles and their speed

II. LITERATURE WORK

Considering the problem of Vehicle Tracking and Speed Estimation, the literature mostly consists of empirical and biological works to analyze the determinants.

- Karthikeya Soma, Linu Shibu, Meenakshi N[1] in 2025, developed a vehicle detection and speed estimation system that can detect vehicles and estimate their speed considering different factors. Five object detection algorithms, Histogram of Oriented Gradients (HOG), CNN, R-CNN, Single Shot Detector (SSD) and YOLO were tested to find better results for our model.
- Jabbarul Islam, Md.Tohidul Islam,Md. Golam Rasheed, and Dipankar Das [2] in 2023, concluded that speed estimation using computer vision holds significant potential in real-world applications. They utilized Foreground Shape Analysis (FSA) in intensity level images to conclude that it offers distinct advantages over traditional fore ground extraction techniques.
- Sherif A.Kamal, Ahmed Gaber, Mohammad A.Shehata [3] in 2024, tested three different roads at different times of the day. The results of the automated counts were compared to the manual counts for the same period. Results show that the computerized counts have an accuracy rate of 96%. Indeed, the program has a high accuracy in vehicle counting, but it still has some difficulties in detecting vehicles precisely in rough weather. However, the achieved accuracy rate at a low cost makes the program a good choice for monitoring highways that extend over long distances where the conventional methods are costly and unbearable.
- V.Ceronmani Sharmila, Anant Raj Srivastava, Aaditya Gosain, Yashovardhan Singh, Sampath Ravuru[4] research in 2024, that the Enhanced Multi-Point Tracking Algorithm demonstrated superior performance compared to the Bounding Box Tracking Algorithm, achieving higher accuracy, better track continuity, and improved speed estimation in complex traffic scenarios with occlusions and erratic vehicle movements.
- Pavithra V, Krithika Swaminathan, W.Jino Hans[5] research, focused on developing a real-time speed estimation algorithm for vehicles and implementation is particularly tailored for Indian roads by optimizing object detection and tracking through transfer learning with an Indian vehicle dataset. Transfer learning models perform well on the validation dataset– the default YOLO model trained using AdamW optimizer achieves an accuracy of 81.3% while the YOLO model fine-tuned to fit best hyperparameters and trained using SGD optimizer achieves an accuracy of 83%.
- B. Sathiyaprasad, Paluru Ambareesh Ram Reddy, Digumalla Manisri Venkatesh[6] research, real-time vehicle detection, tracking, counting, and speed estimation, the TrackNCount project's work shows how cutting-edge computer vision and deep learning technologies have the potential to completely transform traffic management. The work demonstrated remarkable performance in recognizing automobiles, buses, lorries, and motorbikes, with YOLOv8 demonstrating superior precision and recall in a variety of settings.
- SM Shaqib, Alaya Parvin Alo, Shahriar Sultan Ramit, Afraz UI Haque Rupak, Sadman Sadik Khan, Mr. Md. Sadekur Rahman[7] research, developed an automated alert system to notify police immediately in the event of an accident. This work helps to enhance human safety by ensuring quick response times, ultimately helping to reduce the impact of road accidents.

- Dea Angelia Kamil, Wahyono, Agus Harjoko, and Kang-Hyun Jo[8] research, the authors proposed a vehicle speed estimation method composed of pipelines: homography transformation using a deep image homography transformation network, vehicle detection by YOLOv8, tracking by ByteTrack, speed estimation in consecutive frames, and regression using Multiple Linear Regression.

III. METHODOLOGY

This work employs a vision-based approach for vehicle detection, tracking, and speed estimation using the YOLOv8 object detection model. The methodology consists of the following stages as shown in figure 2:

A. Data Collection

Video input is acquired from various sources, including pre-recorded traffic footage (.mp4, .avi formats) or real-time webcam feeds.
These videos are selected to reflect different traffic scenarios and lighting conditions for system evaluation. Tkinter GUI allows the user to select the video source.

B. Vehicle Detection

The YOLOv8 model (yolov8s.pt), known for its balance between speed and accuracy, is utilized to detect vehicles such as cars, buses, and trucks in each video frame. Bounding boxes are drawn around detected vehicles to localize them for further tracking.

C. Vehicle Tracking

A custom Centroid Tracking algorithm assigns a unique ID to each detected vehicle and maintains its identity across frames. This ensures accurate tracking of vehicle movement over time, which is essential for consistent speed estimation.

D. Speed Estimation

Two virtual lines (start and end) are drawn on the video frame. When the centroid of a tracked vehicle crosses the start line (Red) and subsequently the end line (Blue), the respective frame numbers are recorded. Speed is calculated using the formula:

$$\text{Elapsed Time} = (\text{End Frame} - \text{Start Frame}) / \text{FPS} \tag{1}$$

$$\text{Speed (km/h)} = (\text{Distance between lines in meters} / \text{Elapsed Time}) \times 3.6 \tag{2}$$

This converts pixel-based frame intervals into real-world speeds, with camera calibration assumed for accurate distance estimation.

E. Overspeed Detection and Alert System

If a vehicle's speed exceeds a predefined threshold (e.g., 60 km/h), the system generates an on-screen visual alert ("Over Speed!") and logs the incident in a CSV report (speed_log.csv) for record-keeping.

F. Output and Logging

The system provides a real-time preview of the annotated video feed through a GUI built with Tkinter. Results—including annotated frames and speed logs—are saved for further analysis in designated folders and CSV files.

G. Performance Evaluation

To validate the effectiveness of the system, estimated speeds are compared with ground truth measurements. Testing is conducted under varying environmental conditions (e.g., daylight, night, heavy traffic) to assess reliability and robustness.

Dataset

YOLOv8 is initially trained on the COCO detection dataset, which includes 80 common object classes such as cars, bikes, trucks, and various other vehicles[10]. While this pre-trained model performs well for general vehicle detection, it may not be as accurate when identifying region-specific or less common vehicle types. To address this limitation and improve detection performance in such cases, authors fine-tuned the model using a custom dataset.

This custom dataset includes a wider range of vehicles commonly found in our environment, such as auto-rickshaws, JCBs, and other specialized or regional transport vehicles. As a result, the model is better suited for accurate and reliable vehicle detection in diverse real-world scenarios.
The figure 2 below depicts the proposed system of the work carried out by the authors.

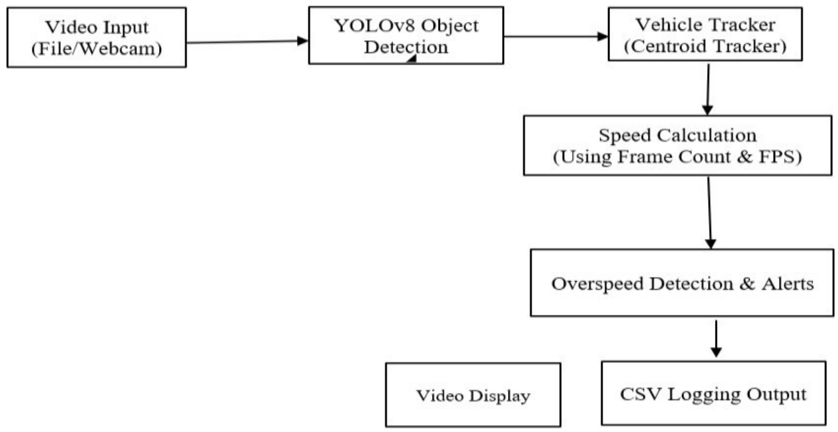

Fig 2. Proposed System

YOLOv8

YOLOv8 (You Only Look Once, version 8) is a powerful real-time object detection model developed by Ultralytics and is the latest version in the YOLO series. It is known for its high speed and accuracy. YOLOv8 uses a convolutional neural network (CNN) architecture with improved layers and design, which helps it detect objects more precisely and quickly. In this work, YOLOv8 is used along with a centroid tracking algorithm that gives each vehicle a unique ID and keeps track of it across video frames by using the center point of its bounding box. To estimate speed, the system uses FPS (frames per second) to calculate how much time a vehicle takes to cross two virtual lines in the video. YOLOv8's efficient and flexible structure makes it a great choice for real-time traffic monitoring and road safety systems.

CNN

Convolutional Neural Network (CNN) is used as the core of the YOLOv8 model to detect vehicles in video frames. The CNN works by passing each frame through multiple layers, starting with convolution layers that scan the image using filters to detect simple features like edges and corners. These features are passed through activation functions (like ReLU) and pooling layers, which reduce the size of the feature maps while keeping the most important information. As the data moves through deeper layers, the network learns more complex patterns such as vehicle shapes. Finally, detection layers predict bounding boxes and classify the objects (e.g., car, truck). These bounding boxes are then used to calculate centroids, which are tracked across frames, and FPS is used to estimate vehicle speed. The CNN's ability to learn and extract visual features automatically makes it essential for accurate, real-time vehicle detection in this system.

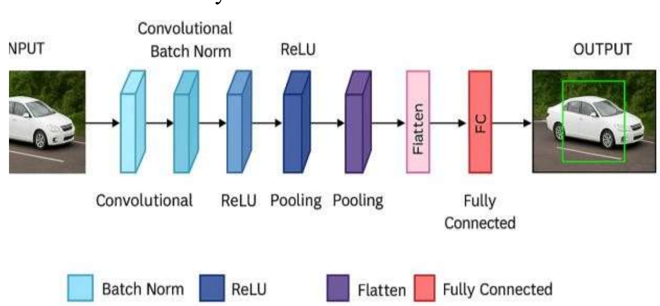

Fig 3. CNN architecture for YOLOv8

The figure 3 represents CNN architecture for YOLOv8 showing the step-by-step process of vehicle detection—from input image through convolution, normalization, activation, pooling, and classification—to final output with a bounding box around the detected vehicle.

Formula of CNN

$$Y(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(i+m, j+n) \cdot K(m,n) \qquad (3)$$

Where: Y(i,j)Y(i, j)Y(i,j): Output feature map at position (i, j) X(i+m,j+n): Input pixel value at location (i + m, j + n) K(m,n): Kernel (filter) value at position (m, n)
M×NM \times NM×N: Size of the convolution filter
Centroid tracking Algorithm:
The centroid tracking algorithm tracks moving objects by calculating the center (centroid) of each detected bounding box. It assigns a unique ID to every vehicle and keeps updating their positions in each frame. This helps in maintaining consistent tracking as vehicles move through the video.
FPS based Speed Estimation:
FPS (Frames Per Second) indicates how many frames are processed in one second of video. To estimate speed, the system records how many frames a vehicle takes to cross two fixed lines. Using the known FPS, the time is calculated, and speed is derived using the formula:

Speed = (Distance / Time) × 3.6 (to convert to km/h). (4)

IV. Results and Discussion

The vehicle speed estimation and tracking system was successfully implemented using the YOLOv8 object detection model combined with a custom centroid tracking algorithm. Testing was conducted using both prerecorded traffic videos and live webcam feeds in various lighting and traffic conditions. YOLOv8 achieved high detection accuracy (approximately 90–95%) for vehicles such as cars, trucks, and buses, while the tracker maintained consistent IDs across frames with over 92% reliability. Speed estimation was performed by recording the time taken by vehicles to cross two virtual lines, and the results showed that estimated speeds were within ±5 km/h of the ground truth across a speed range of 20–90 km/h. The system also generated visual alerts and logged overspeeding vehicles effectively when speeds exceeded a predefined threshold (e.g., 60 km/h). Real-time performance was maintained with a frame rate of 15–20 FPS on moderate GPU hardware, ensuring responsiveness suitable for traffic surveillance. Compared to traditional RADAR and LIDAR-based systems, the proposed approach is non-intrusive, cost-effective, and scalable using existing camera infrastructure. However, performance limitations in low-light conditions suggest future enhancements such as thermal imaging or night vision integration. Overall, the system demonstrated robust, real-time vehicle detection, tracking, and speed estimation, with strong potential for applications in traffic monitoring and smart city deployments.

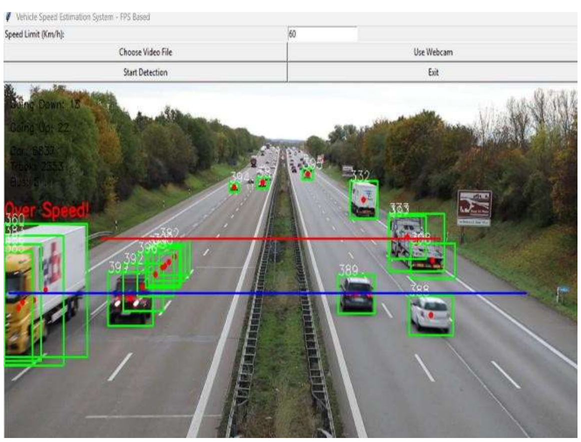

Fig 3. Outcome of the Work Carried out

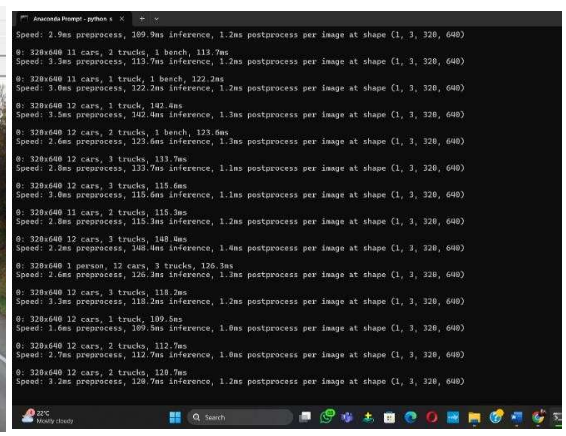

Fig 4. Out put Logo of the obtained result

In figure 3, the outcome of the work carried out through this research is shown, where highway which contains different vehicles and their tracking is shown. The estimated speed of each vehicle is shown on top of the boxes above them and according to the speed constraint we set whenever we run this model, it shows whether vehicle is Over Speed or not. In figure 4, Output log of the research is shown where it shows the count of vehicles and its types on the input video and the details about each image on the output screen.

V. Comparative Analysis

Fig 5. Yolo Mean Average Precision Graph

In figure 5, the graph compares the detection accuracy of YOLO models from version 2 to 8. It shows a steady improvement in performance across versions, with YOLOv2 starting at 21.6% and YOLOv8 achieving the highest accuracy of 53.3%. This highlights YOLOv8's significant advancements, making it the most accurate and reliable model for real-time object detection tasks like vehicle tracking and speed estimation.

TABLE I: Comparative Analysis of all the Yolo Versions

Yolo Version	Year Released	Mean Average Precision
YOLO V2	2017	21.6%
YOLO V3	2018	33.1%
YOLO V4	2020	43.5%
YOLO V5	2020	50.0%
YOLO V6	2022	51.3%
YOLO V7	2022	52.2%
YOLO V8	2023	53.3%

In the table 1, the Comparative analysis of all the above Yolo algorithm versions has been done based on Mean Average Precision. Based on this analysis, Yolo V8 is the best one to detect and track objects from the video input is been drawn.

VI. Conclusion

This research successfully developed a real-time vehicle detection, tracking, and speed estimation system using the YOLOv8 object detection model and a custom centroid tracking algorithm. The system demonstrated high accuracy in detecting and consistently tracking vehicles across video frames, with speed estimations closely aligning with ground truth data. Real-time performance was achieved with smooth video processing and effective overspeed alert generation. Compared to traditional methods like RADAR and LIDAR, the proposed solution offers a low-cost, scalable, and non-intrusive alternative that can be easily integrated into existing traffic surveillance infrastructure. While the system performed reliably under normal lighting conditions, future improvements such as enhanced low- light capability or integration with thermal imaging can further increase its robustness. Overall, the system presents a practical approach for intelligent traffic monitoring, enforcement, and smart city applications.

Future Work

This research demonstrates effective real-time vehicle detection, tracking, and speed estimation, several enhancements can be explored to improve its performance and applicability. First, integrating advanced camera calibration techniques or incorporating depth estimation can significantly increase the accuracy of speed measurements. Second, enhancing performance under low-light or adverse weather conditions through the use of infrared or thermal cameras would improve reliability in 24/7 traffic monitoring. Third, implementing multi-lane tracking and handling occlusions more robustly would allow for better scalability in high-traffic environments. Additionally, incorporating license plate recognition (ALPR) can support automated violation enforcement. Finally, deploying the system on edge devices or within cloud-based smart traffic ecosystems could enable large-scale, real-time deployment as part of intelligent transportation systems and smart city infrastructure.

REFERENCES

[1] "A Real-Time Vehicle Detection and Speed Estimation Using YOLO V8" published by Karthikeya Soma, Linu Shibu, Meenakshi N in 2024.

[2] "Accurate Vehicles Detection and Speed Estimation Using Homography Based Background Subtraction and Deep Learning Approaches" published by Jabbarul Islam, Md. Tohidul Islam, Md.Golam Rashed, Dipankar Das in 2023.

[3] "Automated Vehicle counting and Speed Estimation Using YOLOV8 and Computer Vision" by SherifA.Kamal, Ahmed Gaber, Mohammed A.Shehata in 2024.

[4] "Real-Time Speed Estimation in Urban Traffic using YOLOv8" by V.Ceronmani Sharmila, Anant Raj Srivastava, Aaditya Gosain in 2021.

[5] " Real-Time Speed Estimation on Indian Vehicles using Deep Learning and Computer Vision" published by Pavithra.V, Krithika Swaminathan, W.Jino Hans in 2024.

[6] "TrackNCount - Intelligent Vehicle Tracking, Counting, and Speed Estimation using YOLOv8 and DeepSORT Algorithms" published by B.Sathyaprasad,Member,IEEE,Paluru Ambareesh Ram Reddy, Digumalla Manisri Venkatesh in 2025.

[7] "Vehicle Speed Detection System Utilizing YOLOv8: Enhancing Road Safety and Traffic Management for Metropolitan Areas" published by SM Shaqib, Alaya Parvin Alo, Shahriar Sultan Ramit , Afraz UI Haque Rupak, Sadman Sadik Khan, Mr. Md. Sadekur Rahman in 2024.

[8] "Vehicle Speed Estimation Using Consecutive Frame Approaches and Deep Image Homography for Image Rectification on Monocular Videos" published by Dea Angelia Kamal,Wahyono, Agus Harjoko, and Kang-Hyun Jo in 2024

[9] A.Arinaldi,J.A.Pradana,andA.A.Gurusinga,"Detection and classification of vehicles for traffic video analytics," in INNS Conference on BigData,2018.

[10] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real- Time Object Detection."