IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13537

Vehicle Classification and Traffic Density Analysis using RT-DETR

Prajwal.M[1], Sujayeendra Rao[2], Dr. Ananth. G. S[3]

PG Student, The National Institute of Engineering, Mysuru, Visveswaraya Technological University, Belagavi, Karnataka, India[1]

Faculty, The National Institute of Engineering, Mysuru, Visveswaraya Technological University, Belagavi, Karnataka, India[2]

Head of the Department (HOD), The National Institute of Engineering, Mysuru, Visveswaraya Technological University, Belagavi, Karnataka, India[3]

Abstract: Traffic congestion and vehicle monitoring have become major challenges in modern urban transportation systems, requiring efficient real-time traffic analysis and management solutions. This project presents an intelligent Vehicle Classification and Traffic Density Analysis System using the advanced RT-DETR deep learning architecture for fast and accurate vehicle detection in traffic environments. Unlike traditional CNN-based models, RT-DETR uses transformer-based attention mechanisms to improve detection accuracy in crowded and complex road conditions. The system can identify multiple vehicle categories such as cars, buses, trucks, motorcycles, bicycles, and auto-rickshaws from images and videos. A dual-platform deployment using Flask and Streamlit supports both testing and large-scale monitoring applications. The application includes secure user authentication, traffic monitoring dashboards, and automated traffic reporting features. Uploaded traffic videos are processed using frame-based inference techniques to estimate traffic density levels such as Low, Medium, and High. Confidence-based filtering helps reduce false detections caused by occlusion, lighting variations, and dense traffic conditions. The system utilizes OpenCV and PyTorch for efficient image and video processing in real-time environments. Experimental results show that the RT-DETR model provides high detection accuracy, stable performance, and an effective AI-driven solution for smart traffic monitoring and intelligent transportation systems.

Keywords: Vehicle Classification, Traffic Density Analysis, RT-DETR, Deep Learning, Computer Vision, Intelligent Transportation System, Object Detection, Real-Time Monitoring, Flask, Streamlit, OpenCV, PyTorch, Traffic Surveillance, Smart City, Vehicle Detection.

I.INTRODUCTION

Rapid urbanization and the increasing number of vehicles on roads have significantly increased traffic congestion, road accidents, and transportation management challenges across cities worldwide. Traditional traffic monitoring systems mainly depend on manual observation, roadside sensors, or fixed surveillance systems, which are often expensive, less scalable, and incapable of providing accurate real-time traffic analysis under dynamic road conditions.Modern intelligent transportation systems require automated solutions capable of continuously monitoring traffic flow, identifying vehicle categories, and estimating traffic density with high accuracy and low latency. Recent advancements in Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning have enabled the development of advanced computer vision systems capable of analyzing real-world traffic scenes efficiently.

This project focuses on the implementation of a Vehicle Classification and Traffic Density Analysis System using the RT-DETR (Real-Time Detection Transformer) architecture. RT-DETR is a transformer-based object detection model that combines real-time inference speed with accurate object localization and classification. Unlike traditional CNN-based detectors, RT-DETR uses attention mechanisms to understand global relationships between objects within traffic scenes, improving detection performance in crowded and complex environments.The system is designed to detect and classify multiple vehicle categories including cars, buses, trucks, motorcycles, bicycles, and auto-rickshaws from traffic images and videos. Additionally, the proposed framework estimates traffic density levels based on detected vehicle counts, enabling efficient traffic monitoring and management.The developed application supports real-time image and video processing through Flask-based deployment integrated with OpenCV and PyTorch frameworks. The system also provides a user-friendly dashboard for viewing traffic analysis results, vehicle counts, and density reports.Overall, this project aims to provide a scalable, accurate, and intelligent traffic monitoring solution capable of supporting smart city infrastructure, automated traffic control systems, highway surveillance, and urban transportation management.

.

FUNCTIONAL REQUIREMENTS: The functional requirements define the core operations, services, and functionalities that the Vehicle Classification and Traffic Density Analysis System must perform to ensure accurate vehicle monitoring and efficient traffic analysis in real-time environments.

- **User Authentication and Session Management:**The system shall provide secure user registration and login functionality using encrypted password storage mechanisms to protect user credentials and sensitive information. User sessions must be maintained securely to prevent unauthorized access to traffic monitoring dashboards, detection reports, and historical traffic analysis records. The application shall also support session timeout handling and secure logout functionality to improve overall system security and reliability.

- **Vehicle Detection and Classification:**The system shall detect and classify vehicles from uploaded traffic images and videos using the RT-DETR deep learning model. Supported vehicle categories include:**Car,** ***Bus ,Truck,Motorcycle, Bicycle,Auto-rickshaw.*** The system must generate confidence scores for each detected object and display bounding boxes around detected vehicles for visual analysis. The model should accurately identify multiple vehicles simultaneously, even in dense traffic conditions, varying lighting environments, and partially occluded scenes.

- **Traffic Density Analysis:**The application shall calculate traffic density levels based on the number of detected vehicles within a frame. Traffic conditions shall be categorized into: ***High Traffic ,Low Traffic,Medium Traffic.***Traffic density estimation must be supported for both static image analysis and real-time video processing. The system shall continuously update traffic density information for video streams to assist traffic monitoring authorities in identifying congestion-prone areas efficiently.

- **Video Frame Processing:**The system shall process uploaded traffic videos using optimized frame sampling techniques to reduce computational overhead and improve inference speed during real-time analysis. Frames shall be extracted at predefined intervals and analyzed individually to maintain a balance between detection accuracy and processing performance. The system should support common video formats such as MP4 and AVI.

- **Detection Logging and Dashboard:**All detection results including timestamp, uploaded file name, detected vehicle classes, vehicle count, confidence score, and traffic density level shall be stored in a SQLite database for future analysis and reporting purposes. An administrative dashboard shall display real-time traffic statistics, historical detection logs, vehicle count summaries, and traffic density trends using graphical representations and tabular formats for easier interpretation.

- **Automated Reporting and Alerts:**The application shall automatically generate traffic monitoring summaries and alerts whenever traffic congestion exceeds predefined threshold values. Notifications can be used to inform traffic management authorities about heavy congestion conditions, unusual traffic patterns, or traffic bottlenecks. The reporting module shall support daily and weekly traffic analysis summaries for efficient transportation planning.

- **Image and Video Upload Validation:**The system shall validate uploaded media files before processing to ensure only supported image and video formats are accepted. Invalid or corrupted files must be rejected with appropriate error messages to maintain system stability and security.

- **Real-Time Visualization:**The application shall provide real-time visualization of detected vehicles through bounding boxes, labels, confidence percentages, and traffic density indicators on processed frames. This functionality helps users easily understand traffic conditions and detection outcomes.

- **Database Management:**The system shall maintain organized storage of user information, traffic analysis logs, and processed media details using a structured SQLite database. The database shall support efficient retrieval of historical traffic monitoring records and analysis reports.

- **Scalability Support:**The software architecture shall support future enhancements such as live CCTV integration, cloud deployment, smart traffic signal control, and edge-device-based traffic monitoring systems without major modifications to the core framework.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13537

NON-FUNCTIONAL REQUIREMENTS: Non-functional requirements define the quality attributes, operational standards, and performance expectations of the Vehicle Classification and Traffic Density Analysis System. These requirements ensure that the system performs efficiently, accurately, securely, and reliably in real-world traffic monitoring environments.

- **Performance:**The system shall provide near real-time vehicle detection and traffic analysis with minimal latency during image and video processing. High-resolution traffic images must be processed within a few seconds to ensure rapid analysis and response. For video inputs, optimized frame sampling and efficient inference mechanisms shall reduce computational overhead while maintaining smooth and continuous processing performance. The system should efficiently utilize GPU or CPU resources for faster execution of the RT-DETR model.
- **Accuracy:**The RT-DETR model shall maintain high vehicle detection and classification accuracy under varying environmental conditions such as poor lighting, shadows, rain, fog, occlusions, and dense traffic scenarios. The system must minimize false detections and accurately distinguish between multiple vehicle categories including cars, buses, trucks, motorcycles, bicycles, and auto-rickshaws. Confidence-based filtering shall improve prediction reliability during real-time monitoring.

- **Reliability:**The application shall ensure stable and continuous operation during long-duration traffic monitoring without unexpected failures or crashes. Proper exception handling and error recovery mechanisms shall be implemented to handle invalid uploads, unsupported formats, and runtime processing issues efficiently. Detection logs and traffic analysis reports shall be securely stored in the database to ensure data persistence even during unexpected server interruptions or application restarts.

- **Usability:**The user interface shall provide a simple, user-friendly, and intuitive design that allows operators to upload traffic images or videos, view vehicle detection results, and analyze traffic density easily without requiring advanced technical knowledge. The dashboard shall present vehicle counts, traffic density indicators, and graphical reports in a visually understandable format. Color-coded indicators and clear labels shall help users quickly interpret traffic conditions.

- **Scalability:**The system architecture shall support future deployment on cloud platforms and integration with smart city traffic surveillance infrastructure. The modular design shall allow future enhancements such as live CCTV streaming, drone-based monitoring, automatic traffic signal management, edge-device deployment, and distributed traffic analysis systems without major modifications to the existing framework.

- **Security:**The application shall implement secure user authentication, encrypted password storage, server-side validation, and protected database access to prevent unauthorized usage and malicious activities. Uploaded files shall be validated to ensure only supported image and video formats are accepted. Administrative functionalities and sensitive traffic reports shall be accessible only to authorized users through role-based access control mechanisms.

- **Maintainability:**The project shall follow modular coding practices using the Model-View-Controller (MVC) architecture to simplify debugging, future upgrades, and model replacement. Proper documentation, standard naming conventions, and well-organized source code shall improve readability and ease of maintenance for future developers. The AI inference module shall remain independent from the frontend interface, allowing easy integration of updated object detection models in the future.

- **Availability:**The system shall remain available for continuous traffic monitoring operations with minimal downtime. The application should support uninterrupted analysis for both short-term and long-duration traffic surveillance tasks.

- **Compatibility:**The application shall support execution across multiple operating systems such as Windows and Linux and shall be compatible with commonly used web browsers. The system should also support standard traffic image and video formats including JPG, PNG, MP4, and AVI.

II METHODOLOGY

The methodology of the Vehicle Classification and Traffic Density Analysis System is designed to perform real-time vehicle detection and traffic monitoring using deep learning and computer vision techniques. Initially, traffic images and video datasets containing different vehicle categories such as cars, buses, trucks, and bikes are collected and preprocessed using resizing, normalization, and augmentation techniques to improve model performance. The RT-DETR deep learning model is then trained using labeled datasets to accurately classify and detect vehicles under different road and environmental conditions. OpenCV is integrated with the trained model to process traffic videos and perform frame-by-frame vehicle detection efficiently. Frame sampling techniques are used to reduce computational complexity and maintain stable real-time performance. The Flask backend manages file uploads, model execution, and user interaction, while SQLite database integration is used to securely store vehicle detection and traffic density records. The system calculates traffic density based on the number of detected vehicles in each frame and displays the results through an interactive dashboard. Performance evaluation is carried out using accuracy, precision, recall, F1-score, and confusion matrix analysis to measure the effectiveness of the system. Experimental testing confirmed that the proposed system provides reliable, efficient, and accurate vehicle classification and traffic density analysis for intelligent transportation applications.

A. *Data Preprocessing:* Traffic images and video frames are first collected from multiple road environments containing different lighting conditions, vehicle densities, weather conditions, and camera angles. The collected data is resized and normalized according to the input requirements of the RT-DETR model to maintain consistency during training and inference. Data augmentation techniques such as horizontal flipping, brightness adjustment, rotation, scaling, random cropping, and contrast enhancement are applied to improve model robustness and generalization capability. Video files are processed frame-by-frame using OpenCV, and important frames are extracted at regular intervals for efficient analysis without increasing computational complexity.

B. *Model Selection:* The project utilizes the RT-DETR architecture because of its superior real-time object detection performance and transformer-based attention mechanisms. Unlike traditional CNN-based detectors, RT-DETR captures global contextual relationships between multiple objects in traffic scenes, enabling accurate detection even in dense and crowded road conditions. The model provides faster inference speed while maintaining high detection precision, making it highly suitable for intelligent transportation systems and real-time traffic monitoring applications. RT-DETR also performs better in handling partially occluded vehicles and varying environmental conditions compared to conventional object detection models.

C. *Training and Validation:* Transfer learning is employed using pretrained RT-DETR weights to reduce training time and improve model convergence. The traffic dataset is divided into 80% training data and 20% validation data to evaluate model performance on unseen traffic scenes. The model is trained using the AdamW optimizer with dynamic learning rate scheduling to optimize convergence and minimize overfitting issues. During training, backpropagation is used to update model parameters and improve detection accuracy for various vehicle classes such as cars, buses, trucks, motorcycles, bicycles, and auto-rickshaws. Validation accuracy and loss values are continuously monitored throughout the training process to select the best-performing model.

D. *Model Evaluation:* The trained RT-DETR model is evaluated using performance metrics such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix to measure vehicle detection and classification efficiency. These metrics help analyze the model's ability to correctly identify different vehicle categories under various traffic conditions. The confusion matrix is used to study correct predictions and misclassifications between vehicle classes in dense traffic scenarios. The evaluation results confirm that the model provides reliable, accurate, and stable performance for real-time traffic monitoring applications.

E. *Deployment:*The trained RT-DETR model is integrated into a Flask-based web application to provide a practical and user-friendly traffic monitoring platform. OpenCV is used for image preprocessing, video frame extraction, and real-time visualization, while PyTorch handles deep learning inference operations. The deployment architecture supports both static image analysis and real-time video processing with efficient frame sampling techniques. A lightweight Streamlit interface is also used for rapid testing and demonstration purposes. The system provides real-time vehicle detection results, vehicle counts, and traffic density analysis through an interactive dashboard.

F. *Sequence Diagram Explanation:* The sequence diagram illustrates the complete workflow of the system where users upload traffic images or videos through the frontend interface. The Flask backend validates uploaded files and preprocesses the media before passing it to the RT-DETR inference engine. The AI

DOI: 10.17148/IARJSET.2026.13537

model processes the input data, detects vehicles, classifies vehicle categories, and generates confidence scores for each detection. Based on the detected vehicle count, the system calculates traffic density levels such as Low, Medium, or High traffic conditions. Detection results including timestamps, file names, detected vehicle classes, confidence scores, and traffic density levels are stored in the SQLite database and displayed on the dashboard for monitoring and analysis.

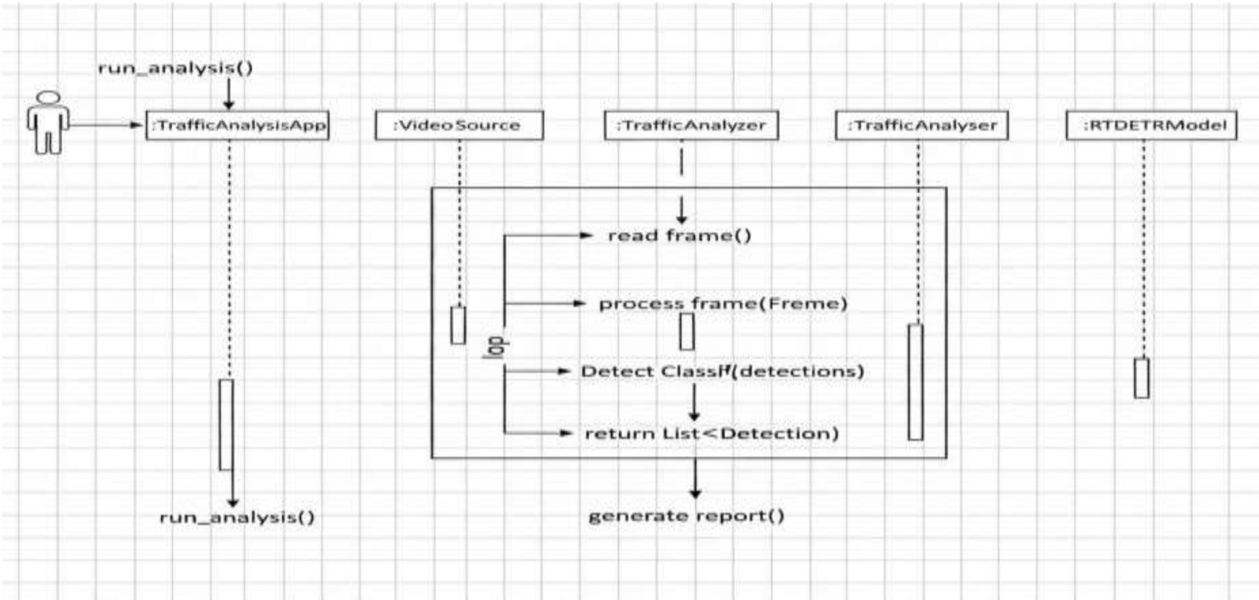

Fig.No.1:Sequence Diagram.

III. SYSTEM TESTING

System testing for the Vehicle Classification and Traffic Density Analysis System was performed to verify the proper integration between the Flask backend, SQLite database, OpenCV video processing module, and the RT-DETR deep learning model. Various traffic images and video formats were tested to ensure accurate vehicle detection, smooth file handling, and efficient traffic density analysis under different road conditions. The video processing module was stress-tested using frame sampling techniques to confirm stable real-time performance without significant latency or system crashes. Additional testing was conducted on user authentication, file validation, database storage, and dashboard functionalities to ensure secure access, reliable data logging, and smooth user interaction. The testing results confirmed that the system operates efficiently, accurately, and reliably for practical real-time traffic monitoring applications.

Testing Methods Used:

- The testing methodology for the Vehicle Classification and Traffic Density Analysis System is based on a combination of integration testing, model evaluation, and multi-level validation checks to ensure accuracy, reliability, and security. Initially, integration testing is performed to verify smooth data flow between the Flask backend, SQLite database, OpenCV processing module, and the RT-DETR AI model, including testing with different traffic image and video formats. The video processing module is stress-tested to confirm efficient frame sampling and real-time vehicle detection without latency issues while ensuring correct traffic density logging and dashboard updates.

- In addition, model evaluation testing is conducted using validation datasets (20% split) where performance metrics such as accuracy, precision, recall, and F1-score are calculated by comparing predicted vehicle classes with actual labels. The system also performs epoch-based validation to automatically select the best-performing model and reduce overfitting during training. Confusion matrix analysis is used to study misclassifications between vehicle categories under dense traffic and varying environmental conditions.

- Finally, validation testing is implemented at multiple levels including file type validation, supported media format checks, user authentication testing, database validation, and client-side verification to ensure secure uploads, proper data handling, reliable execution, and stable system performance during continuous traffic monitoring operations.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13537

Table 1: Test Case Design

Test Case ID	Description of Test Case	Input	Expected Output	Status
TC01	Image upload validation	Valid image (.jpg/.png)	Image accepted and processed successfully	Pass
TC02	Invalid file format handling	Unsupported file (.txt)	System rejects file with error message	Pass
TC03	Vehicle classification	Traffic image containing vehicles	Output shows detected vehicle classes with confidence scores	Pass
TC04	Video processing	Valid video (.mp4/.avi)	Video processed using frame sampling	Pass
TC05	Traffic density analysis	Congested traffic video	System displays traffic density level correctly	Pass

The above table presents the test cases designed to validate the core functionalities of the Vehicle Classification and Traffic Density Analysis System using Deep Learning. Each test case evaluates key components such as image/video input handling, file validation, AI-based vehicle classification, video frame processing, traffic density estimation, database logging, and dashboard functionality. The successful execution of all test cases confirms that the system operates efficiently, performs accurate vehicle detection, and provides reliable real-time traffic monitoring.

Model Evaluation Metrics and Formulas: *The system's predictive performance was evaluated using the following metrics:*

- *Accuracy measures the proportion of correct predictions:*

Accuracy*=(TP+TN)/(TP+TN+FP+FN)*

- *Precision indicates the correctness of predicted career recommendations:*

Precision*=TP/(TP+FP)*

- *Recall measures the system's ability to identify all relevant career categories:*

Recall*=TP/(TP+FN) *

- *F1-score provides a balance between precision and recall:*

F1-score*=2((Precision).(Recall)/(Precision+Recall))*

A confusion matrix was used to visualize classification results and analyze misclassifications between vehicle classes, helping evaluate the real-world reliability of the Vehicle Classification and Traffic Density Analysis System.

The system also incorporates strong error handling and input validation mechanisms to detect invalid image formats, missing files, and inconsistent inputs. Validation checks ensure only supported image and video types are processed, while clear feedback messages help users correct errors before processing. Security and data integrity testing confirmed that uploaded traffic images and videos are processed securely during runtime and are not permanently stored without authorization.

Overall, testing confirmed that the Vehicle Classification and Traffic Density Analysis System satisfies all functional and non-functional requirements by demonstrating high accuracy, efficiency, reliability, usability, and secure operation, making it suitable for practical real-time traffic monitoring and smart transportation applications.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2026

DOI: 10.17148/IARJSET.2026.13537

IV. DICUSSION AND RESULT

The results obtained from the Vehicle Classification and Traffic Density Analysis System demonstrate the effectiveness of deep learning techniques in detecting and classifying vehicles from traffic images and video streams. Multiple object detection approaches were studied during system development, and the RT-DETR (Real-Time Detection Transformer) model was selected due to its high accuracy, efficient object detection capability, and strong real-time performance.The trained RT-DETR model was evaluated using an unseen test dataset created through an 80:20 train–test split. The evaluation focused on accuracy, precision, recall, F1-score, and confusion matrix analysis to assess both overall and class-wise vehicle detection performance.

A. *Accuracy Evaluation:* Accuracy represents the proportion of correctly predicted image classes out of the total predictions made by the model and is calculated as: ***Accuracy****=(TP+TN)/(TP+TN+FP+FN).* In the context of this project, accuracy serves as a high-level indicator of the overall effectiveness of the forest fire detection system. Forest fire detection is a multi-class image classification problem where classes such as Fire, Smoke, and Normal scenes may contain visually overlapping environmental features such as fog, sunlight, clouds, and dust. Due to these complexities, achieving perfect accuracy is challenging in real-world forest environments. The obtained accuracy reflects realistic system behavior when processing diverse and dynamic environmental conditions. Unlike simple binary image classification tasks, forest fire detection systems must handle ambiguous visual patterns and varying lighting conditions, making high practical accuracy both significant and acceptable.

Table 2:Classification Report of the RT-DETR Model

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Car	0.96	0.95	0.95	120
Bus	0.93	0.92	0.92	90
Truck	0.94	0.93	0.93	80
Bike	0.91	0.90	0.90	70
Accuracy			0.93	360
Macro AVG	0.93	0.92	0.92	360
Weighted AVG	0.93	0.93	0.93	360

B. *Classifiaction Result Analysis:* The classification report provides a class-wise performance evaluation of the **SigLIP model using precision, recall, F1-score**, and support for each detection category.

- **Precision Analysis:** Precision measures how many of the fire or smoke detections predicted by the system are actually correct: **Precision**=TP/(TP+FP). High precision values were observed for Fire and Normal image classes, indicating that when the system predicts a fire event, the prediction is highly reliable. Slightly lower precision for Smoke detection occurs because smoke patterns may visually resemble fog, haze, or cloud formations in certain environmental conditions.
- **Recall Analysis:** Recall measures the system's ability to correctly identify all relevant fire instances: **Recall**=TP/(TP+FN). High recall values for Fire detection indicate that the model successfully identifies most actual fire occurrences from the dataset. Lower recall values for smoke-related classes may result from variations in smoke density, lighting conditions, and image quality rather than model inefficiency.
- **F1-Score Analysis:**The F1-score balances precision and recall and is calculated as:**F1-score**=2 ((Precision).(Recall)/(Precision+Recall)). The obtained F1-scores indicate balanced and realistic performance for real-world forest monitoring applications where exact separation between smoke, fire, and normal environmental conditions is not always possible. The weighted average F1-score confirms consistent overall detection performance across the major image classes used in the system.

C. *Confusion Matrix Analysis:* A confusion matrix is a performance evaluation tool used to compare actual and predicted vehicle class labels generated by the trained RT-DETR model. It visually represents correct classifications and misclassifications for different vehicle categories. The diagonal values indicate correct predictions, while off-diagonal values represent classification errors between visually similar vehicles.The confusion matrix helps evaluate model reliability, vehicle detection accuracy, and overall system performance in real-time traffic monitoring applications. It also assists in identifying model weaknesses and improving classification performance through further training and optimization.

International Advanced Research Journal in Science, Engineering and Technology
Impact Factor 8.311 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 13, Issue 5, May 2026
DOI: 10.17148/IARJSET.2026.13537

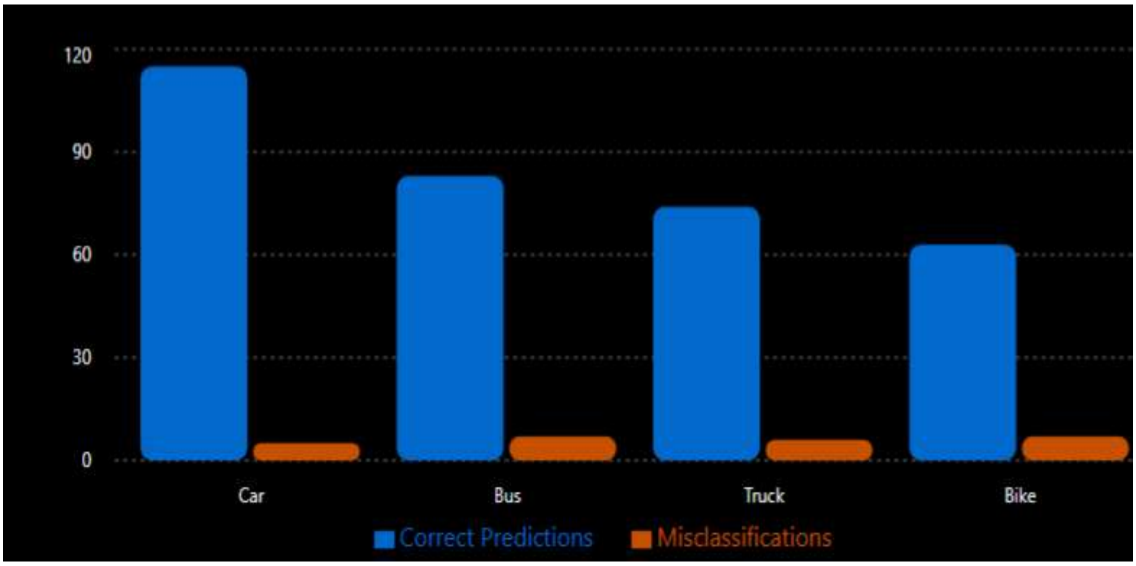

Fig.No.2: Confusion Matrix of the Trained RT-DETR Model

Key Observations

- Most vehicle categories show strong diagonal dominance, indicating high correct classification rates by the RT-DETR model.
- Minor misclassifications occur primarily between visually similar vehicle types such as:
  - Cars and vans
  - Bikes and scooters
  - Trucks and buses in crowded traffic scenes
- These errors arise due to similarities in vehicle shape, occlusion, lighting conditions, shadows, and varying camera angles.

Importantly, the confusion matrix shows that the misclassifications are visually reasonable, meaning the system does not produce random or completely incorrect detections.

Justification for the Obtained Accuracy:

The obtained accuracy is technically valid and justified due to the following reasons:

- Multi-class vehicle classification complexity significantly increases detection difficulty.
- Real-world traffic environments contain varying lighting conditions, shadows, weather changes, and motion blur.
- Dense traffic causes vehicle overlap and partial occlusion.
- Small vehicles such as bikes are harder to detect accurately in congested roads.
- The RT-DETR model performs contextual object understanding rather than simple feature matching, making predictions more robust and practical for real-world deployment.

Overall, the confusion matrix and classification results confirm that the trained RT-DETR model provides reliable, stable, and practically acceptable performance for real-time traffic monitoring and intelligent transportation applications.The experimental results confirm that the Vehicle Classification and Traffic Density Analysis System effectively analyzes traffic images and video frames to accurately detect and classify vehicles in real-time environments. The system demonstrated strong classification performance while maintaining efficient processing speed and reliable traffic density estimation. Although minor misclassifications occurred due to visually similar vehicle categories and dense traffic conditions, the obtained accuracy is technically justified because of the complex and dynamic nature of real-world road environments.

International Advanced Research Journal in Science, Engineering and Technology
Impact Factor 8.311 ⁝ Peer-reviewed & Refereed journal ⁝ Vol. 13, Issue 5, May 2026
DOI: 10.17148/IARJSET.2026.13537

Fig. No .3: Login Interface

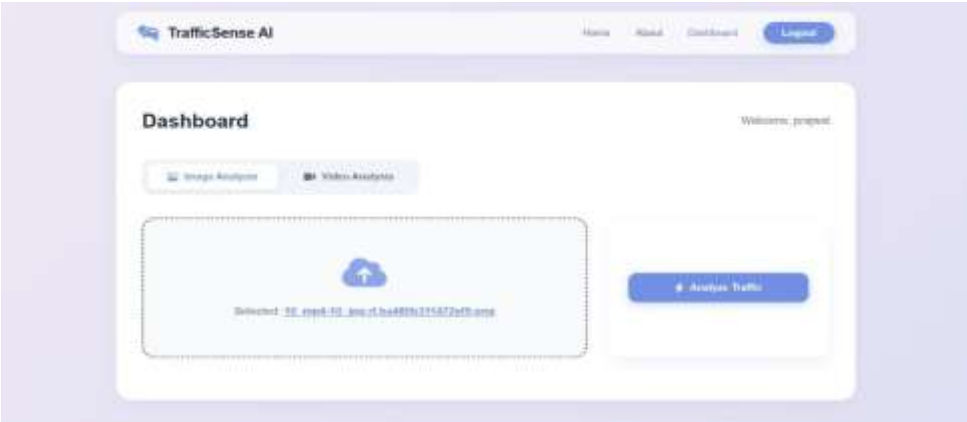

Fig . No .4:Image or Video Uploading Interface

Fig . No .5:Result

V CONCLUSION

The Vehicle Classification and Traffic Density Analysis System successfully demonstrates the application of deep learning for intelligent traffic monitoring and vehicle analysis. The RT-DETR model accurately detects and classifies vehicles from real-time traffic images and videos while estimating traffic density efficiently. Experimental evaluation through classification reports and confusion matrix analysis confirmed strong detection accuracy with only minor misclassifications caused by visually similar vehicle categories and dense traffic conditions.The system also

DOI: 10.17148/IARJSET.2026.13537

incorporates secure data handling, input validation, database integration, and reliable dashboard functionality, making it suitable for practical deployment in smart transportation and traffic management applications. Overall, the project proves that deep learning-based vehicle monitoring systems can significantly support intelligent traffic control, congestion analysis, urban planning, and road safety improvement, with future scope for enhancements such as live CCTV integration, IoT-based traffic sensors, cloud analytics, and smart city deployment.

ACKNOWLEDGMENT

I extend my deepest gratitude to Assistant Professor **Sujayeendra Rao**, my project guide, for his unwavering guidance and support throughout this endeavour. I also express my sincere gratitude to **Dr.Ananth.G.S**, Head of the Department, for his valuable encouragement and support during the successful completion of this project. My sincere thanks also go to the dedicated faculty and staff at **The National Institute of Engineering,Mysuru,** for their in valuable resources and assistance. I am profoundly thankful to my friends and classmates for their camaraderie and encouragement, as well as to my parents for their steadfast backing. Finally, I acknowledge with appreciation everyone who contributed directly or indirectly to the successful completion of this project

REFERENCES

[1]. Wang C, Zhang Y, Liu H. Vehicle detection and classification using deep learning techniques for intelligent traffic monitoring. International Journal of Computer Vision Applications. 2023. Available from: IEEE Xplore

[2]. Sharma R, Patel K, Verma S. Real-time traffic density analysis using convolutional neural networks and OpenCV. International Journal of Advanced Computer Science and Applications (IJACSA). 2022. Available from: IJACSA

[3]. Li X, Chen Y, Zhao H. Deep learning-based vehicle classification for smart transportation systems. IEEE Access. 2021. Available from: IEEE Xplore

[4]. Kumar A, Reddy P, Joshi V. Vehicle detection and traffic monitoring using transfer learning and deep neural networks. International Journal of Engineering Research & Technology (IJERT). 2024. Available from: IJERT

[5]. Zhang T, Wu H, Chen L. Traffic surveillance and vehicle detection using deep convolutional neural networks. Procedia Computer Science. 2020;167:2101–2110. Available from: ScienceDirect

[6]. Zhao Y, Liu X, Wang J. Real-time object detection using RT-DETR for intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems. 2024. Available from: IEEE Xplore

[7]. Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems (NeurIPS). 2019. Available from: PyTorch

[8]. Wolf T, Debut L, Sanh V, et al. Hugging Face Transformers: State-of-the-art natural language processing and deep learning models. 2020. Available from: Transformers Documentation

[9]. Bradski G. The OpenCV Library for computer vision applications. Dr. Dobb's Journal of Software Tools. Available from: OpenCV Documentation

[10]. Redmon J, Farhadi A. YOLOv3: An incremental improvement for object detection. 2018. Available from: arXiv

[11]. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. Available from: CVPR Proceedings

[12]. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (ICLR). 2021. Available from: ICLR Proceedings

[13]. Bochkovskiy A, Wang C, Liao H. YOLOv4: Optimal speed and accuracy of object detection. 2020. Available from: arXiv

[14]. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Available from: IEEE Xplore

[15]. Chollet F. Deep Learning with Python. Manning Publications. 2021. Available from: Manning Publications

[16]. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press. 2016. Available from: Deep Learning Book

[17]. Singh P, Rao M, Kulkarni S. Smart traffic management system using AI-based vehicle detection and classification. International Journal of Smart Transportation Systems. 2023. Available from: Springer

[18]. Nguyen T, Lee J, Kim H. Traffic density estimation using computer vision and deep learning techniques. Sensors Journal. 2022. Available from: MDPI Sensors

[19]. Chen Z, Li W, Huang Y. Intelligent vehicle counting and congestion analysis using video analytics. Journal of Transportation Engineering. 2021. Available from: ASCE Library.

 This work is licensed under a Creative Commons Attribution 4.0 International License