



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Traffic Density Analysis

Dr. A. Mallikarjuna Reddy¹, Anushna Vallampatla², B. Sai Kiran³, Adithya Erra⁴

Associate Professor, Department of Artificial Intelligence, Anurag University, Hyderabad, India¹

U.G. Student, Department of Artificial Intelligence, Anurag University, Hyderabad, India^{2,3,4}

ABSTRACT: Real-time traffic density estimation is a critical component of modern urban and traffic management systems, providing insights into traffic flow patterns, congestion levels, and peak traffic periods. This focuses on developing a comprehensive toolset utilizing advanced deep learning techniques, specifically leveraging the YOLOv8 (You Only Look Once) model, for accurate vehicle counting and density estimation in urban environments. The primary objective is to create a robust framework capable of efficiently detecting and counting vehicles within designated areas in real-time. The methodology encompasses data collection, model development, training, evaluation, and validation stages. By training the YOLOv8 model on a custom dataset and evaluating its performance using standard metrics, including precision, recall, and F1-score, aims to validate the model's effectiveness in diverse traffic scenarios. The toolset's applicability is demonstrated through the analysis of traffic density in sample images and videos, enabling the identification of congestion hotspots and informing data-driven traffic management strategies. Through this endeavor, seeks to contribute significantly to the enhancement of traffic management practices and the optimization of urban mobility, ultimately fostering more sustainable and efficient urban environments.

I. INTRODUCTION

Urbanization trends have led to unprecedented challenges in urban mobility, with traffic congestion being a significant concern for cities worldwide. As urban populations continue to grow, traditional methods of traffic surveillance and management struggle to cope with the increasing demand for efficient transportation solutions. Conventional approaches often rely on manual monitoring and static infrastructure, resulting in limited effectiveness in addressing dynamic traffic patterns and congestion hotspots. In this context, the integration of digital technologies, particularly artificial intelligence (AI) and computer vision, offers a promising avenue for transforming urban mobility management. By leveraging advanced algorithms and real-time data analysis, cities can gain valuable insights into traffic dynamics, optimize public transportation systems, and mitigate congestion-related issues. Deep learning-based object detection techniques, such as the YOLO (You Only Look Once) model, have emerged as powerful tools for accurately detecting and classifying objects, including vehicles, in complex urban environments. The motivation behind this problem statement stems from the urgent need to develop innovative solutions for addressing traffic congestion and enhancing urban mobility. By harnessing the capabilities of state-of-the-art technologies, such as YOLOv8, cities can move towards more sustainable and efficient transportation systems, ultimately improving the quality of life for residents and visitors alike.

II. LITERATURE SURVEY

Traffic Density Monitoring Control System Using Convolution Neural Network oct 2023.

The author proposed a Traffic Density Monitoring System (TDMS) based on Convolutional Neural Networks (CNNs) for real-time traffic density estimation. Unlike traditional methods, this system autonomously learns from traffic photos, improving scalability and efficiency. It also advocates for an intelligent traffic signal control framework using deep reinforcement learning, enhancing adaptability to changing traffic patterns. Validated with real-world data, this approach promises to enhance transportation management, leading to safer and more efficient road networks.

Real-Time Traffic State Measurement Using Autonomous Vehicles Open Data 2023

The author proposed a framework showcasing the potential of autonomous vehicle (AV) technologies to revolutionize urban transportation systems. Leveraging the multi-sensor systems of AVs, particularly LiDAR data, the framework enables real-time measurement of local traffic conditions. It computes key traffic flow parameters including volume, density, and speed, while also constructing traffic time-space diagrams. Testing conducted on the Waymo Open dataset sheds light on the feasibility of using AV data for real-time traffic state estimation, offering valuable insights for traffic operations and management applications.

Physics-Informed Deep Learning for Traffic State Estimation: Illustrations with LWR and CTM Models 14 June 2023
the author proposed exploring the application of a physics-informed deep learning (PIDL) approach for traffic state estimation (TSE) to tackle challenges related to data sparsity and sensor noise. PIDL combines deep learning neural networks with traffic flow theory knowledge, serving as a regularization tool during training to improve accuracy in traffic condition estimation. The implementation of PIDL involves utilizing two widely used traffic physics models: the Lighthill-Whitham-Richards (LWR) model and the cell.

III. PROBLEM STATEMENT

City traffic jams cause big problems for getting around, the environment, and how enjoyable life is. The usual ways we track and manage traffic aren't very good at solving these problems. So, we really need better solutions that use digital technology to carefully study how crowded the roads are, find where traffic jams are worst, and help us make smarter decisions to make city travel better. To deal with traffic jams, we need to use new and smart ways that use computers and data to understand traffic better. By using technology like real-time traffic monitors and fancy analysis tools, cities can figure out more about how traffic works and where the biggest problems are. This helps city leaders make better choices, fix problems where they're most needed, and make city transportation smoother and more eco-friendly.

3.1 Proposed system

The proposed system for Traffic Density Estimation involves:

1. Collecting and preprocessing traffic data.
2. Training the YOLOv8 model using TensorFlow.
3. Evaluating model performance.
4. Implementing real-time traffic density estimation with OpenCV.
5. Visualizing results and generating reports.
6. Deploying and integrating the model into traffic management systems.
7. Continuously improving and maintaining the system for ongoing effectiveness.

IV. DATASET

The dataset is meticulously organized into train and valid folders, with distinct subsets of labeled images representing different traffic scenes. The train folder constitutes the bulk of the dataset, encompassing 3240 images and labels, equivalent to around 90% of the entire dataset. In contrast, the valid folder contains 390 labeled images, constituting the remaining 10%. This partitioning strategy ensures the model is trained on a varied range of traffic scenarios while validating its performance on unseen data. Throughout the dataset, labels provide annotations for vehicles present in the images, furnishing crucial ground truth information for model training.

V. PROPOSED METHODOLOGY

1. Begin
2. Import necessary libraries and select datasets.
3. Perform inference on sample images:
 - 3.1. Initialize YOLOv8 model with pre-trained weights.
 - 3.2. Load sample image and perform inference using YOLOv8 model.
 - 3.3. Annotate detected objects on the image.
 - 3.4. Display annotated image.
4. Train the YOLOv8 model on custom dataset:
 - 4.1. Define dataset path and load YAML file.
 - 4.2. Check and count images in training and validation sets.
 - 4.3. Train the YOLOv8 model with specified parameters.
5. Evaluate model performance:
 - 5.1. Plot learning curves for loss functions.
 - 5.2. Visualize normalized confusion matrix.
6. Validate the best performing model:
 - 6.1. Load validation images.
 - 6.2. Perform inference using the best model.
 - 6.3. Display annotated images from the validation set.
7. Perform inference on sample video:

- 7.1. Copy the sample video to the working directory.
- 7.2. Initiate vehicle detection on the video using the best model.
- 7.3. Convert the output video to .mp4 format.
8. Analyze traffic density in the video:
 - 8.1. Define parameters for traffic analysis (thresholds, lane vertices).
 - 8.2. Open the video and initialize the Video Writer object.
 - 8.3. Read video frames, perform inference, and analyze traffic density.
 - 8.4. Write processed frames to output video.
9. End

Setup and Initialization (Step 1):

In this step, the environment is initialized, and the directory structure for the project is set up. Necessary libraries and dependencies are imported to facilitate subsequent tasks.

Loading YOLOv8 Pre-trained Model (Step 2):

This step involves loading a pre-trained YOLOv8 model, which is a state-of-the-art deep learning architecture commonly used for object detection tasks.

The pre-trained model serves as the starting point for fine-tuning on a custom dataset.

Dataset Exploration (Step 3):

The dataset is explored and analyzed to gain insights into its characteristics, such as size, structure, and content. Understanding the dataset helps in making informed decisions during model training and evaluation.

Fine-Tuning YOLOv8 (Step 4):

The loaded YOLOv8 model is fine-tuned on the custom dataset to adapt it to the specific requirements of the task at hand. Fine-tuning involves adjusting the model's parameters to improve its performance on the target dataset.

Model Performance Evaluation (Step 5):

This step focuses on evaluating the performance of the fine-tuned model through various analyses and metrics.

Learning Curves Analysis (Step 5.1):

Learning curves are analyzed to monitor the training progress of the model and ensure it converges to a satisfactory state.

Confusion Matrix Evaluation (Step 5.2):

The confusion matrix is used to assess the model's classification performance by comparing predicted labels with ground truth labels.

Performance Metrics Assessment (Step 5.3):

Performance metrics such as precision, recall, and F1-score are computed to quantitatively evaluate the model's performance.

Model Inference & Generalization Assessment (Step 6):

This step involves assessing the generalization ability of the model and performing inference on various datasets.

Inference on Validation Set Images (Step 6.1):

The model's performance is evaluated by performing inference on images from the validation set, which were not used during training.

Inference on an Unseen Test Image (Step 6.2):

Inference is performed on a test image that was not included in the training or validation set to assess the model's performance on unseen data.

Inference on an Unseen Test Video (Step 6.3):

Like step 6.2, inference is performed on a test video to evaluate the model's performance in a dynamic scenario.



Real-Time Traffic Intensity Estimation (Step 7):

This step involves estimating traffic intensity in real-time using the deployed model, potentially for applications such as traffic management and analysis.

Model Export for Cross-Platform Deployment (Step 8):

Finally, the model is prepared for deployment across different platforms by exporting it in a suitable format that can be easily integrated into production systems.

Performance Metrics:

Precision: Measures the proportion of correctly identified vehicles out of all detections made by the model.

Recall: Measures the proportion of actual vehicles in the image that were correctly detected by the model.

Mean Average Precision (mAP): A comprehensive metric that considers both precision and recall at various Intersection over Union (IoU) thresholds.

Frames per Second (FPS): Evaluates the model's inference speed, crucial for real-time applications.

Validation Results

Metrics	Value
Precision	0.930
Recall	0.890
mAP50	0.745
mAP50-95	0.570
fitness	0.80

Table 2 metrics

VI. RESULTS

After completing 100 epochs of training on our custom dataset, the model has successfully learned to detect various objects within images. Upon processing an image through the code, the model accurately identifies and returns the detected objects present in the image. Similarly, when applied to video streams using OpenCV and the YOLOv8 model, the system efficiently detects objects in real-time, providing invaluable insights into the visual content. This capability extends to diverse scenarios, enabling the model to detect objects across different environments and conditions. With its robust performance, the system enhances situational awareness and facilitates decision-making processes. Leveraging the power of YOLOv8 and OpenCV, the model ensures accurate and efficient object detection in both static images and dynamic video streams. This integration offers a versatile solution applicable across various domains, from surveillance and security to traffic monitoring and beyond.



Fig 6.1 Traffic density insights 1

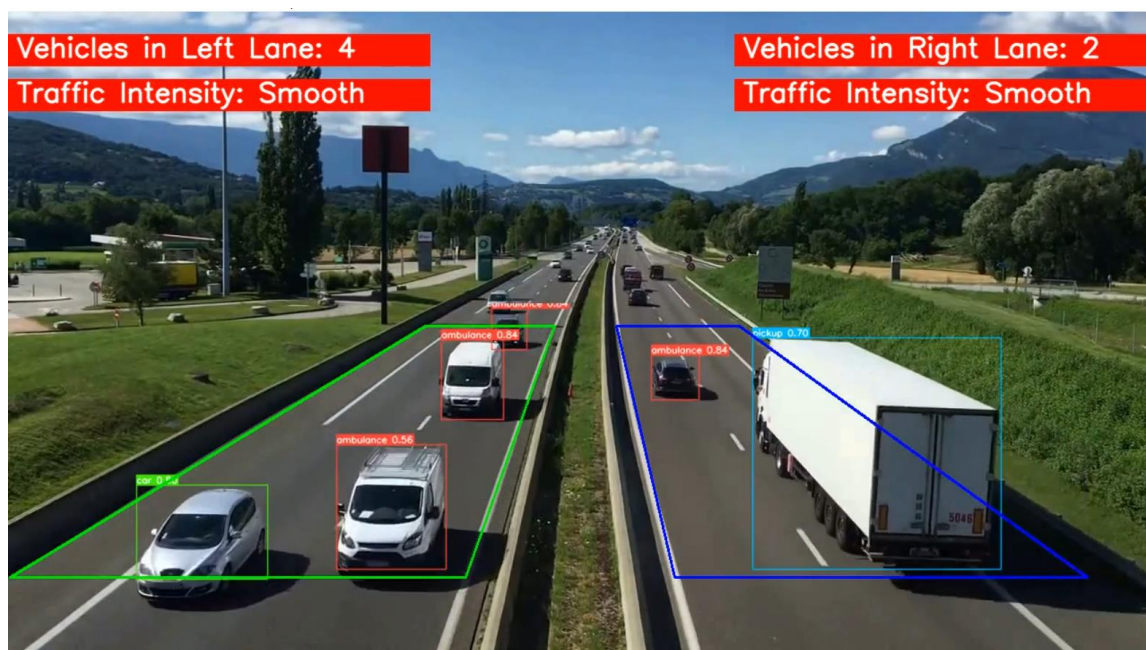


Fig 6.2 density analysis

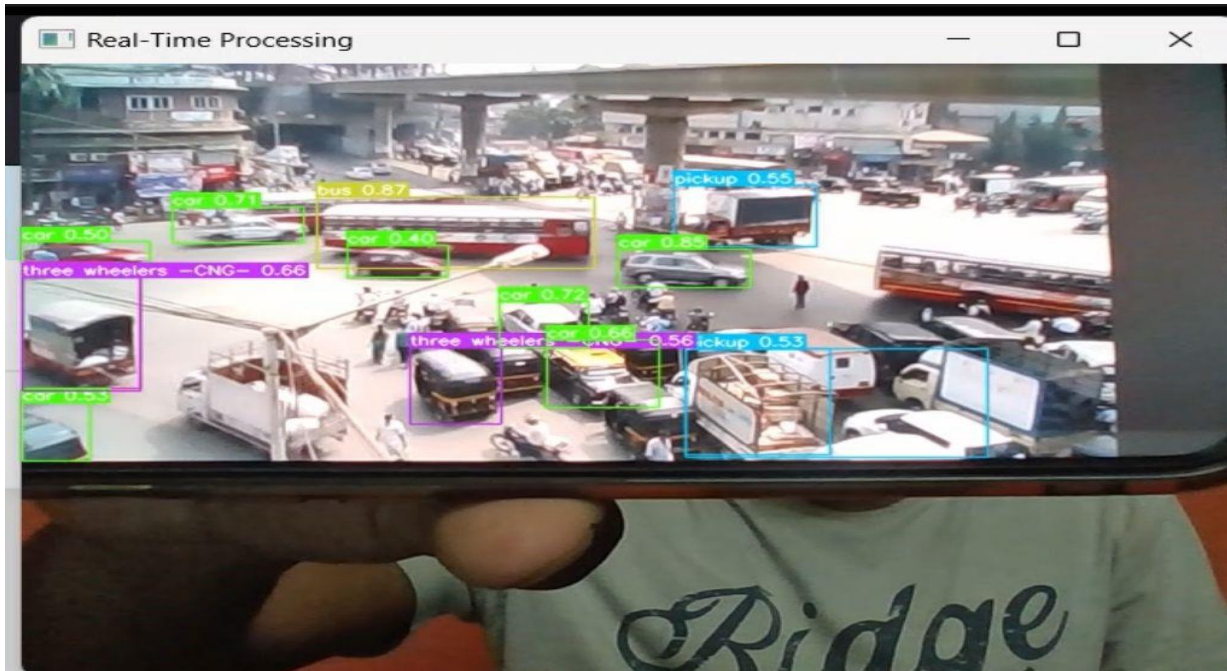


fig 6.3 Live camera access

VII. CONCLUSION

The proposed system introduces a vehicle detection system utilizing a pre-trained YOLOv8 model, demonstrating its capability to accurately recognize vehicles in both images and videos. During testing, the system showcased promising performance, reflecting its overall accuracy. Moreover, it displayed effectiveness in detecting vehicles across various Intersection over Union thresholds, underscoring its robustness. The system's adaptability to different lighting conditions and object densities suggests its suitability for real-world scenarios. Future endeavors may entail testing on more diverse datasets to enhance generalizability, exploring techniques such as hyperparameter tuning and data augmentation for potential performance improvements, and optimizing deployment for specific hardware platforms to enable real-time processing. Additionally, an optional extension to analyze traffic density by tallying vehicles in designated lanes could further enhance the system's capabilities. In summary, this project lays the foundation for a flexible vehicle detection system with promising applications in autonomous vehicles, traffic monitoring, and security surveillance.

ACKNOWLEDGMENT

We want to express our deep-felt gratitude and sincere thanks to our guide Dr.A.Mallikarjuna Reddy, Associate Professor, Department of AI, Anurag University, for his skillful guidance, timely suggestions, and encouragement in completing this project. We want to express our profound gratitude to all for having helped us in achieving this dissertation. Finally, we would like to express our heartfelt thanks to our parents, who were very financially and mentally supportive and for their encouragement to achieve our goals.

REFERENCES

- [1] Traffic Density Monitoring Control System Using Convolution Neural Network oct 2023 Ashish Dibouliya, Rabindranath Tagore University, Bhopal M.P. India Varsha Jotwani, Rabindranath Tagore University
- [2] Real-Time Traffic State Measurement Using Autonomous Vehicles Open Data Zhaohan Wang; Profita Keo; Meead Saberi Date of Publication: 26 July 2023
- [3] Physics-Informed Deep Learning for Traffic State Estimation: Illustrations with LWR and CTM Models Archie J. Huang; Shaurya Agarwal Date of Publication: 14 June 2022
- [4] Toward a Cost-Effective Motorway Traffic State Estimation from Sparse Speed and GPS Data date of publication March 17, 2021, Zied Bouyahia; Hedi Haddad; Stéphane Derrode;

- [5] Yi-Qi Huang, Jia-Chun Zheng, Shi-Dan Sun, Cheng-Fu Yang and Jing Liu, "Optimized YOLOv3 Algorithm and Its Application in Traffic Flow Detections", Appl. Sci. 2020, 10, 3079, doi:10.3390/app1009307.
- [6] Adel Ammar¹, Anis Koubaa¹, Mohammed Ahmed¹, Abdulrahman Saad, "Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3", IEEE, 2020
- [7] Dinh Viet Sang, Duong Viet Hung, "YOLOv3-VD: A sparse network for vehicle detection using variational dropout", SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam.
- [8]. Manoharan, Samuel. "An improved safety algorithm for artificial intelligence enabled processors in self driving cars." Journal of Artificial Intelligence 1, no. 02 (2019): 95-104.
- [9]. Koresh, M. H. J. D., and J. Deva. "Computer vision-based traffic sign sensing for smart transport." Journal of Innovative Image Processing (JIIP) 1, no. 01 (2019): 11-19.
- [10]. Reha Justin, Dr. Ravindra Kumar, "Vehicle Detection and Counting Method Based on Digital Image Processing in Python", ICSCAAIT, E-ISSN: 2348-2273, 2018



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details