Vehicle, Pedestrian Detection, Speed Estimation and License Plate Recognition -Using YOLOV8.

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the internship

By

Lokesh RLN (622246)

Vinuth B (622127)

Uma Mahesh D (622135)

Gowtham M (522156)

Students of National Institute of Technology Andhra Pradesh

Supervisor:

Dr. Mohammad Farukh Hashmi

Assistant Professor

Department of Electronics and Communication Engineering

Department of Electronics and Communication Engineering

National Institute of Technology, Warangal

2023-24

Vehicle, Pedestrian Detection, Speed Estimation and License Plate Recognition - Using YOLOV8.

Lokesh R [a], Uma Mahesh D [a], Vinuth B [a], Gowtham M [b]

[a] *Department of Electronics and Communication, NIT Andhra Pradesh*

[b] *Department of Electrical and Electronics, NIT Andhra Pradesh*

Abstract:

This project utilizes a custom-trained model based on YOLOv8 Large for comprehensive analysis and management of traffic scenarios. The model integrates advanced capabilities including vehicle and pedestrian detection, real-time vehicle tracking, speed estimation, and license plate detection and recognition. Leveraging YOLOv8 Large's robust architecture, the system achieves high accuracy in identifying and tracking vehicles and pedestrians, crucial for enhancing safety and efficiency in urban environments. Real-time speed estimation enhances traffic flow management, while accurate license plate recognition ensures effective law enforcement and vehicle identification. The project highlights the adaptability and efficiency of YOLOv8 Large in handling complex tasks simultaneously, providing a scalable solution for intelligent transportation systems. Experimental results demonstrate the effectiveness of the proposed model across varied environmental conditions, showcasing its potential for deployment in real-world applications.

Keywords:

Vehicle, pedestrian detection, YOLO (You Only Look Once), Speed Estimation, Vehicle Tracking, Computer Vision, OCR (Optical Character Recognition)

1. Introduction:

In recent years, the convergence of computer vision technologies and deep learning has catalyzed significant advancements in Intelligent Transportation Systems (ITS), revolutionizing how we monitor, manage, and enhance traffic scenarios. This project harnesses the capabilities of YOLOv8 Large, a state-of-the-art deep learning model, to address multifaceted challenges in urban mobility and transportation safety.

YOLOv8 Large stands out for its efficiency in real-time object detection, making it ideal for applications requiring rapid and accurate analysis of traffic dynamics. The model's ability to detect vehicles, pedestrians, and even read license plates in varying environmental conditions equips it to support a wide range of ITS functionalities.

Key components of this project include real-time speed estimation, which provides critical insights into vehicle behavior and compliance with speed regulations, thereby contributing to safer roads and optimized traffic flow. Additionally, precise license plate detection and recognition capabilities facilitate tasks such as automated toll collection, vehicle tracking, and law enforcement operations.

By integrating these advanced functionalities into the YOLOv8 framework, this project aims to demonstrate its efficacy in enhancing urban mobility systems. The adaptability and scalability of YOLOv8 Large ensure reliable performance across diverse scenarios, from busy city intersections to highway monitoring. Ultimately, this research contributes to the ongoing evolution of ITS, offering scalable solutions to meet the complex challenges of modern transportation infrastructures.

2. Literature Survey:

The literature survey on vehicle detection, speed estimation, and license plate recognition using YOLO v8 in machine learning reveals a focus on the application of various YOLO models for license plate detection and recognition (LPDR). While YOLO v8 is specifically mentioned in N (2024) as part of an Advanced Automatic Number Plate Detection System, the paper does not detail its use in speed. N (2024) emphasizes the integration of YOLO v8 for precise outcomes in license plate detection and character recognition, highlighting the system's potential for real-time applications in traffic management and surveillance.

Contradictory to the direct application of YOLO v8, other papers discuss different versions of the YOLO model. For instance, Chakravarthy (2021) compares YOLOv3 and YOLOv4, noting YOLOv4's accuracy and YOLOv3's speed with reduced image resolution. Hendry and Chen (2019) and Rathi et al. (2022) utilize YOLO-darknet frameworks for license plate detection, with Hendry and Chen (2019) achieving high accuracy in detection and recognition.

Al-Batat et al. (2022) and Pan et al. (2023) explore the use of YOLOv4 and YOLOv7, respectively, for robust ALPR systems, with Pan et al. (2023) achieving high accuracy in complex environments. Shakeel et al. (2024) specifically addresses the use of YOLO v8 for detecting Bangla license plates, demonstrating a high mean average precision (mAP) of 98.4% and discussing the challenges of data labeling and varying license plate formats.

For vehicle detection and license plate recognition, YOLO models have been widely adopted due to their efficiency and accuracy. Hou et al. (2018) demonstrates a Vehicle License Plate Recognition System (VLPRS) using YOLOv2 and YOLOv3, achieving high precision and speed in detection and recognition tasks. Similarly, Zhang et al. (2019) and Chen et al. (2021) employ YOLOv2 and YOLOv3 respectively, for license plate detection and recognition, with Chen et al. (2021) also incorporating real-time recognition capabilities. Baviskar et al. (2022) extends the use of YOLO models to YOLOv5, tackling challenges such as plate background complexity and lighting inconsistencies, and achieving high precision in detection.

In terms of speed estimation, Luvizon et al. (2014) presents a novel system that uses text detection to locate license plates and estimate vehicle speed with high precision and low average error, although it does not specify the use of YOLO models for this purpose.
Contradictions and interesting facts emerge when comparing the effectiveness of different YOLO versions and other machine learning techniques. For instance, Huang et al. (2019) explores the use of Extreme Learning Machine (ELM) for license plate recognition, showing promising results in terms of speed and accuracy.

Hendry and Chen (2018) and Dhar et al. (2019) discuss the use of YOLO models for license plate detection with adaptations to specific datasets and conditions, such as the AOLP dataset and images captured in Bangladesh. Tusar et al. (2022) and Vedhaviyassh et al. (2022) highlight the use of YOLOv5 for license plate detection, with Vedhaviyassh et al. (2022) comparing OCR methods and finding EasyOCR to be more accurate than Tesseract OCR.

YOLO models, including YOLOv8, are integral to the development of robust vehicle detection and license plate recognition systems. The literature indicates that while YOLOv8 is not explicitly mentioned in the context of vehicle speed estimation, its efficiency and accuracy in real-time applications make it a promising candidate for such tasks. The advancements in YOLO models have facilitated the creation of systems that can operate effectively in complex environments, such as those with varying weather conditions and levels of occlusion (Abbass & Marhoon, 2021; Mane, 2024).

3. Methodology:

YOLOv8 (You Only Look Once version 8) is an advanced object detection model known for its efficiency and accuracy in real-time applications. Building upon the YOLO architecture, YOLOv8 retains the fundamental principle of single-shot detection, processing the entire image at once to predict bounding boxes and class probabilities simultaneously. This approach contrasts with traditional two-stage detectors, resulting in faster inference times while maintaining competitive performance in object detection tasks.

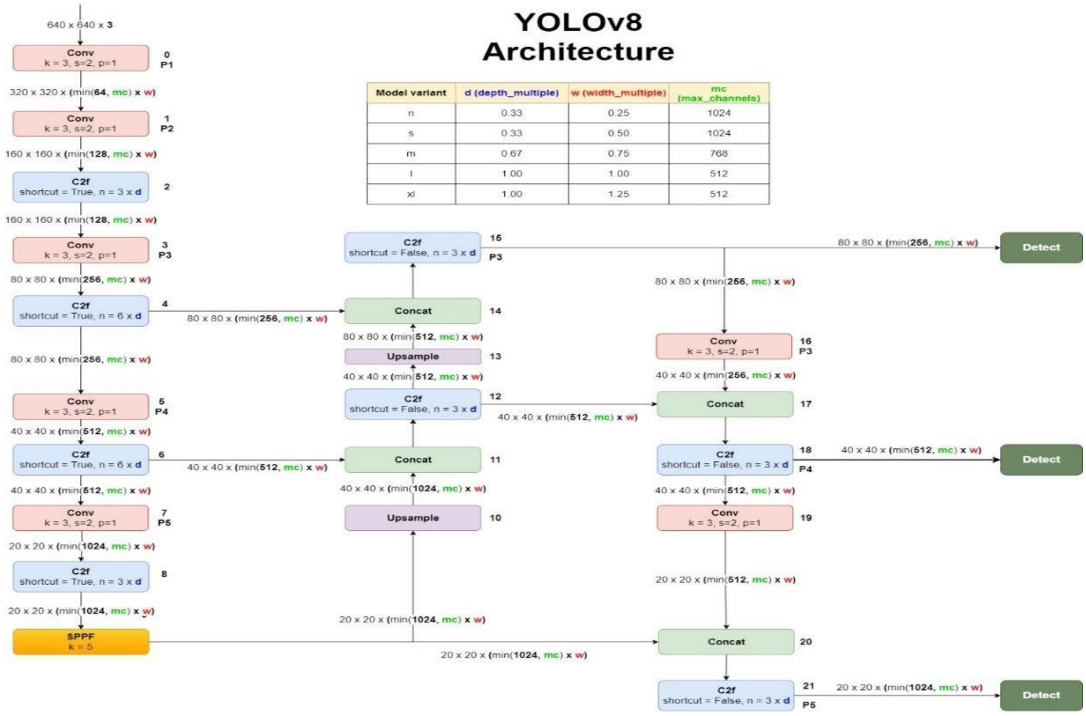
At its core, YOLOv8 typically employs a deep convolutional neural network (CNN) as its backbone network, such as Darknet, to extract hierarchical features from input images. These features are essential for understanding the spatial relationships and contextual information necessary to accurately localize and classify objects within complex scenes. The model often incorporates a feature pyramid network (FPN) or similar mechanisms to capture multi-scale features, enabling it to detect objects of varying sizes and scales across different layers of the network.

One of YOLOv8's distinguishing features is its use of anchor boxes—predefined shapes and sizes used to predict bounding boxes for objects of interest. During training, these anchor boxes are adjusted to better fit the shape and aspect ratio of objects present in the dataset, enhancing the model's ability to accurately localize objects under varying conditions.

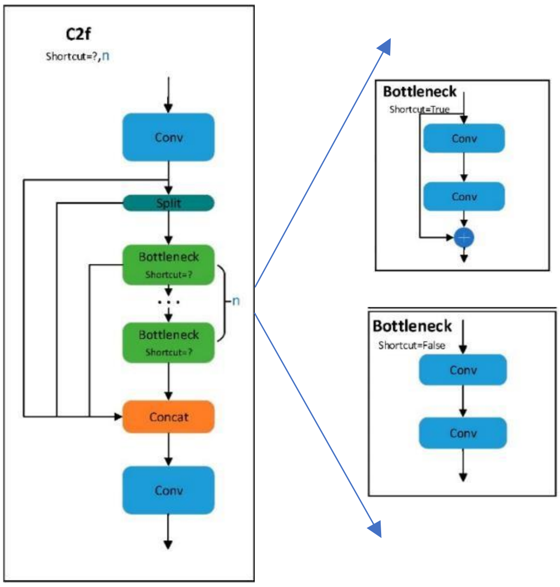
In terms of training and optimization, YOLOv8 utilizes a specialized loss function that combines localization loss and classification loss. This function is designed to optimize the model parameters based on the accuracy of predicted bounding boxes and the correct classification of objects. Techniques such as focal loss may be employed to prioritize difficult-to-detect objects during training, further improving overall detection performance.

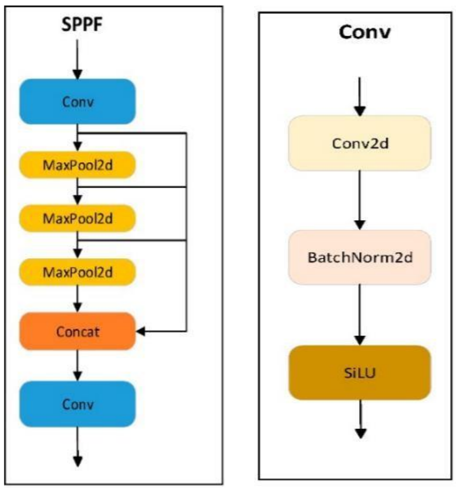
YOLOv8 is particularly noted for its real-time performance capabilities, achieving high inference speeds on both CPU and GPU platforms. This characteristic makes it well-suited for deployment in applications requiring rapid object detection, such as autonomous driving systems, surveillance, and robotics. Moreover, its modularity allows researchers and developers to customize the architecture, fine-tune hyperparameters, and integrate additional components to adapt the model to specific tasks and improve performance on specialized datasets.

This project aims to demonstrate the efficacy and versatility of the YOLOv8 Large model in handling complex, real-world traffic scenarios. By leveraging deep learning and computer vision techniques, the proposed system enhances the capability of existing ITS frameworks, offering scalable solutions for urban mobility challenges.



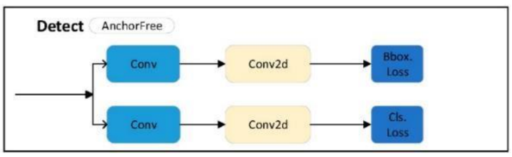
Block Diagram of YOLOv8 Architecture





Block Diagrams Of C2f and Bottleneck

Block Diagrams Of SPPF and Conv



Block Diagram of Detect Block

3.1 Conv Block:

It contains batch normalization block and SiLU activation function. A Convolutional Layer (Conv Layer) in CNNs extracts features from input data using small filters (kernels) that slide over the input to detect patterns. The stride controls the step size, and padding adjusts the output size. An activation function, like ReLU, adds non-linearity. Each filter creates a feature map, and parameter sharing reduces the number of parameters. Conv Layers focus on local regions and provide translation invariance, making them essential for learning and extracting spatial features in CNNs.

3.2 SPPF Block:

It contains Conv Layers and Max Pooling layers which are used to reduce the computations. The SPPF (Spatial Pyramid Pooling Fusion) block combines spatial pyramid pooling with feature fusion in neural networks. It enables effective capture of multi-scale features by pooling features from multiple spatial bins and fusing them, improving the model's capability to handle objects of different sizes in tasks like object detection and image classification.

3.3 C2F Block:

It contains Conv block, a split block, and combination of Bottleneck blocks. C2f (Cross Stage Partial Networks with C3 Fusion) enhances neural network efficiency and feature extraction. It splits and merges feature maps to reduce computation while preserving features (CSP) and integrates information across network stages for better feature reuse and gradient flow (C3 Fusion). This balance improves performance in models like YOLOv8.

3.4 Bottleneck Block:

There are two types here one with shortcut True and another with shortcut False. First one is nothing but residual network and second one is just a combination of conv block. A bottleneck block is a neural network module used for efficient feature extraction. It includes a sequence of layers: a 1x1 convolution to reduce dimensions, a 3x3 convolution for feature processing, and another 1x1 convolution to restore dimensions. This design reduces computational complexity while maintaining effective feature representation, commonly found in architectures like ResNet to enhance performance.

3.5 Detect Block:

The Detect block in YOLOv8 predicts bounding boxes, objectness scores, and class probabilities for objects in an image using anchor boxes. It refines predictions with Non-Maximum Suppression (NMS), ensuring accurate object detection as the final stage of the model.

4. Datasets:

4.1 Kitti Dataset:

Kitti dataset is very popular dataset on vehicle detection projects which contains 4 classes (Biker, Car, Pedestrian, Truck). Using this data set we made our model that can detect vehicles and pedestrians and we also used this in speed estimation module to detect the vehicles. Kitti dataset contain 7638 images for training, 367 images for testing and 740 images for validation purpose.

4.2 VisDrone Dataset:

VisDrone dataset is also very big dataset in which images are captured by drone. Used on vehicle detection contains 10 classes (Pedestrian, People, Bicycle, Car, Van, Truck, Tricycle, Awning-Tricycle, Bus, Motor) we used this model for detecting vehicles and pedestrians. VisDrone dataset contains 6471 images for training, 1610 images for testing and 548 for validating.

4.3 Highway Dataset:

Highway dataset is dataset of different vehicles on highway road. Very famously used in Vehicle Detection Projects. It contains 5 classes (Bus, Car, Motorbike, Person, Truck). Highway dataset contains the 1704 images for training, 467 images for validation, 251 images for testing.

4.4 InRia Dataset:

InRia dataset is very popular dataset which is used to detect the persons it contains only one class(Person). InRia dataset contains 1896 images for training, 180 for testing, 90 for validating.

4.5 License Plate Dataset:

License Plate dataset is big dataset which is used to detect the license plate of vehicle. It contains only one class (License Plate). It contains 21173 images for training, 2046 images for validating, 1019 images for testing.

5. Proposed Methodology:

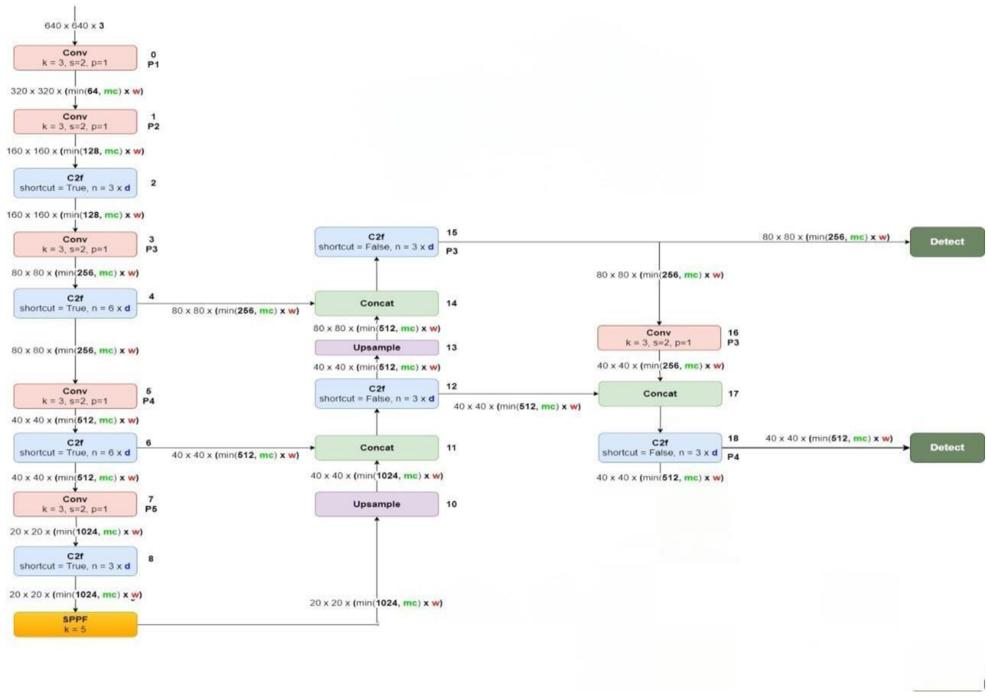
5.1 Modified YOLOv8:

In our project, we customized YOLOv8 to better suit our dataset, which predominantly features smaller to moderate-sized objects. Specifically, we strategically removed layers 19, 20, and 21 from the architecture, responsible for detecting large objects. By reducing the number of layers, we aimed to optimize computational resources and enhance the model's efficiency in detecting prevalent object sizes. The retained components of YOLOv8, including its robust Darknet CNN backbone for feature extraction, were pivotal in maintaining the model's capability to accurately process and classify objects.

Through rigorous training and validation on annotated datasets, we assessed the performance of our modified YOLOv8 model. Results indicated significant improvements in the accuracy of localizing and classifying smaller objects, crucial for applications requiring swift and precise object detection. This tailored approach not only enhanced the model's speed but also underscored its adaptability to meet specific challenges posed by our dataset. By aligning YOLOv8's architecture with our dataset's characteristics, we demonstrated its effectiveness in real-world scenarios demanding efficient and accurate object detection capabilities.

5.1.1 Hyperparameters:

While training and validating the modified yolov8 model for various purposes including speed estimation, tracking, license plate detection we used Epochs=100; Batchsize=12; Imgz=640; Mosaic=1; Mixup=1; Flipud=0.3; Workers=50.



Modified YOLOv8 Block Diagram

5.2 Speed Estimation:

Speed estimation in computer vision involves a sophisticated process that integrates various techniques to accurately determine the speed of vehicles. Here's how it typically works:

5.2.1 Vehicle Detection and Tracking:

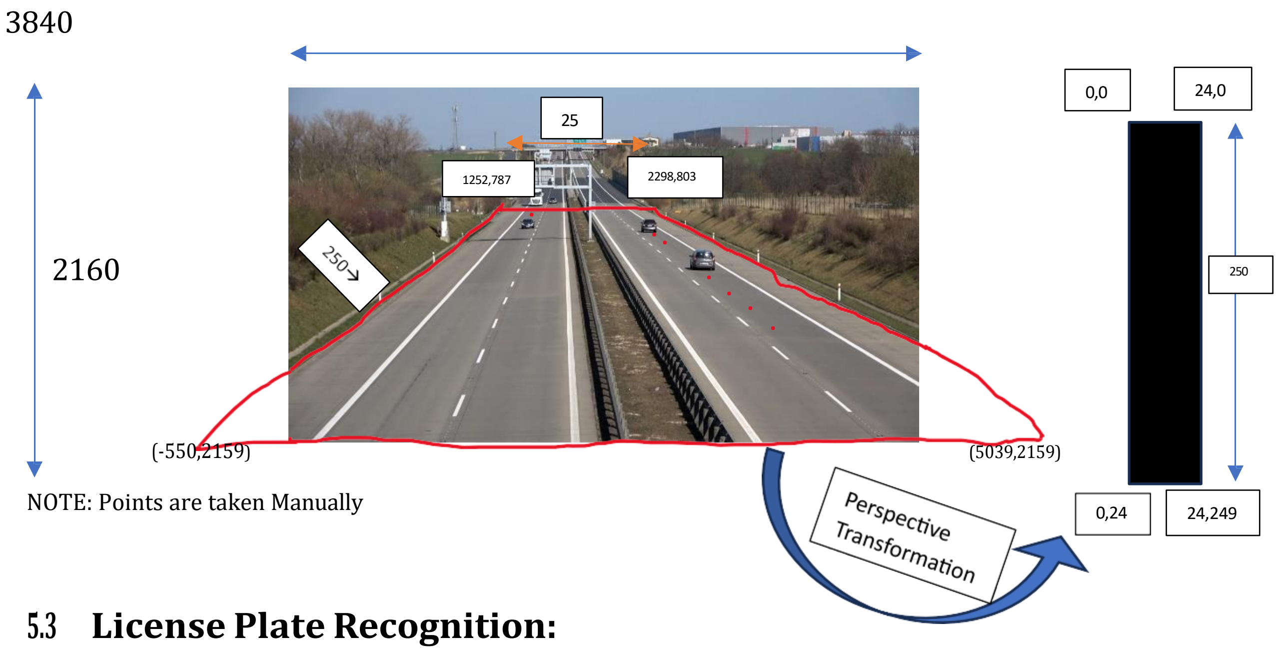
Initially, vehicles are detected within video frames using object detection models such as YOLOv8. Once identified, these vehicles are tracked across subsequent frames using advanced algorithms like ByteTrack. ByteTrack ensures robust tracking by associating vehicle detections over time, accommodating for occlusions and changes in appearance.

5.2.2 Perspective Distortion and Transformation:

As vehicles move within the camera's field of view, their positions relative to the camera perspective change, causing perspective distortions. This phenomenon makes closer vehicles appear larger and farther vehicles smaller. To mitigate this, perspective transformation techniques are employed. These techniques, such as homography, correct the perspective to provide a more accurate representation of the distances on the road surface.

5.2.3 Calculating Speed:

To estimate vehicle speed, the system tracks the positions of vehicles across consecutive frames. By applying perspective correction through homography, the system transforms the detected vehicle positions into a consistent viewpoint. The distance a vehicle travels between frames is then measured accurately on the road surface. Speed is calculated by dividing this distance by the known time interval between frames, derived from the video frame rate.



5.3 License Plate Recognition:

5.3.1 License Plate Detection:

Using computer vision techniques and deep learning models (Modified YOLOv8), the system detects vehicles within each frame of the video. Once a vehicle is detected, the focus shifts to identifying the region of interest containing the license plate.

5.3.2 License Plate Localization:

After locating the license plate within the vehicle's bounding box, image processing algorithms are applied to precisely isolate the plate area. This step ensures that only the license plate area is extracted for further analysis.

5.3.3 Character Recognition:

Optical Character Recognition (OCR) algorithms are employed to extract and interpret the alphanumeric characters on the license plate. These algorithms are trained on large datasets of license plate images to accurately recognize and decode the characters, despite variations in font, size, and lighting conditions.

5.3.4 Data Storage and Visualization:

As the system processes each frame, it records the detected license plate number along with associated metadata (such as timestamp, vehicle ID, and tracking information) into a CSV file or a database. If any frames are missed or if there are gaps in the data due to detection errors or occlusions, the system fills in missing information to ensure continuity in the tracking data.

5.3.5 Output Visualization:

The recognized license plate numbers, along with any additional tracked information, are visualized in real-time on the output video feed or displayed as overlays. This visualization aids in monitoring and verification tasks, such as identifying vehicles of interest or enforcing traffic regulations.

6. Results:

6.1 InRia Dataset:

Class	Images	Instances	mAP(50)	mAP(50-95)
All	90	184	0.79	0.53

➔Results on Original YOLOv8

Results on Modified YOLOv8➔

Class	Images	Instances	mAP(50)	mAP(50-95)
All	90	184	0.87	0.6

6.2 Highway Dataset:

Class	Images	Instances	mAP(50)	mAP(50-95)
Bus	75	126	0.68	0.54
Car	256	332	0.84	0.72
Motorbike	34	44	0.7	0.32
Person	47	65	0.52	0.27
Truck	205	291	0.81	0.68
All	487	858	0.72	0.53

➔ Results on Original YOLOv8

Results on Modified YOLOv8 ➔

Class	Images	Instances	mAP(50)	mAP(50-95)
Bus	75	126	0.75	0.64
Car	256	332	0.91	0.78
Motorbike	34	44	0.72	0.39
Person	47	65	0.56	0.33
Truck	205	291	0.87	0.73
All	487	858	0.77	0.59

6.3 Kitti Dataset:

Class	Images	Instances	mAp(50)	mAP(50-95)
Biker	69	104	0.55	0.39
Car	650	3256	0.82	0.64
Pedestrian	181	558	0.66	0.44
Truck	115	168	0.52	0.37
All	740	4096	0.68	0.52

➔ Results on Original YOLOv8

Results on Modified YOLOv8➔

Class	Images	Instances	mAp(50)	mAP(50-95)
Biker	69	104	0.62	0.45
Car	650	3256	0.84	0.59
Pedestrian	181	558	0.72	0.47
Truck	115	168	0.59	0.41
All	740	4096	0.72	0.57

6.4 Visdrone Dataset:

Class	Images	Instances	mAP(50)	mAP(50-95)
Pedestrian	520	8844	0.64	0.42
People	482	5125	0.39	0.26
Bicycle	364	1287	0.29	0.14
Car	515	14064	0.72	0.58
Van	421	1975	0.48	0.33
Truck	266	750	0.55	0.39
Tricycle	337	1045	0.29	0.16
Awning-tricycle	220	532	0.17	0.13
Bus	131	231	0.58	0.38
Motor	485	4886	0.41	0.25
all	548	38759	0.63	0.44

➔ Results on Original YOLOv8

Results on Modified YOLOv8 ➔

Class	Images	Instances	mAP(50)	mAP(50-95)
Pedestrian	520	8844	0.69	0.48
People	482	5125	0.43	0.32
Bicycle	364	1287	0.35	0.27
Car	515	14064	0.79	0.62
Van	421	1975	0.52	0.4
Truck	266	750	0.58	0.49
Tricycle	337	1045	0.31	0.18
Awning-tricycle	220	532	0.22	0.19
Bus	131	231	0.64	0.42
Motor	485	4886	0.47	0.33
all	548	38759	0.71	0.52

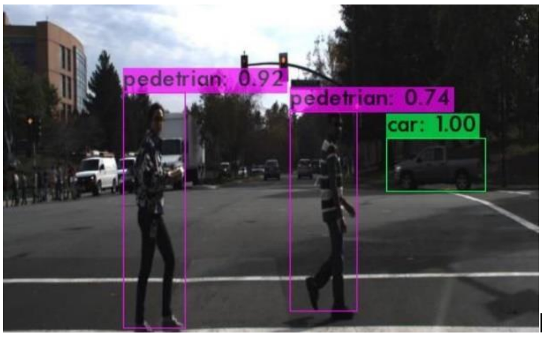
6.5 License Plate Dataset:

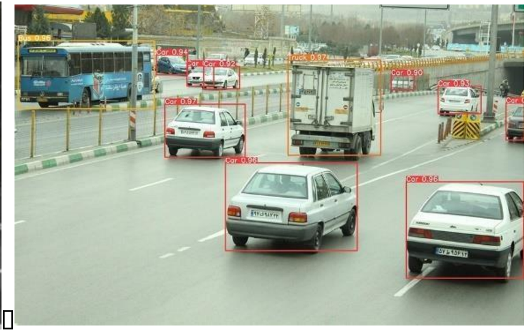
Class	Images	Instances	mAP(50)	mAP(50-95)
All	2046	2132	0.97	0.71

➔ Results on Original YOLOv8

Results on Modified YOLOv8 ➔

Class	Images	Instances	mAP(50)	mAP(50-95)
All	2046	2132	0.97	0.71

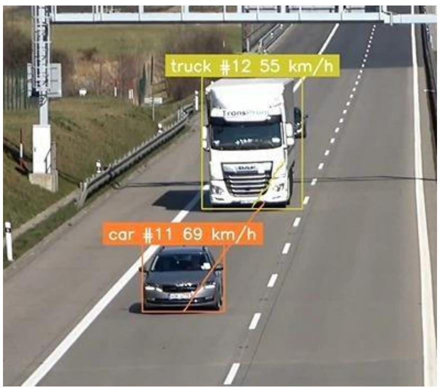




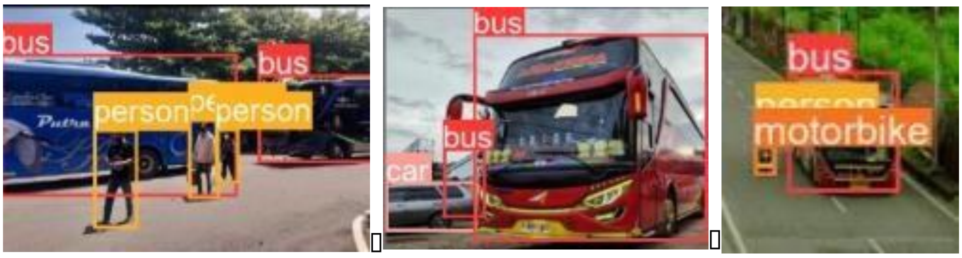
Detections in KITTI Dataset

Detections in VisDrone Dataset

Number Plate Recognition



Speed Estimation



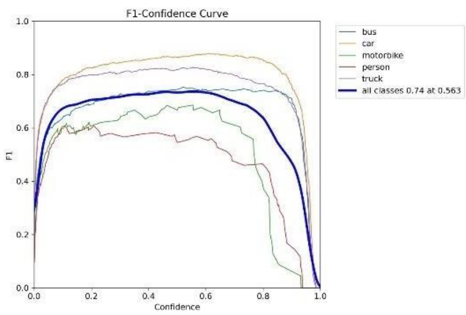
Detections in Highway Dataset

Detections in InRia Dataset

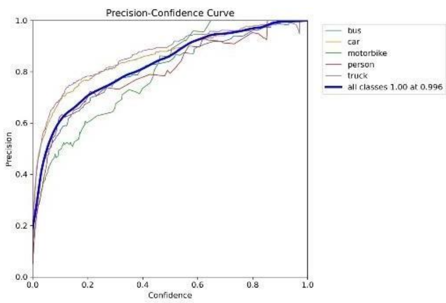
Detections in License Plate Dataset

6.6 Stats (Best Model):

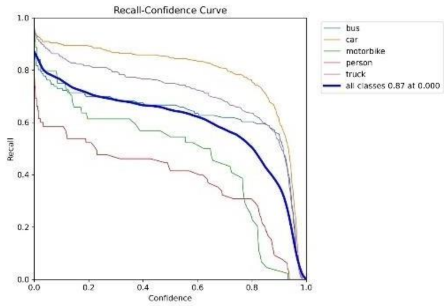
1. Highway Dataset:



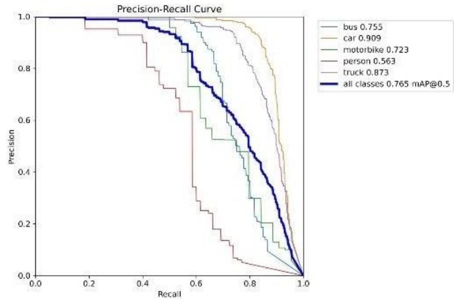
F1 Curve



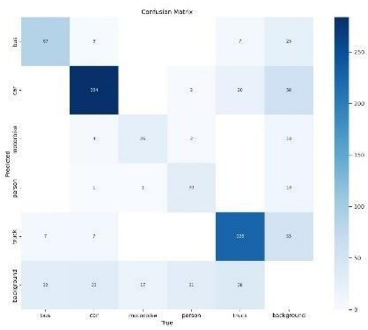
Precision Curve



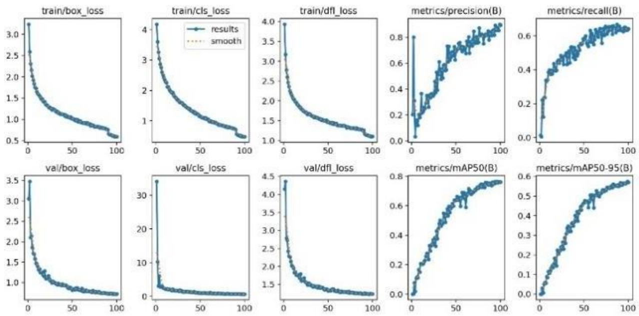
Recall Curve



Precision-Recall Curve

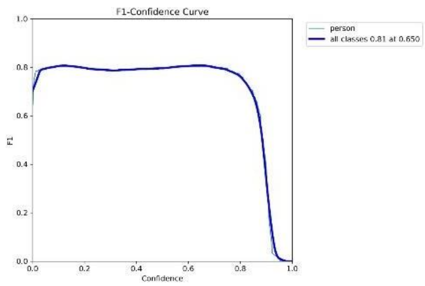


Confusion Matrix

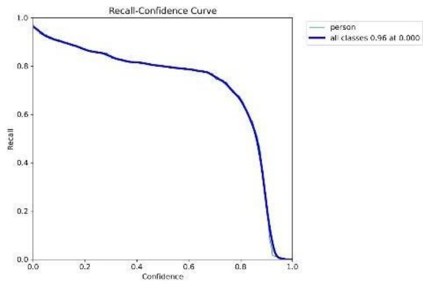


Results

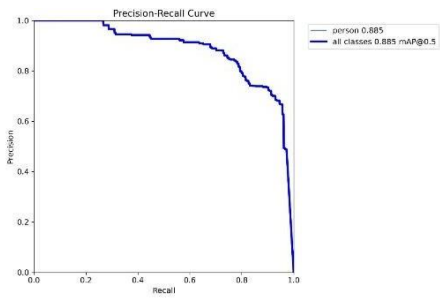
2. InRia Dataset:



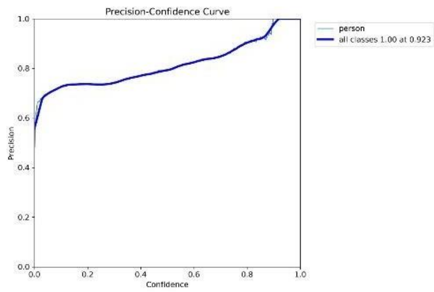
F1 Curve



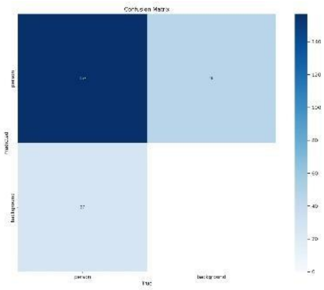
Recall Curve



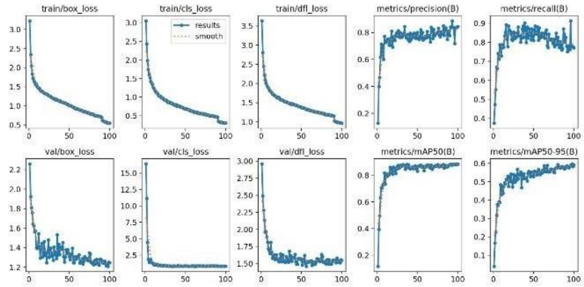
Precision-Recall Curve



Precision Curve

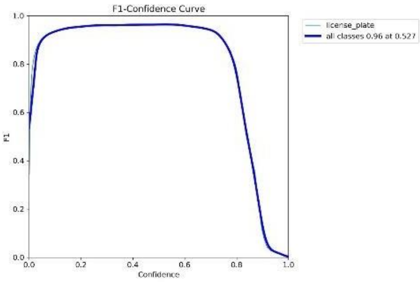


Confusion Matrix

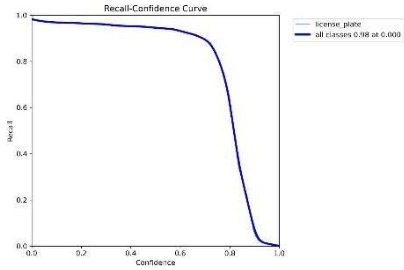


Results

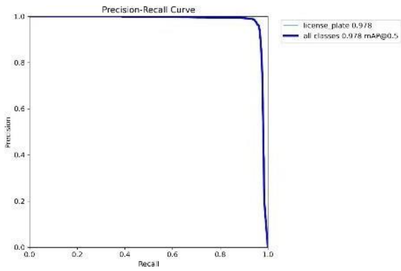
3. License Plate Dataset:



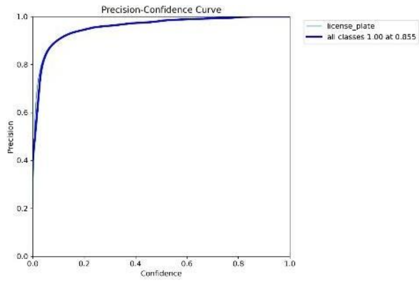
F1 Curve



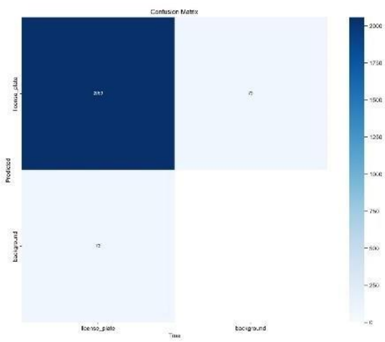
Recall Curve



Precision-Recall



Precision Curve



Confusion Matrix

7. Conclusion and Future Work:

In conclusion, our project successfully customized YOLOv8 to optimize the detection of smaller to moderate-sized objects, achieving significant improvements in speed and accuracy for real-time applications. By strategically removing layers intended for large object detection, we streamlined the model's architecture to better suit our dataset's characteristics. This refinement not only enhanced computational efficiency but also markedly improved the model's ability to precisely detect and classify objects relevant to our specific application scenarios.

Furthermore, integrating advanced techniques such as perspective transformation and robust tracking algorithms like ByteTrack enabled our system to perform accurate speed estimation. Correcting perspective distortions and effectively tracking vehicle movements across frames ensured reliable calculations essential for tasks such as traffic monitoring and safety enforcement.

Looking forward, future work involves further refining the YOLOv8 model through continuous training with diverse datasets to enhance detection robustness across varying environmental conditions. Exploring advanced speed estimation methods, including multi-sensor data fusion and refining algorithms for practical deployment, will be crucial. Additionally, expanding the system's capabilities to include multi-modal object detection and developing user-friendly interfaces for seamless integration into operational settings are priorities for enhancing overall system efficacy and usability in smart city initiatives and beyond.

8. References:

1. Sen Zhang, Shuai Chen, Jie Li, Huichen Zhang An Improved Vehicle-license Plate Recognition Based on Color Clues and Coding Rules Traffic Management Research Institute of the Ministry of Public Security, Wuxi Jiangsu, P.R. China.(2018)
   DOI: https://doi.org/10.1109/icivc47709.2019.8981034
2. Xiaoying Hou, Meixia Fu, Xifang Wu, Zhongiie Huang, Songlin Sun Vehicle License Plate Recognition System Based on Deep Learning Deployed to PYNQ Beijing University of Posts and Telecommunications, Beijing, China.(2018)
   DOI: https://doi.org/10.1109/iscit.2018.8587934
3. D.R Vedhaviyassh, D Arun, M Safa, G Saranya, R Sudhan Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm.(2022)
   DOI: https://doi.org/10.1109/iceca55336.2022.10009215
4. Prabuddha Gacche, Suryakant Rathod, S M Rathod, Ajay Sharma, K Arulprasath, Manesh Kokare Foreign Object Detection and Classification using AI and ML for Radio Images. (2022)
   DOI: https://doi.org/10.1109/iconsip49665.2022.10007443
5. Ayushi Sharma, Muskan Prakash, J N Singh, Jyotsna Pathak Object Detection using OpenCV and Python. (2021)
   DOI: https://ieeexplore.ieee.org/document/9725638
6. Zengfang Shi, Meizhou Liu Moving Vehicle Detection and Recognition Technology based on Artificial Intelligence. (2022)
   DOI: https://doi.org/10.46300/9106.2022.16.49

7. Prof. V. N. Mahawadiwar1, Surendra Chokhandre2, Shreyash Borkar3, Donisha Tawade Study On Advanced Vehicle Number Plate Detection Systems-A Review. (March 2024)
Link: http://www.ijsrem.com
8. Shenghu Pan, Jian Liu and Dekun Chen Research on License Plate Detection and Recognition Southwest Petroleum University, Chengdu 610500, China. (2022)
9. Deepak Mane, Prashant Kumbharkar, Sunil Sangave, Nirupama Earan, Komal Patil, Sakshi Bonde A Metaphor Analysis on Vehicle License Plate Detection using YOLOv8. (2024)
10. Rahul Hegde , Rosha G Naik , Sanobar Patel , Kundan Sagar , K S Shivaprakasha Object Detection using Machine Learning: A Survey. (2019)
DOI: https://doi.org/10.37628/jeset.v5i2.1138
11. Maryam Raad Shihab1, Rana Farid Ghani2, Athraa Jasim Mohammed Machine Learning Techniques for Vehicle Detection. (2022)
DOI: https://doi.org/10.33103/uot.ijccce.22.4.1
12. Zhenyu Lu, Quanbo Ge, Tianming Zhan, Jia Lu Multi-object Detection Method based on YOLO and ResNet Hybrid Networks Nanjing University of Information Science and Technology. (2019)
DOI: https://doi.org/10.1109/icarm.2019.8833671
13. Radha Pandey, Aruna Malik Object Detection and Movement Prediction for Autonomous Vehicle: A Review. (2021)
DOI: https://doi.org/10.1109/icsccc51823.2021.9478167
14. Bo Yang Research on Vehicle Detection and Recognition Technology Based on Artificial Intelligence Sichuan University of Arts and Science. (2023)
DOI: https://doi.org/10.1016/j.micpro.2023.104937
15. Jun-Hwa Kim, Chee Sun Won, Namho Kim High-Speed Detection Based On Yolo-V8 Dongguk University. (2023)
DOI: https://doi.org/10.1109/icassp49357.2023.10095516
16. Muhammad Azhad Bin Zuraimi, Fadhlan Hafizhelmi Kamaru Zaman Vehicle Detection and Tracking using YOLO and DeepSORT Universiti Teknologi MARA. (2021)
DOI: https://doi.org/10.1109/iscaie51753.2021.9431784