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*

E-mail: *622246@student.nitandhra.ac.in*,
622127@student.nitandhra.ac.in,
622127@student.nitandhra.ac.in,
622127@student.nitandhra.ac.in

Abstract:

The main focus of the following project is on general traffic scenario analysis and management with a custom-trained YOLOv8 Large-based model. Advanced capabilities of vehicle and pedestrian detection, real-time vehicle tracking, estimation of speed, and license plate detection with recognition are built in. This system leverages the powerful architecture of YOLOv8 Large for high-accuracy vehicle and pedestrian detection and identification in order to enhance safety and efficiency in urban scenarios. Real-time velocity estimation provides a means to better administer the flow of traffic, while accurate plate recognition at the stage of vehicle identification contributes to effective law enforcement. This project demonstrates the flexibility and efficiency of YOLOv8 Large by handling complex tasks concurrently, which makes it a scalable solution for intelligent transportation systems. Experimental results further reflect the effectiveness of the proposed model in varied environmental conditions, proving it feasible for real-life implementation.

Keywords:

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

1. Introduction:

During the past years, the convergence of computer vision and deep learning technologies has really accelerated some meaningful developments related to Intelligent Transport Systems, changing the way traffic scenarios were monitored, managed, and improved. This research work adapts the most recent state-of-the-art capabilities of the deep learning model YOLOv8 Large towards addressing some complex challenges in urban mobility and transportation safety.

In the real-time world of object detection, YOLOv8 Large is incomparable as long as the application requires rapid and accurate analysis of traffic dynamics. This, therefore, creates room for many ITS functionalities, considering that the model detects vehicles, pedestrians, and even reads license plates in varying environmental conditions.

The core building blocks of this project are real-time speed estimation—the information crucial for comprehension of vehicle behavior and complying with speed laws, which helps pave the way toward safer driving conditions and more efficient traffic flow—and high-accuracy license plate or number plate detection and recognition, making possible automated toll collection, tracking of vehicles, or the execution of many law enforcement operations.

In view of this, the project encompasses state-of-the-art functionalities within YOLOv8 and hence plans to prove its efficacy regarding the enrichment of urban mobility systems. The flexibility and scalability of YOLOv8 Large ensure reliable performances across a large variety of scenarios, from city intersection monitoring to highway monitoring. Ultimately, this research contributes to the further evolution of ITS, adding scalable solutions for 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 frame works 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:

Basically, YOLOv8 is an object detection algorithm that is highly efficient and accurate for real-time applications. At a high level of abstraction, the architecture of YOLOv8 still inherits core concepts from the original Yolo architecture-based single-shot detectors; here, the model predicts the bounding boxes and class probabilities by feeding the image full. Compared to traditional two-stage detectors, this approach brings faster speed. This project is oriented to demonstrate the efficacy and versatility of the YOLOv8 Large model in handling complex real-world traffic scenarios, attaining, with competitive performance, state-of-the-art levels of object detection tasks by leveraging inference times.

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.

Probably one of the most important things about YOLOv8 is anchor boxes—predefined shapes and sizes of bounding boxes in which to predict objects of interest. During training, these anchor boxes are adjusted to fit more toward the shape and aspect ratio of objects present in a dataset, increasing the model's localization capability.

As for training and optimization, YOLOv8 uses a special loss function that integrates localization loss and classification loss, which allows model parameters to be optimized based on how accurately the predicted bounding boxes are located and objects within them are classified. Other techniques, such as focal loss, can be applied to pay more attention to hard objects during training and further improve detection performance.

In particular, YOLOv8 has always been a real powerhouse in working performance and runs at high inference speed on both CPU and GPU platforms. This makes it quite suitable for deployment in applications requiring fast object detection, such as in autonomous driving systems, surveillance, and robotics. Furthermore, modularity allows researchers and developers to both adapt the architecture and tune the hyperparameters, but it also enables the integration of additional components in order to adapt models for special tasks and reach high performance on particular datasets. With the use of such deep learning and computer vision techniques, the proposed system will increase the potential of existing ITS frameworks by providing scalable solutions against the challenges of urban mobility.

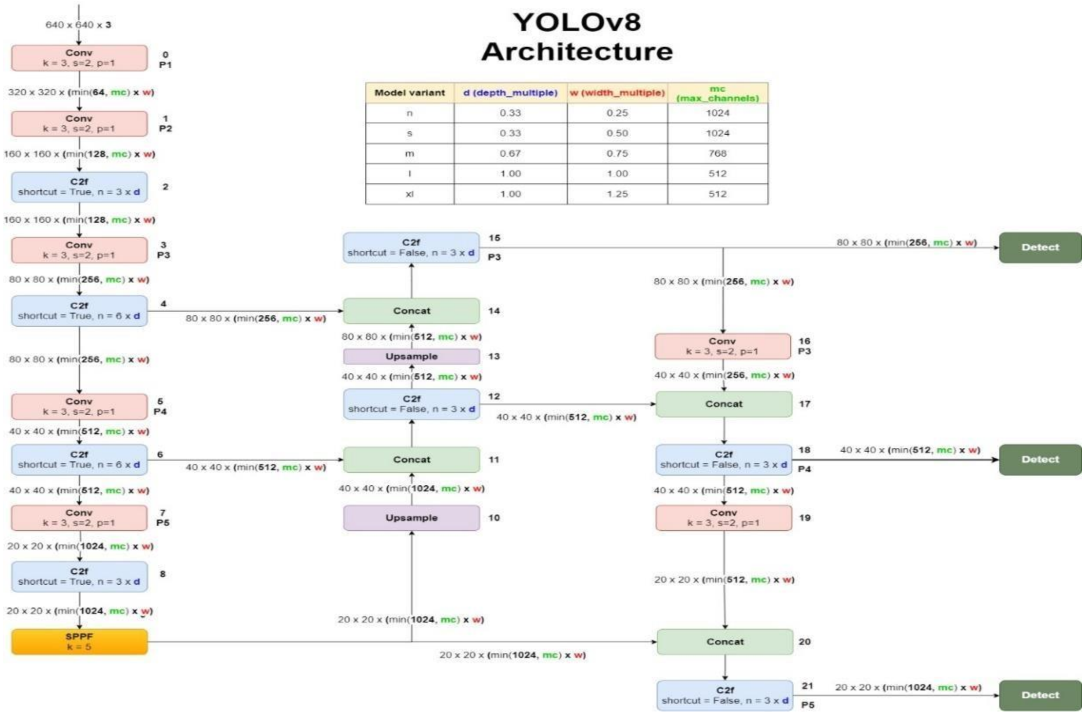

Fig: 1 Block Diagram of YOLOv8 Architecture

3.1 Conv Block:

It contains a batch normalization block with the SiLU activation. A Convolutional Layer (Conv Layer) of CNN extracts features from input data using small filters (kernels) which move along the input detecting a pattern. The stride controls the step size, while padding adjusts the size of the output. Adding nonlinearity is introduced through an activation function like ReLU. The result for each filter applied is a feature map, and by sharing parameters, the number of parameters is reduced.

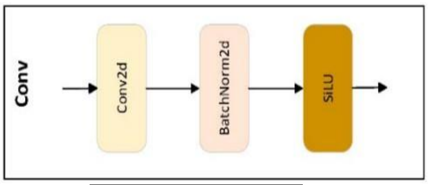

Fig: 2 Conv Block

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.

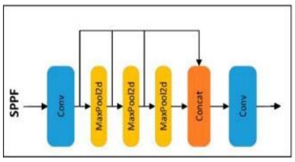

Fig: 3 SPPF Block

3.3 C2F Block:

Inside, it has a Conv block, a split block, and a combination of Bottleneck blocks. C2f, Cross Stage Partial Networks with C3 Fusion, promotes neural network efficiency and feature extraction. This fractionalizes feature maps, combines them to reduce computation, but keeps the features preserved through CSP (Cross-Stage Partial connections). Simultaneously, it merges the information from all the network stages into the features for better feature reuse and gradient flow using C3 Fusion.

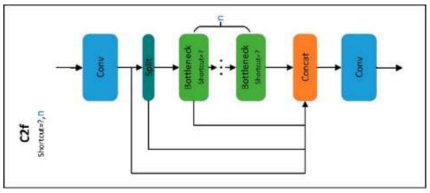

Fig: 4 C2f Block

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.

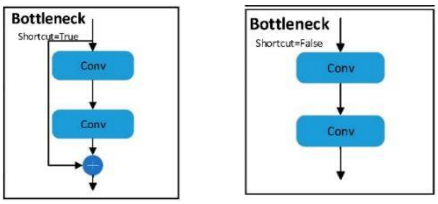

Fig: 5 Bottleneck Block

3.5 Detect Block:

Prediction of bounding boxes, objectness scores, and class probabilities of the objects left in an image by a YOLOv8 Detection block occurs in respect to anchor boxes. It refines predictions with Non-Maximum Suppression to have high-accuracy object detection at the final model stage.

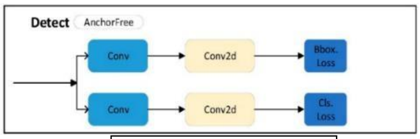

Fig: 6 Detect Block

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

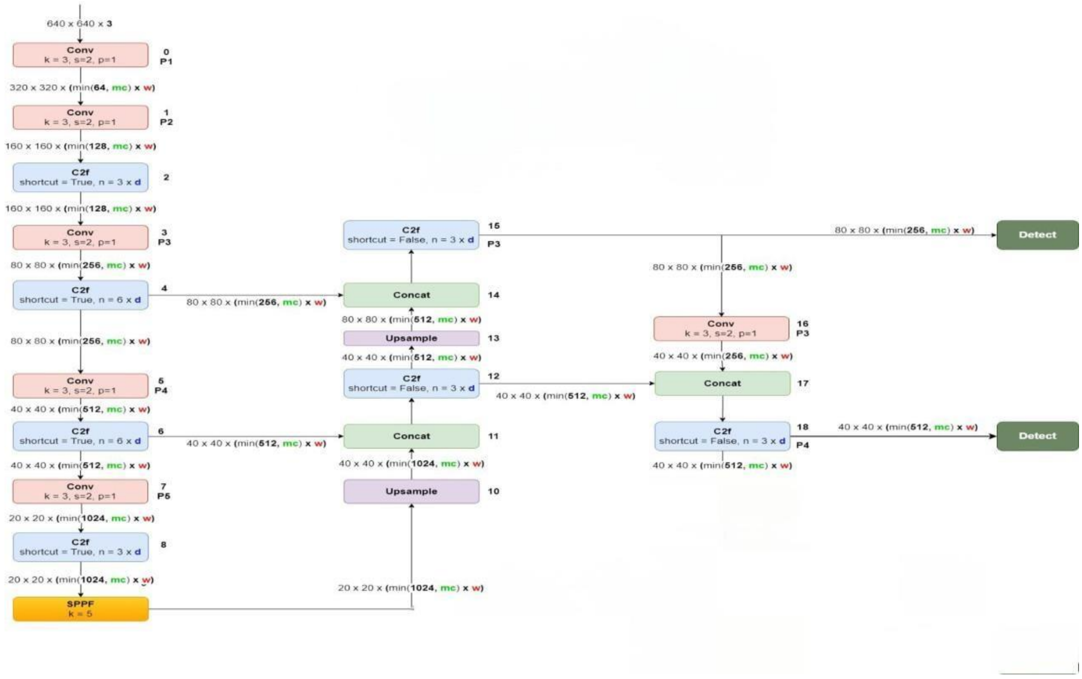

Fig: 7 Modified YOLOv8 Architecture

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 detect vehicles.

We evaluated the performance of our modified YOLOv8 model through rigorous training and validation on annotated datasets. Results showed that this offers a significant enhancement in the accuracy of localization and classification of smaller objects, very relevant especially for applications requiring fast and accurate object detection. The ethos further developed to make the model faster underscores adaptability to meet any challenges that our dataset may pose. This was an intended approach toward the design of YOLOv8's architecture, which corresponded to the nature of our dataset and further proved to be effective in real-world scenarios calling for 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.

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:

First, it detects the vehicles present in a video frame by object detection models such as YOLOv8. Then, the detected vehicles are further tracked along consecutive frames with state-of-the-art algorithms of ByteTrack. ByteTrack performs robust tracking by linking detections of the same vehicle across frames, tolerating occlusions and appearance changes.

5.2.2 Perspective Distortion and Transformation:

The position of moving vehicle changes with respect to the camera view, thus producing perspective distortion, under which vehicles closer to the camera view picture bigger and vice versa. These perspectives are reduced by using perspective transformation techniques. These techniques correct the perspectives and give a truer representation of the road distances; an example is homography.

5.2.3 Calculating Speed:

The system uses vehicle position information across consecutive frames to make an estimate of vehicle speed. That means that the detected vehicle positions are first normalized to a common view by applying perspective correction through homography. The system precisely measures the traversed distance a vehicle covers between frames on the road surface. The speed is calculated by dividing this distance by the known time interval between frames derived from the video frame rate.

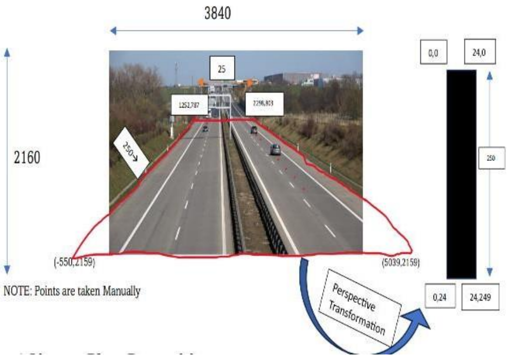

Fig: 8 Speed Estimation Technique

5.3 License Plate Recognition:

5.3.1 License Plate Detection:

The system extracts every frame of the video and detects, through computer vision techniques and deep learning models modified YOLOv8, the vehicles present in the scene. After detecting the vehicle, the region of interest containing license plate images will be the focus.

5.3.2 License Plate Localization:

First of all, the location of the license plate will be found with respect to the vehicle's bounding box. Image processing algorithms are then applied to isolate the plate area. This step ensures that just the area of the license plate is extracted for further analysis.

5.3.3 Character Recognition:

These alphanumeric characters on the license plate are extracted and interpreted with the help of OCR algorithms. Such algorithms are trained against large datasets of license plate images to accurately recognize and decode the characters in front, despite variations in font or size and differences in lighting conditions.

5.3.4 Data Storage and Visualization:

As the system processes each frame, it captures the detected license plate number in that frame with associated metadata of timestamp, vehicle ID, and tracking information in a CSV file or a database. If a frame is missing, a gap in the detection, or occlusions occur, the system fills the gaps for a continuous flow of the tracking data.

5.3.5 Output Visualization:

These license plate numbers, along with other tracked information if any, are output in real time either on the output video feed or as overlays. This kind of visualization could be helpful for monitoring tasks, like searching for a vehicle of interest or assisting in the implementation of some traffic regulations.

6. Results:

In the case of this research, an Intel Core i5 CPU at 2.60 GHz and an NVIDIA Tesla T4 GPU were used as a composition for the computational tasks. All programming was done in Python 3.8, with Jupyter Notebook, Google Colab, and PyCharm for develop -ment and experimentation. This setup was ideal for our study since it provided both efficiency and functionality.

6.1 InRia Dataset:

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

Table: 1 Results on Original YOLOv8

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

Table: 2 Results on Modified YOLOv8

InRia dataset is very popular dataset which is used to detect people it contains huge amount of data. We trained InRia Dataset with original YOLOv8 model and we trained with our modified YOLOv8 as well. Since our improved model can detect small object accurately we got 10% improvement in mAP(50) and 13% improvement in mAP(50-95) which is very clearly mentioned in Table 1 and 2

6.2 Highway Dataset:

Highway Dataset is a dataset which contains data of some Vehicles. We trained Highway dataset with both original and modified YOLOv8 model. Our model is got 7% of mAP(50) higher than original model which is clearly mentioned in Table 3 and 4

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

Table: 3 Results on Original 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

Table: 4 Results on Modified YOLOv8

6.3 Kitti Dataset:

Kitti Dataset is very famous dataset which is popularly used in vehicle detection. We trained this dataset with both original model and our modified model. We got 6% improvement in mAP(50) compared to original model and 9% improvement in mAP(50-95). Results are detailly mentioned in Table 5 and 6

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

Table: 5 Results on Original 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

Table: 6 Results on Modified YOLOv8

6.4 Visdrone Dataset:

Visdrone Dataset is vehicles dataset which is frequently used in vehicle detection and self-driving applications. We trained this model with both original and our modified model we got improvement of 12% in mAP(50) and 19% in mAP(50-95) compared with original model. Detailed results in mentioned in Table 7 and 8

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

Table: 7 Results on Original 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

Table: 8 Results on Modified YOLOv8

6.5 License Plate Dataset:

License Plate Dataset is used to detect the license plate of vehicles. We used this dataset to detect and recognize the license plate. We trained this dataset with original and our modified model. We got 3% improvement in mAP(50) and 5% improvement in mAP(50-95). Detailed results are given in Table 9 and 10

Class	Images	Instances	mAP(50)	mAP(50-95)
All	2046	2132	0.96	0.71

Table: 9 Results on Original YOLOv8

Class	Images	Instances	mAP(50)	mAP(50-95)
All	2046	2132	0.98	0.75

Table: 10 Results on Modified YOLOv8

Fig:

6.6 Stats (Best Model):

1. Highway Dataset:

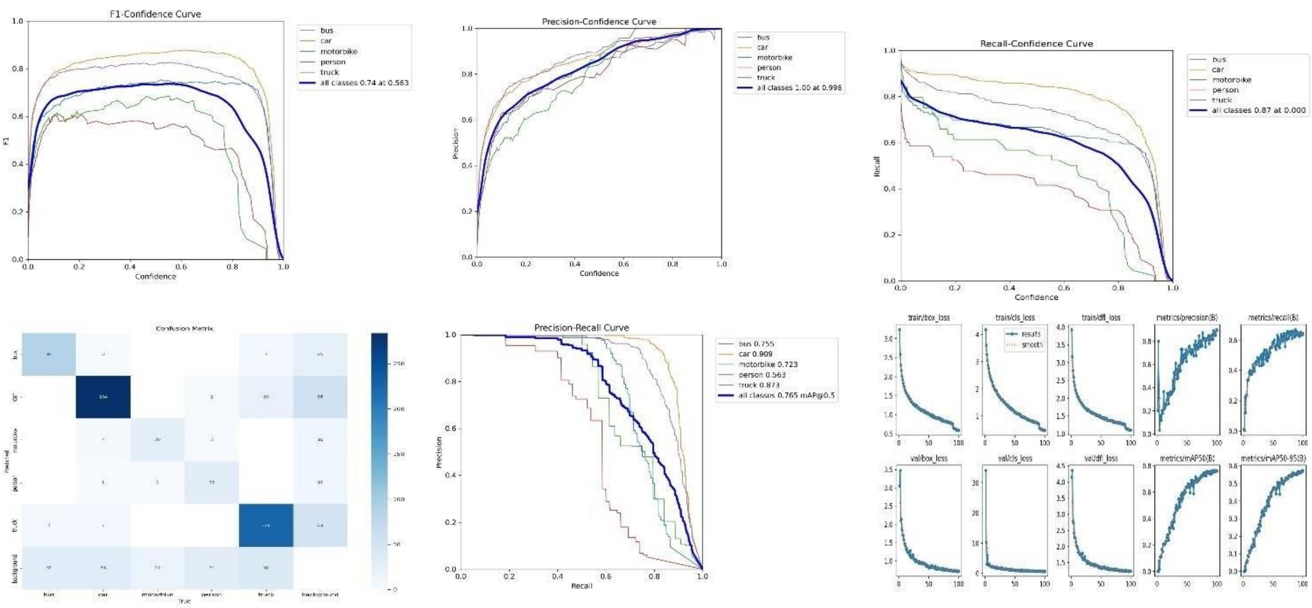

Fig:9 Above are the Stats and Analysis of the our best model(Highway)

This image shows several classifications for the model's performance. The F1-Confidence, Precision-Confidence, and Recall-Confidence curves all result in pointing to the best model performance at a "confidence" level of 0.563%, yielding an overall F1 score of 0.74. The best class is car, while person is the worst. The accuracy for both cars and trucks is very high as can be observed from the confusion matrix. The Precision vs. Recall curve results in a mAP of 0.765 overall. Training and validation loss, as well as precision-recall curves, both demonstrate that the model can learn effectively and converge properly. In Confusion Matrix point of view True positives are most important thing to observe. A True positive is an outcome where the model correctly predicts the positive class. True Positives of Bus, Car, Motorbike, Person, Truck are 87, 284, 26, 30, 226 respectively.

2. InRia Dataset:

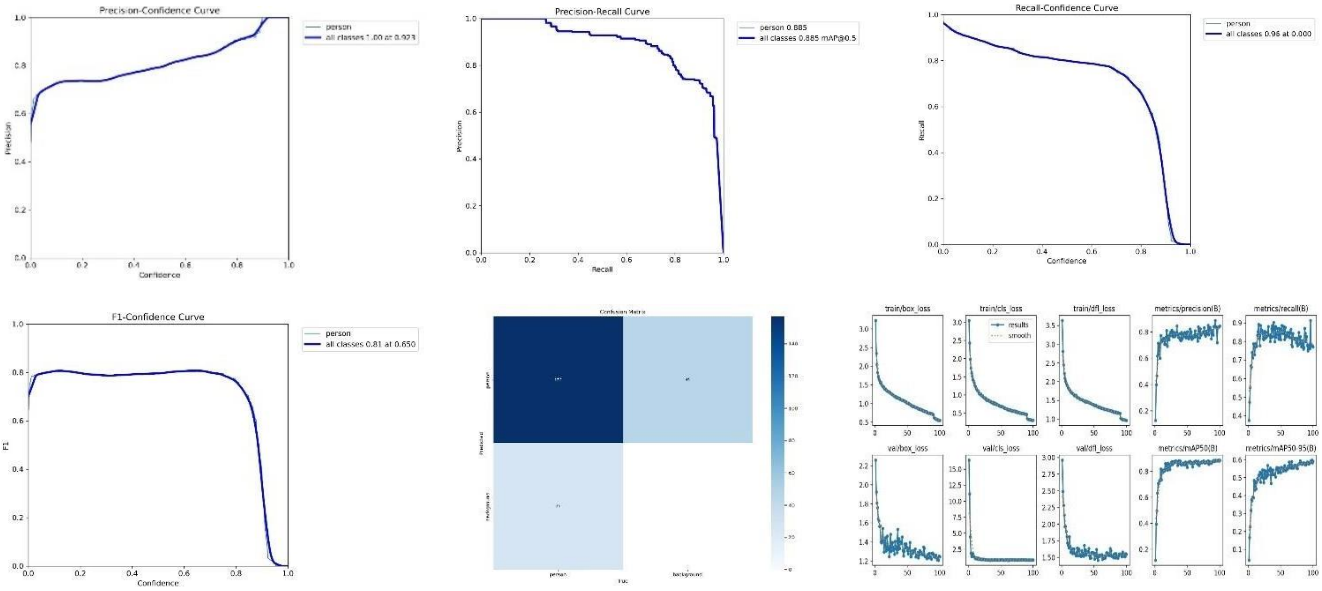

Fig:10 Above are the Stats and Analysis of our best model (InRia)

It plots some performance metrics for a model trained to classify the person category. The peak of the precision-confidence curve is 1.0 with a confidence level of 0.923, while for the F1-confidence curve, it peaks at 0.810 with a confidence level of 0.650. The precision-recall curve provides an mAP of 0.885. The confusion matrix presented will yield very high accuracy in person detection. The training and validation loss curves paired with precision and recall metrics indicate that the model was effectively trained and converged. True positives of Person from Inria dataset is 157 respectively.

3. License Plate Dataset:

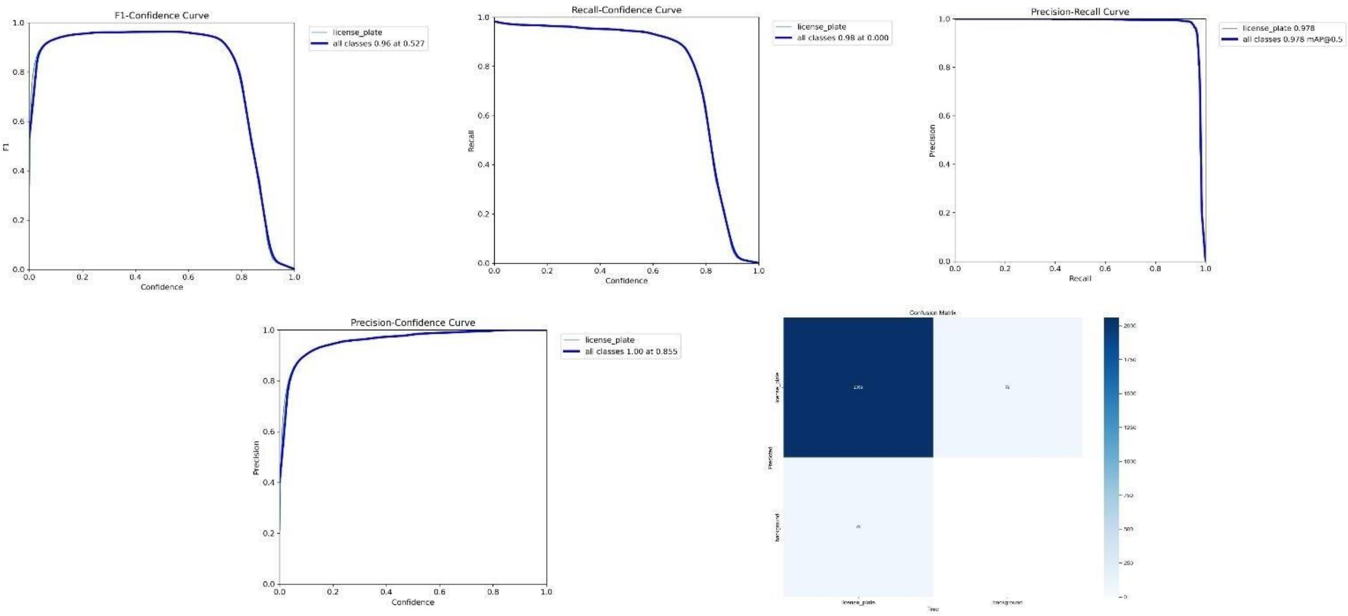

Fig :11 Above are the Stats and Analysis of our best model(License Plate)

This image presents performance metrics for a License Plate Recognition system. The Peak of the F1-Confidence Curve is at a middle confident threshold, while the Recall-Confidence Curve illustrates great recall when given low confidence, tapers off strongly after 0.8. Precision-Recall Curve: High precision and high recall with an overall mAP of 0.978. High values are near 0.978 The Precision-Confidence Curve peaking around 0.85 Conf. Confusion Matrix Showing a strong system w/ very few misclassifications, high true positive and true negative rates. True positives of license plate from our training is 2059 repectively.

6.7 Detections:

Fig :12Here are the Validation examples of Highway dataset

Fig 12 shows the detection results of our model which is trained on Highway dataset using proposed YOLOv8 architecture. We got decent results in this particular model. Accuracy also improved around 7% compared to original model.

Fig:13 Above are Validation examples of InRia dataset

Fig 13 are the detection results of our model which is trained on inRia dataset using proposed YOLOv8 architecture. We got around 10% improvement of accuracy compared to original YOLOv8 architecture. And Fig 14 are the detection results on kitti dataset as shown in figure we got good results and 6% improvement in accuracy compared to original YOLOv8 architecture.

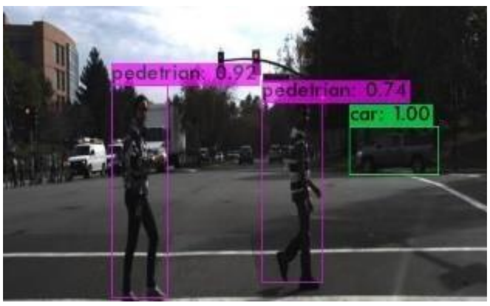

Fig 14: Above are Validation results of Kitti dataset

Fig :15 Above are validation examples of visdrone

Fig 15 shows the detection results of our model which is trained on visdrone dataset using proposed YOLOv8 architecture. We got decent results in this particular model. Accuracy also improved around 10% compared to original model.

Fig :16 Above are the validation examples of License plate

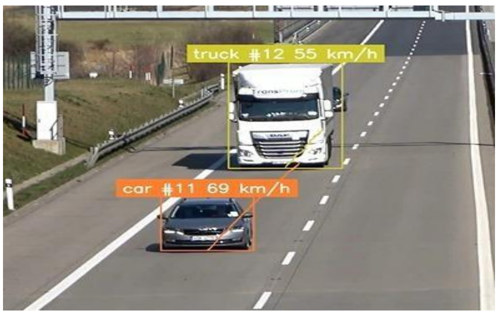

Fig :17 Speed Estimation

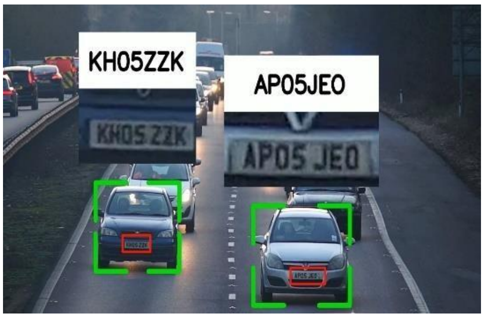

Fig :18 License Plate Recognition

Fig 16 are the Detections results we got with license plate dataset build by proposed model.in terms of accuracy we got 15% increment compared with original architecture of YOLOv8. Fig 17 is the speed estimation example which is designed by the help of Percpetive Transformation which is clearly shown in Fig 8. And Fig 18 is Licence plate recognition using easyOCR which is a python library used to extract the data from an image.

6.8 Comparitive analysis:

Fig 19 shows the comparison of mAP(50) between the model which is trained on original YOLOv8 architecture and the model which is trained on our proposed YOLOv8 architecture. This comparison analysis includes all results of datasets(Highway, InRia, Kitti, Visdrone, Licence Plate).

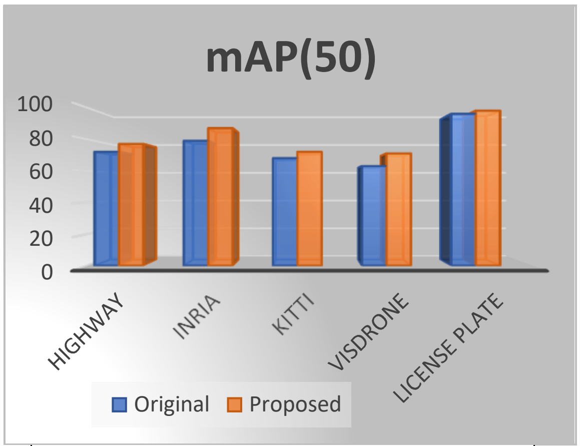

Fig: 19 Comparison Report of mAP(50)

Fig 20 is the comparison report of mAP(50-95) between the model which is trained on original YOLOv8 architecture and the model which is trained on our proposed YOLOv8 architecture. This comparison analysis includes all results of dataset(Highway, InRia, Kitti, Visdrone, License Plate).

7. Conclusion and Future Work:

In summary, we efficiently tuned the YOLOv8 model for small to medium-sized objects, bringing about a good boost in speed and accuracy for real-time applications. Targeted removal of layers responsible for large object detection has leanified the architecture to be more compliant with our dataset characteristics. It has such refinement that not only brings about significantly higher computational efficiency but greatly enhances model precision in particular object detection and classification tasks.

Advanced techniques, including perspective transformation and robust algorithms for tracking, allowed our system to perform accurate speed estimation. Correcting perspective distortions and correctly following vehicle movements across frames delivered reliable calculations necessary for tasks such as traffic monitoring and safety enforcement.

Future work would most likely be focused on further refinement of the YOLOv8 model through continuous training using diverse datasets for improved detection robustness across varying environmental conditions. This would involve methods for advanced speed estimation, multi-

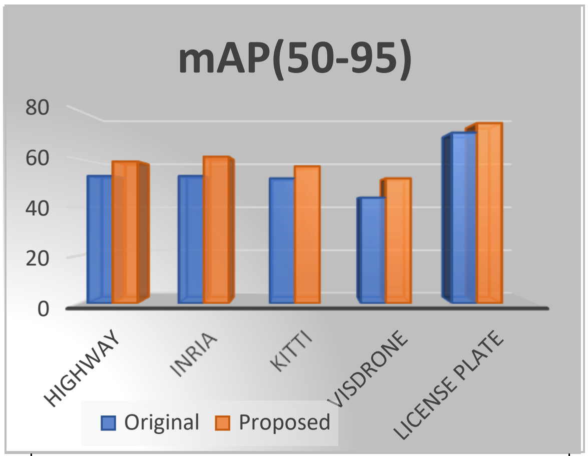

Fig: 20 Comparison Report of mAP(50-95)

sensor data fusion, and refinement algorithms for practical deployment. Furthermore, the enhancements necessary for the accomplishment of multi-modal object detection using the system, and the user-friendly interfaces for operation in real-world scenarios, are priority considerations in efforts aimed at enhancing general system efficiency and usability beyond smart city applications.

8. References:

1. Sen Zhang, Shuai Chen, Jie Li, Huichen Zhang An Improved Vehicle-licensePlate 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, KArulprasath, Manesh Kokare ForeignObject 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