Real-World Traffic Detection: Achieving High Accuracy using Deep Learning based YOLOv5 and YOLOv8 Architectures

Rahul Pavithran
Dept. Of Computer Science
New Jersey Instiute of Technology
Newark, NJ, USA
rp999@njit.edu

Jinal Jagdishkumar Thakkar
Dept. Of Computer Science
New Jersey Instiute of Technology
Newark, NJ, USA
jt644@njit.edu

Sunit Sanjay Shirke
Dept. Of Computer Science
New Jersey Instiute of Technology
Newark, NJ, USA
ss4945@njit.edu

Vaishnavi Masapalli
Dept. Of Computer Science
New Jersey Instiute of Technology
Newark, NJ, USA
vm599@njit.edu

Arashdeep Kaur
Dept. Of Computer Science
New Jersey Instiute of Technology
Newark, NJ, USA
arashgulati@gmail.com

***Abstract*— In numerous nations, the imperative role of traffic monitoring systems is essential for overseeing and controlling vehicular and pedestrian traffic. In recent years, various techniques have been presented for automated detection to optimize traffic. The methods presented in the literature have their own pros and cons. This paper proposes two different traffic detection models using YOLOv5 and YOLOv8. In addition, this paper proposes an efficient data pre-processing algorithm to achieve better accuracy for detecting various classes of vehicles in the traffic including pedestrians. An efficient loss optimization strategy is proposed and adopted while training the model to reduce the training loss. This paper discusses the choice between two deep learning models, YOLOv5 and YOLOv8, for identifying different types of objects of interest on the road in urban areas. The efficiency of the proposed models is evaluated in this paper using multiple performance metrics, including their accuracy. The comparative analysis of the proposed models with existing models indicate that the proposed models are on par in terms of accuracy with existing strategies while integrated with the additional complexity of pedestrian detection.**

Keywords*— *deep learning, neural networks, traffic detection, yolo, object detection models, traffic monitoring systems.

I. INTRODUCTION

In the rapidly evolving field of artificial intelligence and computer vision, traffic detection systems are pivotal in advancing intelligent transportation systems. An important factor in a nation's ability to advance economically and improve quality of life is the efficiency of its road infrastructure. In numerous nations, traffic monitoring systems are essential for overseeing and controlling vehicular and pedestrian traffic at crucial road infrastructure locations, ensuring the uninterrupted operation of logistics and related businesses. Typically, they are made up of regulated traffic lights, sensors, and video feeds that are all coordinated by a central processing system to allow routing and control decisions to be made based on the state of traffic. This study delves into the implementation and comparative analysis of two sophisticated object detection models, YOLOv5 and YOLOv8, for the task of identifying various entities of interest on the road via an image or camera feed. Emphasizing the balance between accuracy and speed, the study aims to evaluate these models under varied traffic scenarios, contributing to the optimization of traffic monitoring and management systems. Through this analysis, this paper seeks to illuminate the practical strengths and limitations of these models, offering a comprehensive perspective beneficial for future advancements in the realm of intelligent transportation, which deal with the care of many lives daily. The dataset used in this study has 6,633 structured images from various locations, including Turkey's Bursa, Istanbul, and Konya as well as from numerous other nations. Sample images from this dataset can be observed in Figure 1.

(a) (b)

Fig. 1. (a) and (b): Sample images from the dataset

This study aims to provide valuable insights into the practical implications of choosing between the aforementioned object identification models for traffic detection systems, which would aid decision-makers in traffic system design by offering guidance on selecting the most appropriate model based on specific priorities, whether it be speed, accuracy, or a combination of both. This work contributes to the broader understanding of how these object detection methods can be strategically employed in the development of efficient and responsive traffic management systems, addressing needs such as real-time traffic monitoring and accident detection.

The rest of the paper is organized as follows: Section II provides a detailed literature survey of various existing methods for traffic detection systems. Section III gives a detailed overview of the dataset used for the training and testing of the proposed models. Section IV gives the proposed methodology, and the experimental results are provided in

Section V. The exhaustive comparative analysis is given in Section VI and finally Section VII concludes the paper.

II. LITERATURE REVIEW

The healthy activity of research and development on detection systems for traffic monitoring has well-endorsed its necessity in maintaining public safety and economic growth. S. Sanjana et al.[1] have examined various approaches that have been employed and how they have changed over time to produce better results, increasingly relying on machine learning. Y. Li et al.[2] introduced a fuzzy theory-based method for traffic anomaly detection, achieving 93.40% accuracy. Y. Wei et al.[3] combined Harr and HOG features in a two-step detection approach. Q. Hu et al.[4] used the Viola-James framework and a modified AdaBoost classifier, attaining 76.66% precision in challenging detections. F. Faisal et al.[5] suggested Haar Cascade Classifiers for real-time traffic detection, requiring extensive training datasets.

The integration of deep learning has significantly advanced object detection research. J. A. Smitha et al.[6] developed a two-phase vehicle detection method combining HOG descriptors with an Optimum Feedforward Neural Network and an Improved Grasshopper Optimization Algorithm, achieving superior accuracy over previous techniques. Convolutional Neural Networks (CNNs) revolutionized image feature extraction in Computer Vision. S. Karungaru et al.[7] improved CNN-based extraction and combined CNN-SVM for increased generalization, all resulting in a compact, faster, and more accurate system for real-world applications. C. -C. Tsai et al.[8] and L. Chen et al.[9] enhanced CNN models with optimized architectures and occlusion awareness, achieving superior performance on specialized datasets.

The exponential research growth in deep learning over the past decade showed promising innovations in neural network architecture. H. Gao et al.[10] combined a feature fusion module with Single Shot Multi-box Detection (SSD) for accurate and speedy results. R. Theagarajan et al.[11] proposed the EDEN (Ensemble of Deep Networks) vehicle classification system that consists of an ensemble model of three ResNet-50 networks, excelling in noisy data processing. Z. Li et al.[12] proposed the Denoise AutoEncoder-Generative Adversarial Network (DAE-GAN) combining denoising autoencoders to generate pseudoanomalies, enhancing semi-supervised detection models and reducing training data needs.

Q. C. Mao et al.[13], K. Khazhukov et al.[14], S. Yang et al.[15] and C. Chen et al.[16] proposed detection systems based on the YOLOv3 model, an older iteration of the YOLO series of object detection models that are known for their speed due to their ability to detect objects within a single pass over an image. V. Singh et al.[17] innovated with YOLOv7 and UAV technology, significantly advancing aerial vehicle detection, classification, and tracking for improved traffic management and analysis.

III. DATASET DESCRIPTION

The Traffic Detection Project[18] dataset used for model design and analysis in this study, encompassing 6633 images from diverse global traffic, lighting, and weather conditions, serves as the foundation for this study's model design and analysis. It includes 5805 training, 549 testing, and 279 validation RGB images, all standardized to 640x640 pixels and rigorously preprocessed for noise reduction. The dataset, maintained under the CC BY-NC-SA 4.0 license, underwent auto-orientation, resizing, and quality checks for annotation accuracy, exemplified in Figures 2(a) and 2(b), showcasing the effectiveness of the denoising process (Algorithm 1).

This paper's primary objective is to classify four vehicle types – bus, car, bicycle, and motorbike – and to identify pedestrian traffic, both stationary and walking. This multifaceted focus adds complexity to the model's design.

(a) (b)

Fig. 2. (a) Original Sample Image (b) Denoised Image

Figure 3 gives a representation of five randomly sampled images from the dataset chosen and their corresponding histograms. Analysis indicates that dataset contains images of various background and lighting conditions that provide a diverse set of properties to improve the generalization ability of the model.

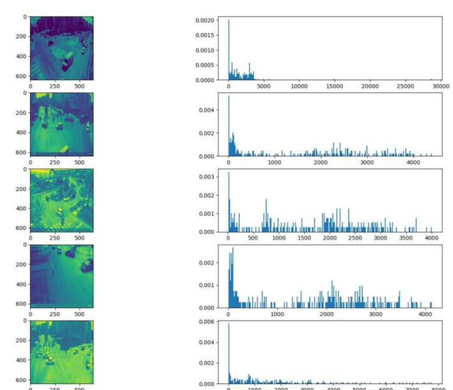


Fig. 3. Histograms of Sample images

IV. PROPOSED METHODOLOGY

This section outlines two proposed models for precise traffic prediction: YOLOv5, the fifth iteration of the YOLO (You Only Look Once) series, and YOLOv8. Key differences are in the feature fusing block design and the use of a decoupled head. The framework of these models is detailed in Figure 4.

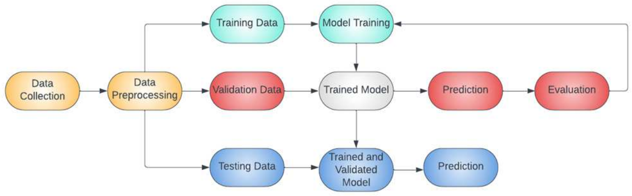


Fig. 4. Block Diagram of Proposed Models

Data Preprocessing: Preliminary analysis of the images provided a basic framework for classifying the objects in the images into one of five broad categories: bicycle, bus, car, motorbike, and person. For each image, labels and normalized bounding box coordinates (x, y, w, h) were created, with x and y denoting the object's center and w and h its bounding box dimensions. This classification formed the basis for individual label text files that were prepared corresponding to each image. Image dataset analysis indicated a significant proportion exhibiting salt-and-pepper noise, which is a form of noise on images characterized by sparsely occurring black and white pixels, potentially impairing model accuracy. A specialized preprocessing unit, depicted in Figure 5, was implemented to mitigate this issue.

Algorithm 1: Pre-Op =Data-Preprocessing (Ip)

IP: Input Image
Pre-Op: Output of Algorithm 1 – Pre-processed image

Step 1*: Ip = Input Image*

Step 2*: Initialize p=0 and q = 0*

Step 3*: while p and q <= size(Ip)*

Op(p, q) = Mdn(Nbd(Ip(p,q)))

where, (p, q) represent a pixel value

Nbd denotes the pixel values in the neighborhood of (p, q)

Mdn(x), returns the middle value within a set of values.

Step 4*: Initialize m=0 and n=0*

Step 5*: while m and n <= size(Op)*

Pre-Op(m, n) = Op(m, n) + Weight × Laplacian(Op(m, n))

where, (m, n) represent a pixel value.

Laplacian(Op(m, n)) signifies the Laplacian of the pixel value at (m, n).

Weight, is used to control the strength of the enhancement.

Step 6*: Display pre-processed Image and exit.*

As detailed in Algorithm 1, Step 3 reduces the effect of salt-and-pepper noise, also known as impulse noise, by replacing each pixel value in an image with the median value of its neighboring pixel values. While this equation in Step 3 makes a significant correction to the pixels affected by noise, it introduces a side effect of pixel smoothening on the details of the original image. To reduce the impact of smoothening on edge detection by the proposed models, Step 4 was performed on all the images in the dataset. This enhancement works by improving the contrast along the edges in the image, thus making them more prominent. This helps the model in improved edge detection, which further extends into better identification of higher-level features. The results of execution of Algorithm 1 are demonstrated using Table I.

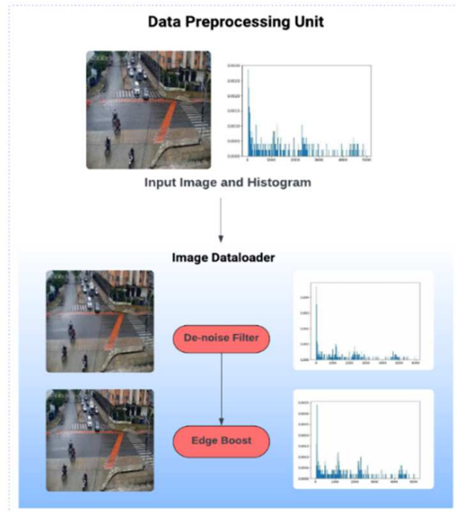


Fig. 5. Data preprocessing unit

Model Training

In this study, YOLOv5 and YOLOv8 are employed for training a traffic prediction model. These models segment the image into a grid, focusing on configuring grid cells to encompass an object. Object detection involves generating a bounding box around the grid cells containing the object, with the object's center as a reference. The architectural frameworks of YOLOv5 and YOLOv8 are depicted in Figure 6 and Figure 7, respectively.

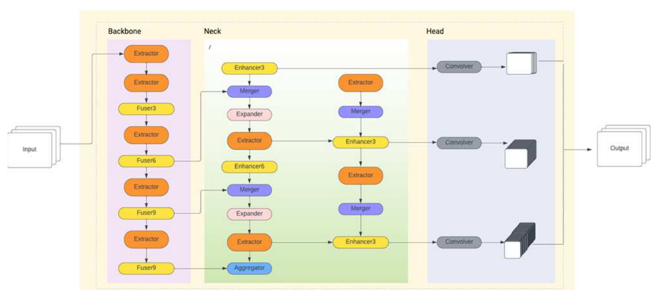


Fig. 6. YOLOv5 Model Architecture

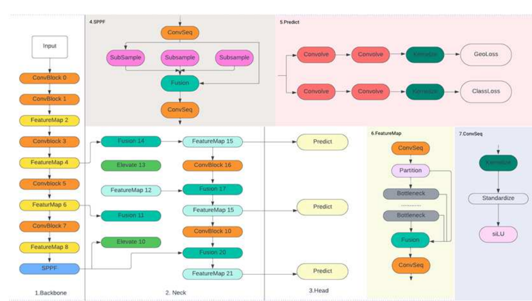
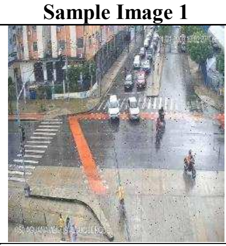
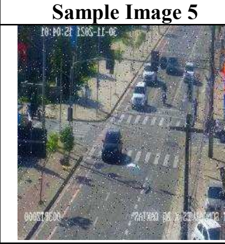
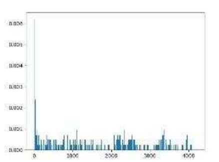
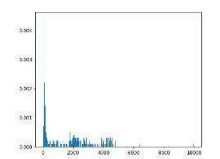
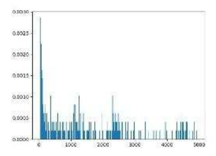
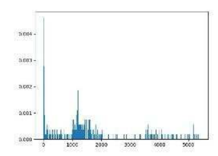
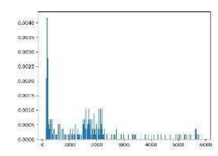
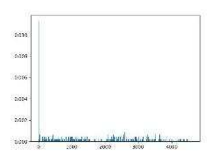
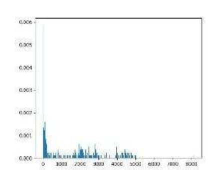
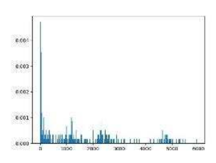
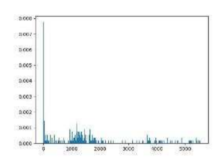
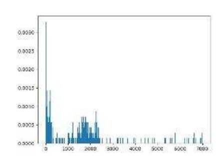
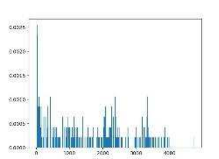
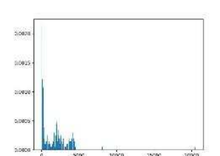
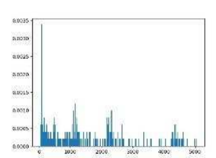
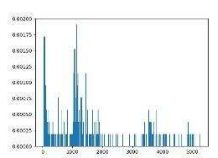
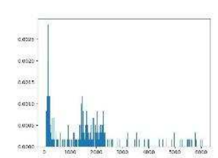


Fig. 7. YOLOv8 Model Architecture

YOLOv5 features a backbone structure based on Darknet53 for feature extraction, a neck comprising Enhancers and Mergers for feature augmentation, and a head with Convolvers for detection. This architecture is optimized for efficient and precise object detection, making it suitable for diverse applications from detection to automation.

YOLOv8, advancing the YOLO framework, improves upon a sophisticated architecture with a Convolutional Block for multi-scale feature extraction and Spatial Pyramid Pooling Fusion (SPPF). The refined 'Neck' and 'Head', equipped with Fusion, Elevation, Convolutional Sequences, Kernelization, Standardization, and SiLU activation functions, enable unparalleled accuracy and performance.

Table I: Few sample images with preprocessing results after the execution of Algorithm 1 on the images dataset

Step	Sample Image 1	Sample Image 2	Sample Image 3	Sample Image 4	Sample Image 5
Original Image					
Histogram _Original _Image					
Output of Step 3 (Op)					
Histogram (Op)					
Output of Step 4 (Pre-Op)					
Histogram (Pre-Op)					

YOLOv5, while fast and generally accurate, showed limitations like misclassifying a ‘person’ as a ‘motorbike’ with high confidence. Figure 8 provides a parallel comparison of the object detection results generated by the proposed models. The output of YOLOv8, depicted by Figure 8(b) shows higher confidence scores in its class prediction than those obtained from YOLOv5, as shown in Figure 8(a). Table II provides the description of various hyperparameters used for training these models. Algorithm 2 provides the basic framework of loss evaluation in the proposed model.

(a) (b)

Fig. 8. (a): YOLOv5 object detection on dataset sample, (b) YOLOv8 object detection on dataset sample

Table II: Model training parameters

Scale	Epoch	Batch Size	Image Size	Optimizer
n (Nano)	60	16	640x640	AdamW

Algorithm 2: Loss_Evaluation (Class, Coordinates)

Input: Class: *Predicted Class of vehicle, Ground Truth Class,* ***Coordinates:*** *Predicted Bounding Box Co-ordinates(x,y,w,h), Ground Truth Co-ordinates*
Output: Total Loss

Step 1*: Ip = Input (c, c' (x, y, w, h), (x', y', w', h'))*
Step 2*: Initialize cls_loss = 0, obj_loss = 0, ciou_loss = 0 and total_loss = 0*
Step 3*: cls_loss =* $\sum_{f=1}^{F} \mathrm{cf} * \log(\mathrm{c'f})$
Step 4: *ciou_loss = IoU –* $\frac{P^2}{Q^2}$
Where P is distance between predicted and ground truth bounding box centers, and Q is the diagonal length of the overlapping box region.
Step 5: *if model == 'yolov5'obj_loss = c * log(c) + (1 – c') * log(1 – c') else*
*obj_loss = (alpha * (1 – p_c))^(s * log(p_c))*
where alpha and s are hyperparameters controlling the precision-recall tradeoff and loss function sharpness respectively. p_c indicates the estimated probability of ground truth class c.
Step 6: *total_loss = w_ciou * ciou_loss + w_obj * obj_loss + w_cls * cls_loss*
where w_ciou, w_obj and w_cls are hyperparameters that control the weightage of the loss values.
Step 7: *Back Propogate total_loss to the model*

V. Experimental Results

This section presents the experimental results of the proposed models, each outputting bounding boxes with class labels and confidence scores. Validation set analysis led to confusion matrices for both models, delineating True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN), reflecting the accuracy of model predictions against ground truth. These matrices informed the assessment of five major performance metrics, facilitating a comparative analysis of the models' effectiveness.

1. **Accuracy**: It is the ratio of the correct predictions made by the model to all its predictions made on the input data as given in eq. 1.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

2. **Precision:** It is the ratio of true positives to the sum of true positives and false positives. It is a measure of the accuracy of positive predictions and is described in eq. 2.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. **Recall**: It is the ratio of true positives to the sum of true positives and false negatives. It measures how well the model captures all the positive instances as shown in eq. 3.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

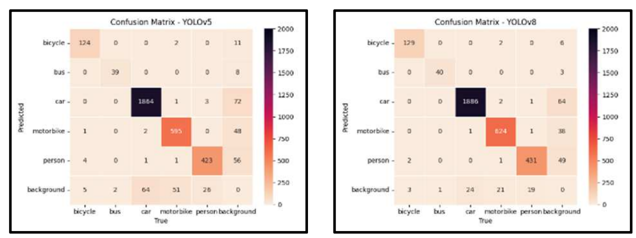
4. **Mean Average Precision (mAP):** It is the mean of Average Precision (AP) values across multiple classes. AP measures the precision-recall trade-off for each class. mAP values range from 0 to 1, with higher values indicating better performance. It is computed as given in eq. 4.

$$\mathrm{mAP} = (1/\mathrm{N}) * \sum \mathrm{AP(c)} \quad (4)$$

5. **F-score:** It is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. F-score values also range from 0 to 1, with 1 being the best, refer eq. 5.

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5)$$

Figure 9 presents the confusion matrices for (a) YOLOv5 and (b) YOLOv8, respectively, summarizing the counts of true positives, true negatives, false positives, and false negatives, thereby providing a comprehensive evaluation of each model's performance.



(a) (b)

Fig. 9. Confusion Matrices of (a) YOLOv5 and (b) YOLOv8

Table III summarizes the results of the performance evaluation of YOLOv5 and YOLOv8 models employed in this study. These results concur with the observations made in Section IV indicating that YOLOv8 provided better prediction results with respect to YOLOv5. Table IV gives the training iteration history of the YOLOv5 and YOLOv8 models proposed in this paper. It can be seen from the table that YOLOv8 has a faster detection speed, making it more suitable for real-time frame-by-frame detection in a video feed.

Table III: Performance metrics of proposed models

Model	Precision	Recall	mAP	F-Score	Accuracy
YOLOv5	0.941	0.867	0.721	0.834	89.48%
YOLOv8	0.928	0.893	0.732	0.847	92.89%

Table IV: Training iteration history for YOLOv8 model

Model	YOLOv5						YOLOv8					
Iteration	1	2	3	4	5	6	1	2	3	4	5	6
No. of Epochs	10	60	70	80	90	100	10	60	70	80	90	100
Is Data Corrected	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Accuracy %	73.84	80.05	82.39	84.73	87.07	89.48	75.32	82.13	85.03	87.95	90.90	92.89
Precision	0.866	0.912	0.915	0.917	0.925	0.932	0.883	0.930	0.933	0.935	0.944	0.955
Recall	0.805	0.856	0.859	0.878	0.885	0.901	0.821	0.878	0.886	0.911	0.924	0.942
mAP	0.614	0.685	0.691	0.712	0.715	0.736	0.626	0.703	0.713	0.739	0.746	0.786
Weights File Size (kB)	5147	5153	5159	5165	5171	5177	6084	6091	6098	6105	6112	6119

VI. COMPARATIVE ANALYSIS

This section explains comparative analysis of various proposed models. Table V offers a comprehensive view of accuracy and precision scores, guiding comparisons across diverse methodologies.

Table V: Comparison of Proposed Models with existing Techniques in terms of Accuracy

Reference	Methodology	Accuracy (%)	Precision
Li et al.[2]	Fuzzy Logic	93.40	NR
Wei et al.[3]	Harr & HoG AdaBoost	NR	97.96
Hu et al. [4]	Shrinkage AdaBoost	NR	94.55
Faisal et al.[5]	Haar Cascade	69	NR
Smitha et al. [6]	HoG-based OFFNN	91.93	NR
Karungaru et al.[7]	CNN-SVM	98.87	NR
Tsai et al. [8]	CNN	90.3	94.0
Chen et al. [9]	Occlusion-aware CNN	NR	87.16
Gao et al.[10]	YOLOv3	NR	90.4
Theagarajan et al. [11]	ResNet Ensemble	97.80	94.39
Li et al.[12]	DAE-GAN	NR	98.6
Mao. et al. [13]	YOLOv3	86.81	90.22
Khazukov et al.[14]	YOLOv3 + SORT	NR	89.33
Chen et al.[16]	YOLOv3	92	NR
Singh et al.[17]	YOLOv7	93	93
Proposed Model	**YOLOv5**	**89.48**	**93.2**
Proposed Model	**YOLOv8**	**92.89**	**95.5**

* NR: Not Reported

The comparative study of the proposed model with various existing models based on accuracy and precision indicate that the proposed models deliver on par performance with existing methodologies even with the integration of added complexities such as the detection of pedestrian and two-wheeler entities in on-road environment.

VII. CONCLUSION

This study conducts a comparative analysis of YOLOv5 and YOLOv8 in traffic detection systems, highlighting YOLOv5's "lighter precision" in object localization, beneficial in scenarios with limited computational resources or when balancing accuracy with efficiency. In contrast, YOLOv8 excels in detection speed and accuracy, making it ideal for real-time applications like traffic monitoring and accident detection. The insights guide the selection of appropriate object detection methods for varying traffic management needs, suggesting YOLOv8 for situations requiring swift and precise responses, and YOLOv5 where scene complexity is higher but immediate response is less critical. Future research scope includes evaluation using other object detection deep learning models and diversifying the dataset for better localization.

VIII. REFERENCES

[1] S. Sanjana, V. R. Shriya, V. Gururaj, and K. Ashwini, “A review on various methodologies used for vehicle classification, helmet detection and number plate recognition,” Evolutionary Intelligence, vol. 14, no. 2, pp. 979–987, Sep. 2020, doi: 10.1007/s12065-020-00493-7.

[2] Y. Li, T. Guo, R. Xia and W. Xie, "Road Traffic Anomaly Detection Based on Fuzzy Theory," in IEEE Access, vol. 6, pp. 40281-40288, 2018, doi: 10.1109/ACCESS.2018.2851747.

[3] Y. Wei, T. Qiu, J. Guo, W. Huang, and J. Cao, “Multi-vehicle detection algorithm through combining Harr and HOG features,” Mathematics and Computers in Simulation, vol. 155, pp. 130–145, Jan. 2019, doi: 10.1016/j.matcom.2017.12.011.

[4] Q. Hu, S. Paisitkriangkrai, C. Shen, A. van den Hengel and F. Porikli, "Fast Detection of Multiple Objects in Traffic Scenes With a Common Detection Framework," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1002-1014, April 2016, doi: 10.1109/TITS.2015.2496795.

[5] F. Faisal, “Automated Traffic Detection System Based on Image Processing”, JCSTS, vol. 2, no. 1, pp. 18–25, Jun. 2020.

[6] J. A. Smitha and Rajkumar, “Optimal feed forward neural network based automatic moving vehicle detection system in traffic surveillance system,” Multimedia Tools and Applications, vol. 79, no. 25–26, pp. 18591–18610, Mar. 2020, doi: 10.1007/s11042-020-08757-1.

[7] S. Karungaru, L. Dongyang, and K. Terada, "Vehicle Detection and Type Classification Based on CNN-SVM," *International Journal of Machine Learning and Computing*, vol. 11, no. 4, pp. 304-310, 2021.

[8] C. -C. Tsai, C. -K. Tseng, H. -C. Tang and J. -I. Guo, "Vehicle Detection and Classification based on Deep Neural Network for Intelligent Transportation Applications," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 2018, pp. 1605-1608, doi: 10.23919/APSIPA.2018.8659542.

[9] L. Chen, Y. Ruan, H. Fan, H. Zhu, X. Chen, and Q. Chen, “Occlusion-Aware detection for internet of vehicles in urban traffic sensing systems,” Mobile Networks and Applications, vol. 26, no. 3, pp. 981–987, Oct. 2020, doi: 10.1007/s11036-020-01668-3.

[10] H. Gao and X. Li, “Vehicle Detection in High Resolution Image Based on Deep Learning”, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B3-2020, 49–54, https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-49-2020, 2020.

[11] R. Theagarajan, F. Pala and B. Bhanu, "EDeN: Ensemble of Deep Networks for Vehicle Classification," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017, pp. 906-913, doi: 10.1109/CVPRW.2017.125.

[12] Z. Li, S. Chen, H. Dai, D. Xu, C. -K. Chu and B. Xiao, "Abnormal Traffic Detection: Traffic Feature Extraction and DAE-GAN With Efficient Data Augmentation," in IEEE Transactions on Reliability, vol. 72, no. 2, pp. 498-510, June 2023, doi: 10.1109/TR.2022.3204349.

[13] Q.-C. Mao, H.-M. Sun, L.-Q. Zuo, and R.-S. Jia, “Finding every car: a traffic surveillance multi-scale vehicle object detection method,” Applied Intelligence, vol. 50, no. 10, pp. 3125–3136, May 2020, doi: 10.1007/s10489-020-01704-5.

[14] K. Khazukov, V. Shepelev, T. Karpeta, S. Shabiev, I. Slobodin, I. Charbadze and I. Alferova, “Real-time monitoring of traffic parameters,” Journal of Big Data, vol. 7:84, doi: 10.1186/s40537-020-00358-x.

[15] S. Yang, L. Gao and Y. Zhao, "A Detection Model of the Complex Dynamic Traffic Environment for Unmanned Vehicles," in IEEE Access, vol. 10, pp. 51873-51888, 2022, doi: 10.1109/ACCESS.2022.3174859.

[16] C. Chen, B. Liu, S. Wan, P. Qiao and Q. Pei, "An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1840-1852, March 2021, doi: 10.1109/TITS.2020.3025687.

[17] V. Singh, P. Kaewprapha and C. Boonmee, "Ad-hoc Aerial-view Vehicle Detection and Tracking for Real-Time Traffic Monitoring Using YOLOv7," 2023 International Electrical Engineering Congress (iEECON), Krabi, Thailand, 2023, pp. 327-331, doi: 10.1109/iEECON56657.2023.10127081.

[18] Y. B. Saridogan, “Traffic Detection Project, Version 1”, Kaggle, September 5, 2023. [Online]. Available: https://www.kaggle.com/datasets/yusufberksardoan/traffic-detection-project. [Accessed: November 11, 2023].