e-ISSN 2549-7472
Edumatic: Jurnal Pendidikan Informatika
Vol. 9 No. 1, April, 2025, **Hal. 41-50**
DOI: 10.29408/edumatic.v9i1.29244

Zebra Cross Violation Detection with YOLOv9: A Novel Approach for Traffic Regulation in Indonesia

Muhammad Najmi Kamil [1,*], Gamma Kosala [1]

[1] Department of Informatics, Telkom University, Indonesia
[*] Correspondence: najmikamil@student.telkomuniversity.ac.id

Received: 9 January 2025 | Revised: 12 January | Accepted: 19 January 2025 | Published: 9 April 2025

Abstract
Technological advancements in zebra cross-violation detection are necessary to address traffic rule violations in Indonesia, especially zebra cross violations. The You Only Look Once (YOLO) algorithm has been effective for detecting objects in various situations. The objective of this research is to focus on detecting zebra cross violations using YOLOv9, the improved accuracy and efficiency from earlier versions of YOLO. Consisting of two models to detect violations of the zebra crossing. The first model, a segmentation model called YOLO, is used for zebra cross localization, while the second model, a pretrained YOLO, detects the vehicles. The results of these two models are used for calculations in considering violations by drivers. Two datasets were used in this research. One of the datasets has 1100 images of zebra crosses, while the other comprises 100 surveillance videos from CCTV in Yogyakarta, Indonesia, for testing. The findings from this study indicate that the approach enables effective and efficient detection and classification of zebra crossing violations with an accuracy of 93%. This research demonstrates the approach's enhanced ability to handle real-world scenarios with diverse camera angles and varying traffic conditions. Additionally, it underscores the potential for practical applications in automated traffic monitoring and enforcement.

Keywords: object detection; object segmentation; traffic violation; yolo; zebra cross

INTRODUCTION

The statistics of motorized vehicles in Indonesia according to the Indonesian National Police Traffic Corps (*Korlantas Polri*) show that the motorized vehicles reached a staggering 160,652,675 in February of 2024, which has significantly increased from the previously recorded 148,261,817 in 2022. This represents a significant increase over the past two years. However, alongside this increase, the infrastructure necessary for the vehicles hasn't improved and the roads have not expanded. Due to this imbalance, the traffic conditions became unbearable which in turn resulted in higher traffic violations as well.

Among the traffic violations, zebra-cross violations are particularly concerning due to their direct impact on pedestrian safety (Nkurunziza et al., 2023). Pedestrian crossings are often disregarded by drivers, leading to accidents and increasing the risk to vulnerable road users. For instance, according to data from *Korlantas Polri*, 8.274 traffic accidents occurred in 2023 involving pedestrians crossing the road. While the government has issued traffic regulations as regulated in Article 287 paragraph 1 of Law Number 22 of 2009 concerning Traffic and Road Transportation (LLAJ Law) concerning the obligation of drivers to obey command or prohibition signs and road markings, enforcement still remains a challenge, emphasizing the importance of technological solutions to detect and mitigate violations.

Advancements in computer vision and machine learning have enabled the development of automated methods for traffic monitoring and rule enforcement. The YOLO (You Only Look Once) algorithm has proven effective in detecting objects quickly and efficiently

(Bochkovskiy et al., 2020). While previous studies have explored the use of YOLO for zebra-cross violation detection, these efforts were limited by dataset scope, real-world applicability, and the ability to handle complex scenarios, such as multiple-object detection and varying camera angles.

This research applies YOLO, a real-time object detection method that efficiently detects objects in a single pass, streamlining the computational process (Wang et al., 2021). YOLO interprets image data as a regression problem, using deep learning to generate bounding boxes, labels, and confidence scores for detected (Reis et al., 2023; Wang et al., 2024). It is widely used in applications such as traffic signal detection, pedestrian monitoring, and parking identification due to its speed and accuracy (Al-qanees et al., 2021; Hsu & Lin, 2021; Naftali et al., 2022).

YOLOv9 offers a superior balance of speed and accuracy compared to other object detection algorithms, making it ideal for real-time applications like zebra-cross violation detection. Unlike Faster R-CNN, which uses a slower two-stage approach, YOLOv9's single-stage design predicts bounding boxes and classes simultaneously, enabling faster inference (Sharma et al., 2024). While SSD provides real-time performance, it struggles with small object detection, an area where YOLOv9 excels due to its advanced GELAN backbone (Leng & Liu, 2020; Yang et al., 2024; Yaseen, 2024). Compared to RetinaNet, YOLOv9 maintains similar accuracy but achieves higher frame rates, making it more efficient for real-world scenarios. Additionally, YOLOv9's flexibility with model variants (e.g., YOLOv9-n, YOLOv9-s) allows users to optimize performance based on hardware capabilities, further enhancing its usability.

In this study, we focus on developing a zebra-cross violation detection method that can handle real-world scenarios more dynamically using YOLOv9. The YOLOv9's improved architecture offers significant improvements over earlier versions (Imran et al., 2024), including enhanced accuracy, faster detection speeds, and better handling of complex scenarios (Glučina et al., 2024). The YOLO algorithm is suitable for this research because YOLO performs object detection and classification in a single end-to-end drilled network, which allows for more efficient learning compared to other models that require multiple training stages (e.g., models like Mask R-CNN that require a region proposal stage (Sapkota et al., 2024). YOLO is also known for its real-time prediction speed (Dewi et al., 2022; Lavanya & Pande, 2024). The model processes the entire image in a single convolutional step, making it very fast compared to other models that use region proposal-based approaches or pixel-wise segmentation (Yang et al., 2020) .

Several studies have been conducted to address zebra-cross violation. A previous study, developed a zebra-cross violation detection system using the YOLOv4 algorithm (Chianyung, 2022). This research also enabled the real-time transmission of violation detection results integrated with a Telegram bot. However, there are shortcomings in this study, such as the primary dataset used, which has been collected only on the campus of Telkom University, and does not reflect real traffic conditions. In addition, the system in this study was designed only for the detection of motorcycle violations and failed to detect multiple objects which are critical for addressing real-world traffic complexities.

The study by Tonge et al. (2020) employed the Mask R-CNN segmentation model to detect zebra crossings and vehicles, with additional functionality to extract license plates of violators. While the study achieved an impressive mAP of 96% using a dataset of 150 images, its reliance on y-axis overlaps calculations for violation detection limited its adaptability to various camera angles. This approach worked effectively only with camera angles directly facing vehicles, making it unsuitable for diverse road intersection layouts.

Although the methods from previous studies have successfully detected zebra-cross violations, further improvements are necessary to represent real-world conditions better. These enhancements are needed to address aspects that were not adequately handled in prior research,

such as varying camera angles, multi-object detection for vehicles, and more dynamic overlap calculations.

The research works in three stages: zebra-cross segmentation to identify the location of the crosswalk, vehicle detection with bounding boxes, and violation determination using the segmentation mask and vehicle detection results. The primary objective of this research is to develop a novel approach for zebra crossing violation detection by employing two distinct models to address each subtask. This research is expected to enhance traffic law enforcement, contribute to more effective pedestrian protection, and play a key role in supporting road safety initiatives across urban areas.

METHOD

Our focus is on building zebra cross violation detection, shown in figure 1, processes input video from a CCTV camera, functioning only during the day when the traffic light is red. Initially, a YOLOv9 model detects and segments the zebra cross within the first 10 seconds of the video to address potential congestion. The segmentation mask serves as the reference location for detecting violations.

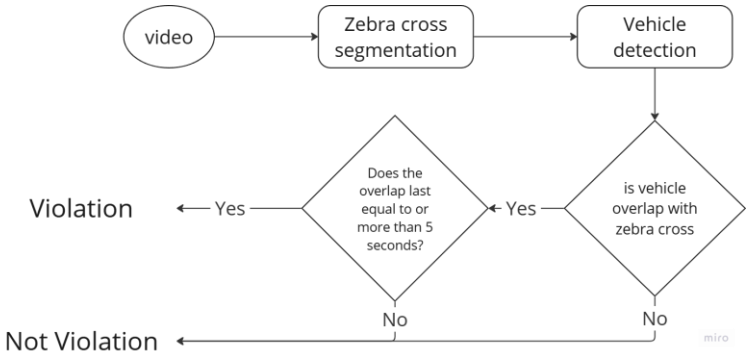

Figure 1. Zebra cross violation detection flow process

A pretrained YOLOv9 model detects vehicles (e.g., motorcycles, cars, buses, trucks) frame by frame, calculating the Intersection over Union (IoU) between each vehicle's bounding box and the zebra-cross mask. Vehicles with at least 10% overlap are flagged as potential violators. If overlap persists for more than 5 seconds, the vehicle is confirmed as violating the zebra crossing. Confirmed violations trigger frame capture for evidence.

Figure 2. First dataset examples

In this study, two types of datasets are utilized to address the subtasks of zebra-cross segmentation and violation classification. For zebra-cross detection and segmentation, a secondary dataset consisting of 1,100 zebra-cross images is sourced from the Roboflow

platform, as shown in figure 2. These images are masked and augmented through techniques such as noise addition, blurring, and rotation to improve robustness against environmental variations like uneven lighting or partial obstructions. The dataset is divided into 80% for training, 10% for validation, and 10% for testing, ensuring a balanced and effective model evaluation. For the violation classification task, the primary dataset consists of 100 red-light intersection videos recorded from 10 CCTV locations across the Special Region of Yogyakarta with 10 different camera angles, as illustrated in figure 3. The drawback of this dataset is that it only contains videos from sunny weather, so vehicle misdetection is possible. This dataset includes 50 labeled "Non-Violation" and 50 labeled "Violation." Videos are resized to 384x640 resolution for uniformity.

Figure 3. Second dataset examples

Given the two distinct subtasks in this study, we employed separate evaluation metrics for each task. We utilised mean Average Precision (mAP), specifically mAP50 and mAP>50, for the zebra-cross detection and segmentation task. These correspond to the average precision (AP) with an Intersection over Union (IoU) threshold of 50%, and mAP with IoU thresholds ranging from 50% to 95%. Additionally, we calculated the mean pixel accuracy, which represents the average accuracy per class. This metric is derived by dividing the number of correctly classified pixels for each class by the total number of pixels in that class, then averaging across all classes. For the main task (zebra-cross violation classification), standard classification metrics, including accuracy, precision, recall, and F1 score, were utilized to assess the model's ability to correctly identify violations and non-violations. Beyond these, average inference time were also considered to evaluate the real-time performance of the program. These metrics assess the research's suitability for deployment in real-world applications where timely detection is critical.

RESULTS AND DISCUSSION

Results

This experiment will involve several configuration scenarios for the zebra-cross segmentation subtask. The training will utilize the hyperparameter configurations specified in Table I, along with several fixed hyperparameters. These include an image size of 640, a batch size of 8, and 100 epochs. The combinations of parameters and hyperparameters are as follows in table 1.

Table 1. Experiment configuration table

Scenario	Weight	Optimizer	Learning Rate
1	YOLOv9c-seg	AdamW	0.001
2	YOLOv9c-seg	AdamW	0.01
3	YOLOv9c-seg	SGD	0.001
4	YOLOv9c-seg	SGD	0.01
5	YOLOv9e-seg	AdamW	0.001
6	YOLOv9e-seg	AdamW	0.01
7	YOLOv9e-seg	SGD	0.001
8	YOLOv9e-seg	SGD	0.01

Table 2. Training result on the segmentation model

Scenario	Using Data				Pixel Accuracy	Avg Inference Time (ms)
	Val		*Test*			
	mAP50	*mAP>50*	*mAP50*	*mAP>50*		
1	0.9836	0.8614	0.9833	0.8267	0.9639	33.3
2	0.1111	0.0293	0.0698	0.0202	0.7123	35
3	0.9829	0.8706	0.9841	0.8280	0.9529	30.6
4	0.9882	0.8770	0.9740	0.8114	0.9653	31
5	0.9895	0.8715	0.9752	0.8147	0.9694	55.6
6	0.3261	0.1369	0.3709	0.1480	0.8231	57
7	0.9907	0.8871	0.9831	0.8354	0.9425	55.6
8	0.9899	0.8722	0.9820	0.8295	0.9660	56.1

The results presented in Table 2 indicate that Scenario 1 and Scenario 5 produced the high performance, as both showed impressive mAP and Pixel Accuracy. Scenario 1, which uses YOLOv9c-seg, the AdamW optimizer, and a learning rate of 0.001, achieved excellent mAP on both the validation and test datasets, along with a high Pixel Accuracy of 0.9639. This suggests that the model from Scenario 1 is highly accurate in pixel segmentation. Meanwhile, Scenario 5, with the same configuration but utilizing YOLOv9e-seg, also yielded a high Pixel Accuracy of 0.9694.

Scenarios 1 and 5 demonstrated the best performance in terms of mAP, and Pixel Accuracy, making them the optimal choices for zebra-cross segmentation. Scenario 1, using YOLOv9c-seg with the AdamW optimizer and a learning rate of 0.001, achieved high accuracy (Pixel Accuracy: 0.9639) with an efficient inference time of 33.3 ms, making it suitable for resource-limited systems. Scenario 5, utilizing YOLOv9e-seg with the same configuration, delivered the highest Pixel Accuracy (0.9694) but at a slightly higher inference time of 55.6 ms, making it ideal for precision-critical applications. These configurations balance accuracy and real-time efficiency, ensuring reliable segmentation in diverse scenarios. These findings, consider the authors to choose the models from Scenario 1 and Scenario 5 to be utilized for object segmentation in the main program

As can be seen in Figure 4, scenarios with a learning rate of 0.01 and the AdamW optimizer (Scenarios 2 and 6) performed poorly, showing very low mAP values on both validation and test datasets, along with significantly reduced Pixel Accuracy. AdamW utilizes weight decay regularization to mitigate overfitting, typically with a default value of 0.0005, which is effective when combined with a small learning rate. However, a high learning rate causes weight decay to impede convergence, making it challenging for the model to learn essential parameters and patterns from the data. This results in a substantial drop in detection and segmentation performance.

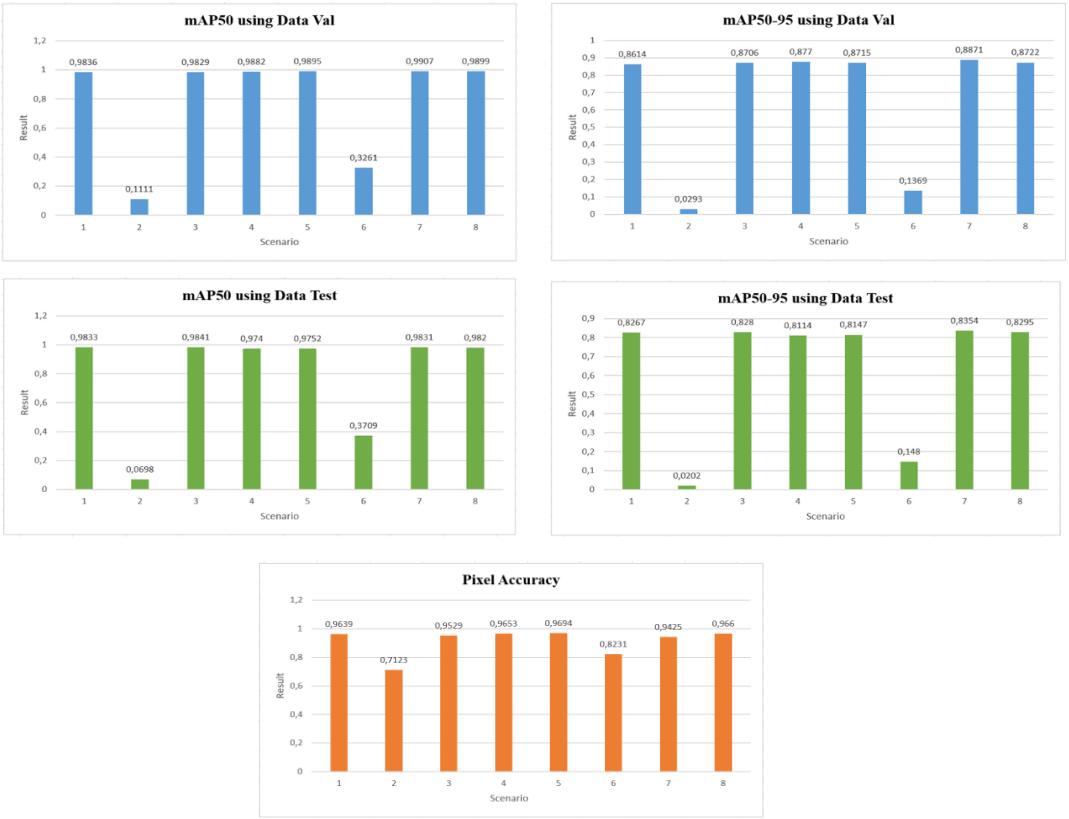

Figure 4. Visualization of segmentation model result

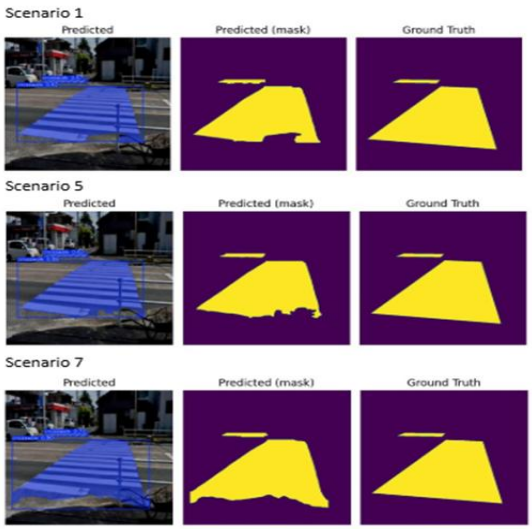

Figure 5. Mask comparison between scenario 1, scenario 5, and scenario 7

Although Scenario 7 demonstrates a very high mAP on both the validation and test datasets, with mAP50 and mAP50-95 outperforming other scenarios, the quality of the segmentation masks generated, as shown in Figure 5, reveals certain limitations. The masks in Scenario 7 appear less precise in delineating the zebra-cross boundaries compared to Scenario 1 and Scenario 5. This suggests that while Scenario 7 excels in detecting objects at a bounding box level, it struggles with fine-grained, pixel-level segmentation. The use of the SGD optimizer, which may lead to less stable convergence in tasks requiring detailed segmentation. Conversely, the AdamW optimizer employed in Scenario 1 and Scenario 5 offers better stability and generalization, particularly for pixel-level tasks, as evidenced by the sharper and

more accurate segmentation masks in these scenarios. This discrepancy is further highlighted in Scenario 3, which also uses SGD with a learning rate of 0.001. Although it achieves a mAP close to Scenario 1, its lower Pixel Accuracy of 0.9529 indicates a reduced ability to produce precise and accurate segmentation masks.

The main program testing will assess the main task's accuracy using the videos dataset in identifying whether an input video includes a violation. This evaluation will employ the optimal weights obtained from the first subtask experiments as the zebra-cross segmentation model. For vehicle object detection (second subtask), pretrained YOLO models with YOLOv9c and YOLOv9e weights will be utilized. The main program scenarios will integrate the top-performing segmentation model from prior training with the YOLOv9c and YOLOv9e object detection models.

Table 3. Main program result

Pretrained	*Segmentation Model*	*Precision*	*Recall*	*F1-score*	*Accuracy*
YOLOv9c	1	0.895	0.89	0.89	0.89
YOLOv9c	5	0.9	0.9	0.9	0.9
YOLOv9e	1	0.905	0.9	0.9	0.9
YOLOv9e	5	0.935	0.93	0.93	0.93

Table 3 presents the testing results based on four scenarios combining the trained segmentation models with the pretrained YOLO object detection models. The pretrained weights refer to the YOLO model weights used for vehicle detection. The segmentation models are derived from the best-performing scenarios identified earlier, specifically Scenario 1 (YOLOv9c-seg, LR 0.001, AdamW optimizer) and Scenario 5 (YOLOv9e-seg, LR 0.001, AdamW optimizer).

The integration of YOLOv9e with the segmentation model from Scenario 5 delivers optimal results for violation detection. YOLOv9e provides precise and reliable vehicle detection, while the Scenario 5 model ensures accurate zebra cross segmentation. This combination enables this new method to identify violations with exceptional accuracy and reliability. As shown in Figure 6, the crosswalk is segmented perfectly, and vehicle detection yields accurate results, enabling the overlap calculation to identify violations correctly.

Figure 6. Visualization of detected violation

However, future work could explore optimization techniques, such as model pruning or quantization, to reduce computational demands to improve the real-time applicability. Also the method is currently limited to functioning effectively only under sunny weather conditions. The accuracy of zebra cross segmentation and vehicle detection can be significantly impacted during night time and adverse weather, such as rain or fog, which can reduce visibility and complicate the detection process. This needs to be addressed in further research.

Discussion

The development of zebra-cross violation detection using YOLOv9 shows promise for enhancing pedestrian safety and traffic enforcement. While results of the most optimal model of segmentation task, YOLOv9e-seg, achieved mAP50 of 0.9752, mAP>50 of 0.8147, and pixel accuracy of 0.9694, give a good result to the main program with the result of F1-score, Recall, Precision, and Accuracy of 0.93, achieving this involves trade-offs. YOLOv9e offers higher segmentation accuracy, making it ideal for complex scenarios but has slower inference times compared to the lightweight and faster YOLOv9c, which suits with limited resources or higher frame rate needs.

The study's main limitation is its reliance on datasets collected under ideal conditions. The models perform well in sunny weather but may struggle in low-light, adverse weather, or varying camera angles and resolutions. Greater dataset diversity is needed to address these challenges and enhance performance across broader environmental and geographical contexts. Future research should expand the dataset to include diverse conditions, such as nighttime, varied weather, and different camera setups, to improve the method's robustness. Integrating advanced preprocessing techniques, like contrast adjustment or denoising, could enhance performance under challenging conditions. Additionally, exploring optimization methods to reduce computational demands and incorporating features like license plate recognition would make it more efficient and practical for real-world applications.

In comparison to prior research, this study highlights several improvements. The use of YOLOv9, with its advanced GELAN backbone, surpasses the performance of YOLOv4 and Mask R-CNN models in terms of both speed and accuracy. For example, Chianyung (2022) YOLOv4-based system achieved real-time detection but was limited to motorcycle violations in a single geographic location, whereas this new approach demonstrated broader applicability and higher precision across multiple vehicle types. Similarly, the study by Tonge et al. (2020) employed Mask R-CNN for zebra-cross violation detection but faced difficulties in handling diverse camera angles. Their approach relied heavily on y-axis overlap calculations, making it effective only for frontal camera views.

By integrating YOLOv9's segmentation techniques, this study partially overcomes these challenges by achieving robust performance across varying camera angles, thus improving its adaptability to different road layouts and surveillance setups. However, despite its strengths, the YOLOv9 still has limitations. While its GELAN backbone provides enhanced detection accuracy, it is computationally intensive, which can lead to higher inference times on resource-constrained systems. This issue is particularly relevant for real-time applications where latency must be minimized.

The contributions of this research lie in its improved real-time detection capabilities, adaptability to various road conditions, and robust performance across challenging scenarios, offering a clear advantage over earlier studies. These innovations establish a foundation for future work aimed at further addressing computational efficiency and environmental adaptability, thus advancing the field of automated traffic monitoring.

CONCLUSION

This study demonstrates the potential of YOLOv9 in improving zebra-cross violation detection, offering a more adaptable and efficient solution compared to previous methods. By addressing challenges such as diverse vehicle types and varying camera angles, the approach enhances real-time detection and provides practical applications for traffic law enforcement. While the study's findings show promise, future improvements are needed in dataset diversity and computational efficiency to ensure robustness across different conditions and optimize real-world applicability. These advancements could contribute to the development of more scalable and reliable systems for intelligent transportation.

REFERENCES

Al-qanees, M. A., Abbasi, A. A., Fan, H., Ibrahim, R. A., Alsamhi, S. H., & Hawbani, A. (2021). An improved YOLO-based road traffic monitoring system. *Computing*, *103(2)*, 211–230. https://doi.org/10.1007/s00607-020-00869-8

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. https://doi.org/10.48550/arXiv.2004.10934

Chianyung. (2022). Cross Zebra Violation Detection System on Motorcycle Vehicles Using The YOLOv4 Algorithm. *E-Proceeding of Engineering*, *10*(1), 815–822.

Dewi, C., Chen, R. C., Zhuang, Y. C., & Christanto, H. J. (2022). Yolov5 series algorithm for road marking sign identification. *Big Data and Cognitive Computing*, *6*(4), 1-16. https://doi.org/10.3390/bdcc6040149

Glučina, M., Žigulic, N., Frank, D., Lorencin, I., & Matika, D. (2024). Comparative Analysis of YOLOv9 and YOLOv10 Algorithms for Urban Safety Improvement. *IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, 185-190. IEEE. https://doi.org/10.1109/SISY62279.2024.10737612

Hsu, W. Y., & Lin, W. Y. (2021). Adaptive Fusion of Multi-Scale YOLO for Pedestrian Detection. *IEEE Access*, *9*, 110063–110073. https://doi.org/10.1109/ACCESS.2021.3102600

Imran, A., Hulikal, M. S., & Gardi, H. A. (2024). Real Time American Sign Language Detection Using Yolo-v9. *arXiv preprint arXiv:2407.17950*. https://doi.org/10.48550/arXiv.2407.17950

Lavanya, G., & Pande, S. D. (2024). Enhancing Real-time Object Detection with YOLO Algorithm. *EAI Endorsed Transactions on Internet of Things*, *10*, 1-9. https://doi.org/10.4108/eetiot.4541

Leng, J., & Liu, Y. (2019). An enhanced SSD with feature fusion and visual reasoning for object detection. *Neural Computing and Applications*, *31*(10), 6549-6558. https://doi.org/10.1007/s00521-018-3486-1

Naftali, M. G., Sulistyawan, J. S., & Julian, K. (2022). Comparison of object detection algorithms for street-level objects. *arXiv preprint arXiv:2208.11315*. https://doi.org/10.48550/arXiv.2208.11315

Nkurunziza, D., Tafahomi, R., & Faraja, I. A. (2023). Pedestrian Safety: Drivers' Stopping Behavior at Crosswalks. *Sustainability*, *15*(16), 1-17. https://doi.org/10.3390/su151612498

Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-time flying object detection with YOLOv8. *arXiv preprint arXiv:2305.09972*. https://doi.org/10.48550/arXiv.2305.09972

Sapkota, R., Ahmed, D., & Karkee, M. (2024). Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. *Artificial Intelligence in Agriculture*, *13*, 84-99. https://doi.org/10.48550/arXiv.2312.07935

Sharma, A., Kumar, V., & Longchamps, L. (2024). Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. *Smart Agricultural Technology*, *9*, 100648. https://doi.org/10.1016/j.atech.2024.100648

Tonge, A., Chandak, S., Khiste, R., Khan, U., & Bewoor, L. A. (2020). Traffic Rules Violation Detection using Deep Learning. T*he 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, 1250–1257. IEEE https://doi.org/10.1109/ICECA49313.2020.9297495

Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. (2024). Yolov9: Learning what you want to learn using programmable gradient information. *European conference on computer vision*, 1-21. Springer, Cham. https://doi.org/10.1007/978-3-031-72751-1_1

Wang, J., Zhang, T., Cheng, Y., & Al-Nabhan, N. (2021). Deep learning for object detection: A survey. In *Computer Systems Science and Engineering*, *38*(2), 165-182. Tech Science Press. https://doi.org/10.32604/CSSE.2021.017016

Yang, M., Chen, B., Lin, C., Yao, W., & Li, Y. (2024). SGI-YOLOv9: an effective method for crucial components detection in the power distribution network. *Frontiers in Physics*, *12*, 1517177. https://doi.org/10.3389/fphy.2024.1517177

Yang, X., Wang, Y., & Laganiere, R. (2020). A scale-aware YOLO model for pedestrian detection. *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, 15-26. Springer International Publishing.

Yaseen, M. (2024). What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. *arXiv preprint arXiv:2409.07813.* https://doi.org/10.48550/arXiv.2409.07813