Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id JTiP-UNP

Hyperparameter Tuning of YOLOv8n for Real-Time Material Truck Detection

I Gede Angga Saputra[1*]✉, I Nyoman Eddy Indrayana[1], Ida Bagus Adisimakrisna Peling[1]
[1]Department of Information Technology, Politeknik Negeri Bali, Badung, Indonesia
Corresponding Author: gedeangga424@gmail.com

Article Information

Article history:

No. 1138
Rec. April 16, 2026
Rev. May 10, 2026
Acc. May 11, 2026
Pub. May 13, 2026
Page. 1565 – 1579

Keywords:

- *YOLOv8n*
- *object detection*
- *truck detection*
- *hyperparameter tuning*
- *real-time monitoring*

ABSTRACT

The increasing number of material trucks on arterial roads has posed challenges for traffic surveillance and regulatory compliance. Traditional monitoring techniques that rely on manual observation are often ineffective and susceptible to irregularities, highlighting the need for automated real-time monitoring systems. This study proposes a lightweight object detection approach using YOLOv8n to improve real-time truck detection performance in traffic monitoring applications. A quantitative experimental methodology was employed by performing hyperparameter tuning through adjustments to the number of epochs, batch size, optimizer, and learning rate. The dataset was collected from real traffic environments using smartphone cameras and CCTV (TP-Link Tapo C320WS). A total of 36 experimental configurations were evaluated using Precision, Recall, F1-score, mAP@50, and mAP@50–95 metrics. Experimental results showed that the optimal configuration, consisting of 100 epochs, a batch size of 16, the Adam optimizer, and a learning rate of 0.001, achieved a mean Average Precision (mAP)@50 of 0.9302 and mAP@50–95 of 0.7226. Although the performance improvement over the baseline YOLOv8n model was relatively modest, repeated experiments demonstrated improved model stability and consistency after hyperparameter optimization. Real-time deployment on a local GPU achieved a stable processing speed of 14–23 Frames Per Second, with an average of 19 FPS, enabling real-time monitoring performance aligned with the camera input rate. The integrated system successfully combines object detection, tracking, and license plate recognition for practical traffic monitoring applications. However, smaller objects such as license plates remained more challenging to detect due to localization limitations under occlusion and low-light conditions.

How to Cite:

Saputra, I. G. A., & et al. (2026). Hyperparameter Tuning of YOLOv8n for Real-Time Material Truck Detection. Jurnal Teknologi Informasi Dan Pendidikan, 19(2), 1565-1579. https://doi.org/10.24036/jtip.v19i2.1138

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

1. INTRODUCTION

The increasing number of material trucks on arterial highways has resulted in numerous issues, such as traffic congestion, road deterioration, and heightened accident risks. In Bali, this situation is further complicated by the enforcement of regional restrictions, specifically the Circular Letter of the Governor of Bali Number 2 of 2026, which governs vehicle access and operational compliance in designated areas. Traditional monitoring techniques dependent on manual observation frequently exhibit inefficiencies and limitations in reliably identifying violations or suspected fraudulent activities, underscoring the necessity for an automated, real-time monitoring system.

Recent advancements in computer vision, namely in object detection using deep learning, have facilitated the creation of intelligent monitoring systems. You Only Look Once (YOLO) model family demonstrates strong performance in real-time detection, attributed to its speed and precision [1], [2], [3]. YOLOv8n features a lightweight architecture ideal for implementation in resource-limited settings [4], [5].

Numerous prior studies have utilized YOLO-based models for vehicle detection. Nonetheless, the majority concentrate on broad vehicle classification, neglecting real-time system integration and the optimization of training parameters for particular scenarios [6], [7], [8]. Moreover, insufficient focus has been directed towards the influence of hyperparameter tuning on enhancing detection accuracy in practical traffic monitoring systems [9].

To address these limitations, YOLOv8n was optimized through hyperparameter tuning, including adjustments to epochs, batch size, optimizer, and learning rate [10]. The dataset utilized in this work was gathered employing CCTV (Tapo C320WS) across diverse environmental conditions. The optimized model is subsequently integrated into a real-time system that integrates object detection, tracking with DeepSORT, and license plate identification utilizing EasyOCR [11].

This study primarily contributes by evaluating the effectiveness of hyperparameter tuning in enhancing YOLOv8n performance for particular object identification tasks and illustrating its implementation in a real-time traffic system. The findings are expected to provide insights into the creation of effective and precise monitoring systems for traffic regulation enforcement.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

2. RESEARCH METHOD

This study utilizes a quantitative experimental methodology to assess the performance of YOLOv8n under various hyperparameter setups. The research pipeline encompasses data collection, data preprocessing, hyperparameter tuning, system integration, and model evaluation.

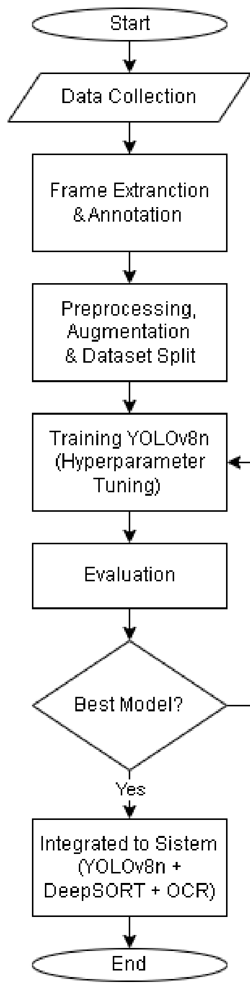


Figure 1. Research workflow of the proposed system

The dataset utilized in this study was collected from real traffic environments using a smartphone camera and a security camera (TP-Link Tapo C320WS) installed on major roads in Bali [12]. The data collection was conducted under various environmental conditions to improve model robustness. The dataset was primarily collected under daytime traffic conditions, while nighttime scenarios remained relatively limited in this study. The recorded videos were converted into image frames, yielding a total of 2,990 images.

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

All photos were carefully annotated utilizing Roboflow with three object classes: Targeted Truck (truk_target), Truck (truk), and License Number (plat_nomor) [13]. The Targeted Truck class denotes transport trucks utilized for carrying materials in Bali, namely those employed for the transportation of excavation materials, referred to locally as "galian C." These trucks generally possess unique visual attributes, including dump trucks, open cargo, and particular structural forms that distinguish them from standard cargo trucks [14]. The preprocessing procedures encompass scaling images to 640×640 pixels [15] and employing data augmentation techniques, including brightness adjustment of -25% and +25%, blur up to 1.5px, and noise up to 1.72% of pixels. The dataset was split into training (70%), validation (20%), and testing (10%) subsets [16], [17].

(a) (b) (c)

Figure 2. Sample dataset and annotation results (a) Targeted Truck + License Number, (b) Multi-object Trucks, (c) Targeted Truck + License Number

This study employs the YOLOv8 Nano (YOLOv8n) object detection model, chosen for its lightweight architecture and appropriateness for real-time applications [18]. Hyperparameter adjustment was performed to enhance model performance without altering the network architecture [19], [20]. Training was conducted in a cloud-based environment utilizing GPU acceleration (Google Colab), employing a baseline model with default settings as a reference point.

Table 1. Hyperparameter configurations used in this study.

No	Hyperparameter	Values
1	*epochs*	50, 100, 150
2	*batch_size*	16, 32, 64
3	*imgsz*	640
4	*optimizer*	*SGD, Adam*
5	*lr0*	0.01, 0.001

Four primary hyperparameters were assessed: the number of epochs (50, 100, and 150), batch sizes (16, 32, and 64), optimization algorithms (Adam and SGD), and learning rates (0.01 alongside 0.001). The input image size is fixed at 640×640 pixels, yielding a total of 36 experimental configurations[21].

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

To evaluate the consistency of the training process, repeated experiments were conducted under randomized training conditions using the same hyperparameter configurations for both the baseline and optimized YOLOv8n models.

The optimal YOLOv8n, integrated with DeepSORT and EasyOCR, was deployed in a real-time monitoring system. It was evaluated on a local machine (Intel Core i5, 16GB of RAM and GPU RTX 3050) processing 15 FPS RTSP streams from a TP-Link Tapo C320WS camera or archived videos. Additionally, a time logic and vSirtual line technique was implemented to detect Targeted Truck (truk_target) infractions across designated boundaries.

To measure the detection capabilities, metrics including Precision, Recall, F1-score, and Mean Average Precision (mAP) were calculated. The principal assessment metric employed in this study is mAP@50, whereas mAP@50–95 serves as a supplementary metric to evaluate localization accuracy across various IoU thresholds .

3. RESULTS AND DISCUSSION

This section presents the comprehensive findings of the hyperparameter tuning process for the YOLOv8n model and evaluates its performance in the context of material truck detection. The discussion focuses on how different configurations influence key detection metrics and the subsequent effectiveness of the model when integrated into a real-time monitoring system. By comparing the results, the analysis highlights the stability and precision gains achieved through systematic tuning. Furthermore, this section examines the practical implications of the model's performance on various object classes, particularly in handling the challenges posed by different object sizes and environmental conditions during live deployment

3.1. Experimental Results

A number of experiments were performed to assess the accuracy of the YOLOv8n model in detecting material trucks and license plates. A total of 36 hyperparameter configurations were evaluated, encompassing differences in epochs, batch size, optimizer, and learning rate. The performance of every configuration was quantified through Precision, Recall, F1-score, mAP@50, and mAP@50–95. A baseline model utilizing the default parameters of YOLOv8n was incorporated for comparative analysis.

Table 2 displays the ten most effective hyperparameter settings in conjunction with the baseline model. Due to space constraints, only the most effective configurations are presented, while the comprehensive findings of all 36 tests are excluded. All setups demonstrated notably high mAP@50 values (exceeding 0.91), signifying that YOLOv8n possesses robust baseline performance. The optimal model was achieved with Adam optimizer with a learning rate of 0.001, trained over 100 epochs with a batch size of 16,

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

resulting in a mAP@50 of 0.9302 and a mAP@50–95 of 0.7226. In comparison to the baseline model, the enhancement was modest, indicating that the default YOLOv8n setup is already close to optimal[22].

Table 2. Training Results

Epoch	Batch	Optimizer	lr	precision	recall	mAP50	mAP50-95
100	16	Adam	0.001	0.882007743	0.913275641	0.930209921	0.722611267
100	default	default	default	0.872930367	0.925992585	0.930032853	0.721433409
100	64	SGD	0.001	0.852430794	0.941785546	0.929819906	0.706633951
50	32	Adam	0.001	0.849649038	0.930332716	0.926760561	0.719732534
50	16	SGD	0.01	0.875230148	0.922385787	0.92528883	0.71685624
150	32	Adam	0.001	0.862433875	0.924718874	0.925043999	0.720805019
150	16	SGD	0.01	0.856389422	0.935481748	0.922854652	0.716364914
150	16	Adam	0.001	0.882308011	0.897843208	0.922536071	0.71826711
150	32	SGD	0.001	0.877329013	0.898975498	0.921963528	0.703479882
100	32	SGD	0.001	0.859074669	0.925333164	0.921559725	0.709689264

A class-wise analysis was performed to evaluate the detection capabilities across different target groups, in addition to the overall performance evaluation. The contrast between mAP@50 and mAP@50-95 scores for each class is depicted in Figure 3**.**

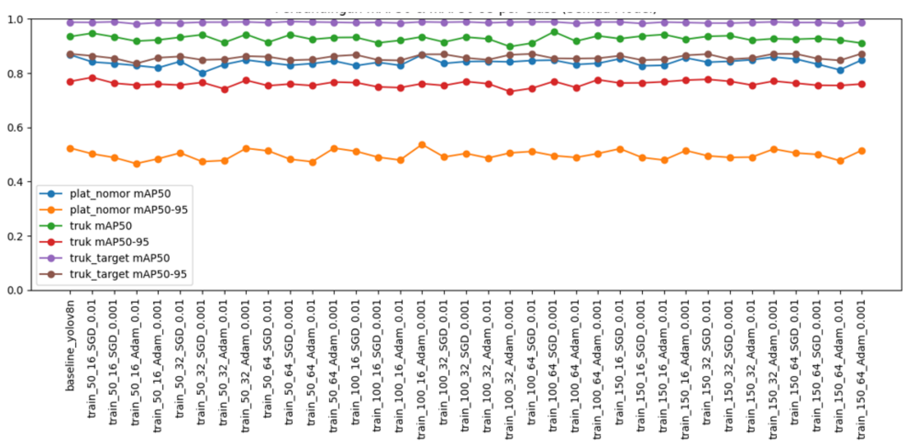


Figure 3. The contrasted results of mAP@50 and mAP@50-95 metrics for each object class across all experimental configurations

Figure 3 illustrates a significant disparity between the results of the mAP@50 and mAP@50-95 metrics, particularly concerning the License Number (plat_nomor) category. The mismatch mostly arises from the reduced object size and increased detection complexity, rendering precise localization more difficult. The model accurately detects license plates (high mAP@50) but fails to generate precise bounding boxes at elevated IoU thresholds, leading to diminished mAP@50–95 scores. Conversely, larger objects like Truck (truk) and Targeted Truck (truk_target) exhibit more reliable performance across both parameters, signifying enhanced localization precision and greater detection stability.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

3.2. Evaluation Metrics

To evaluate the performance of detection models, metrics such as Precision, Recall, and F1-score are commonly utilized. Precision quantifies the proportion of correct positive predictions, whereas Recall assesses the system's ability to successfully retrieve all relevant targets. Meanwhile, the F1-score is defined as the harmonic average between both Precision and Recall metrics.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (3)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{C} AP_i \quad (4)$$

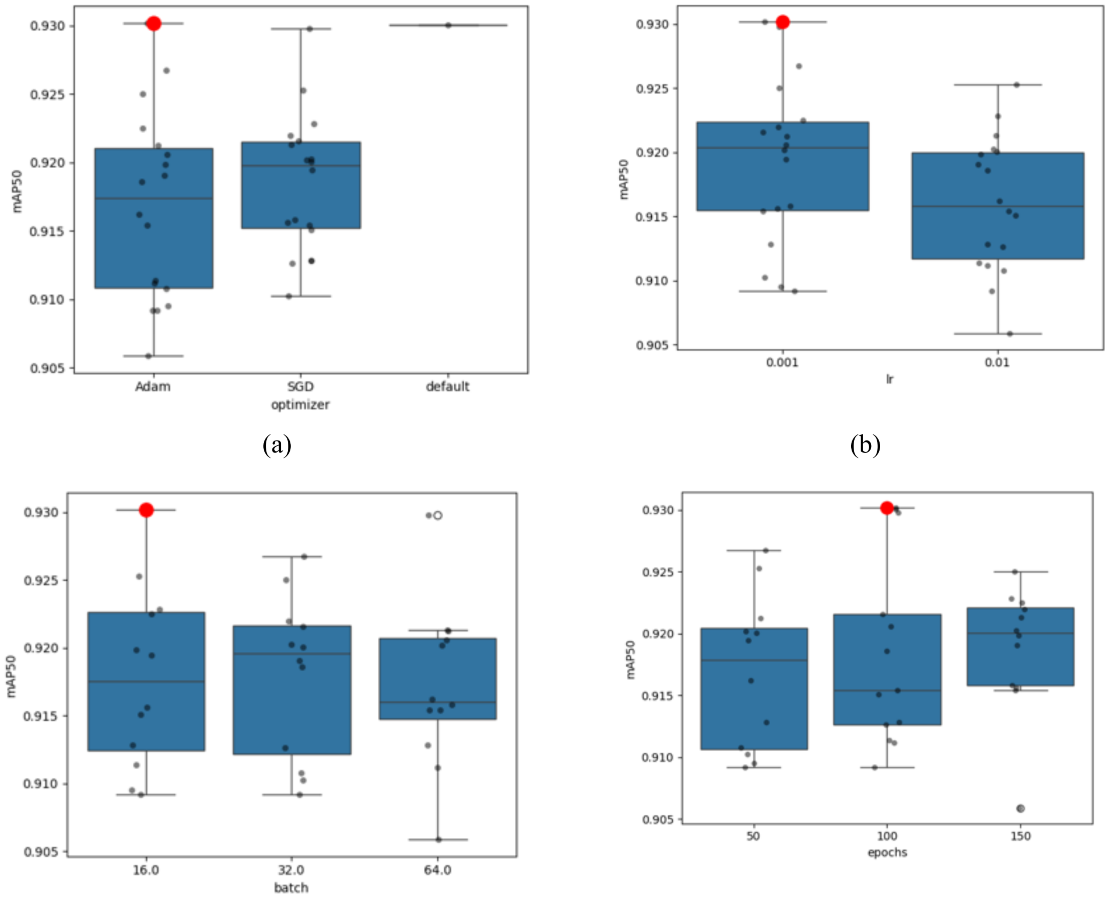
Within these mathematical formulas, the terms TP, FP, and FN stand for true positives, false positives, and false negatives, respectively. To obtain the Average Precision (AP), one must calculate the integral area under the precision-recall graph for every individual category. Furthermore, the mean Average Precision (mAP) is formulated by taking the mean of the AP values across all existing categories. In object detection tasks, mAP@50 is frequently employed as the primary evaluation parameter, whereas mAP@50–95 offers a more rigorous assessment across several IoU thresholds [19].

3.3. Hyperparameter Analysis

Subsequent research was conducted to evaluate the influence of each hyperparameter on model performance[20]. The results are presented in Figure 4. Figure 4(a) illustrates that the Adam optimizer exhibited enhanced stability with a reduced learning rate of 0.001, whereas SGD demonstrated somewhat elevated median values in certain configurations. Nonetheless, the disparities were not substantial.

Figure 4(b) demonstrates that a learning rate of 0.001 consistently yielded better and more stable performance relative to 0.01, signifying smoother convergence throughout training. Figure 4(c) indicates that smaller batch sizes (16) generally attain better peak performance, whereas larger batch sizes yield more consistent results. Figure 4(d) demonstrates that extending the number of epochs beyond 100 did not markedly enhance performance, showing that the model had already achieved optimal convergence.

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138



(a) (b)

(c) (d)

Figure 4. Hyperparameter analysis of YOLOv8n performance based on mAP@50: (a) optimizer comparison, (b) learning rate comparison, (c) batch size comparison, and (d) epoch variation.

3.4. Comparison

The accuracy of hyperparameter tuning was evaluated by comparing the optimal model to the baseline YOLOv8n model. The findings of the comparison are presented in Table 3. The baseline model exhibited competitive performance, suggesting that YOLOv8n yields robust results even in the absence of considerable optimization [21].

Table 3. Comparison of Optimized YOLOv8n and Baseline YOLOv8n

Model	Precision	Recall	F1-Score	mAP50	mAP50-95	Size
Optimized YOLOv8n	0.8820	0.9133	0.8974	0.9302	0.7226	5.9 MB
Baseline YOLOv8n	0.8729	0.9260	0.8986	0.9300	0.7214	6 MB

As observed in Table 3, the optimized model exhibits a trade-off between Precision and Recall when compared to the baseline. Specifically, the Optimized YOLOv8n yields a

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

higher Precision (0.8820 compared to 0.8729) alongside a slightly lower Recall (0.9133 compared to 0.9260). Functionally, this indicates that the hyperparameter tuning encourages the model to be more selective in its positive predictions, which helps in reducing False Positives (e.g., minimizing the misclassification of standard pickup trucks as targeted material trucks). While the slight drop in Recall implies a minor increase in missed detections (False Negatives), the overall detection performance remained relatively balanced.

Table 4. Statistical consistency analysis of repeated experiments

Model	Mean mAP50	Std. Dev	95% CI
Optimized YOLOv8n	0.9230	0.0050	0.9106 – 0.9354
Baseline YOLOv8n	0.9167	0.0046	0.9052 – 0.9281

To further evaluate the consistency of the proposed hyperparameter tuning strategy, repeated training experiments were conducted under randomized training conditions using the same configurations for both the baseline and optimized YOLOv8n models. The statistical results are summarized in Table 4. The optimized configuration achieved a slightly higher average mAP@50 score (0.9230) compared to the baseline model (0.9167). However, minor performance variations were observed across repeated runs, reflecting the stochastic characteristics of deep learning training processes.

The overlapping confidence interval ranges indicate that the observed improvement should be interpreted as a modest enhancement rather than a substantial performance increase. Nevertheless, the optimized configuration demonstrated slightly improved average detection performance and more consistent localization behavior across repeated experiments.

Table 5. Comparison of YOLOv8 Variants

Model	Precision	Recall	F1-Score	mAP50	mAP50-95	GFLOPs	Parameters	Size
Optimized YOLOv8n	0.8820	0.9133	0.8974	0.9302	0.7226	8.1	3.0 M	5.9 MB
YOLOv8s	0.8750	0.9120	0.8927	0.9250	0.7080	28.4	11.1 M	21.5 MB

To provide additional context regarding model selection, the optimized YOLOv8n model was compared with the larger YOLOv8s architecture using the same hyperparameter configuration. The comparison results are presented in Table 5. Although YOLOv8s achieved competitive detection performance, the optimized YOLOv8n model produced slightly higher Precision, F1-score, mAP@50, and mAP@50–95 values.

In contrast, YOLOv8s required substantially higher computational complexity, as reflected by its larger number of parameters, GFLOPs, and model size. Despite the increased model capacity, YOLOv8s did not demonstrate substantial performance improvements compared to the optimized YOLOv8n model. These findings indicate that the optimized

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

YOLOv8n configuration provides a more efficient balance between detection accuracy and computational cost, making it more suitable for real-time traffic monitoring applications.

3.5. Model Evaluation

To further assess the accuracy of the top-performing model, supplementary analysis was performed utilizing precision-recall curves and confusion matrix evaluation.

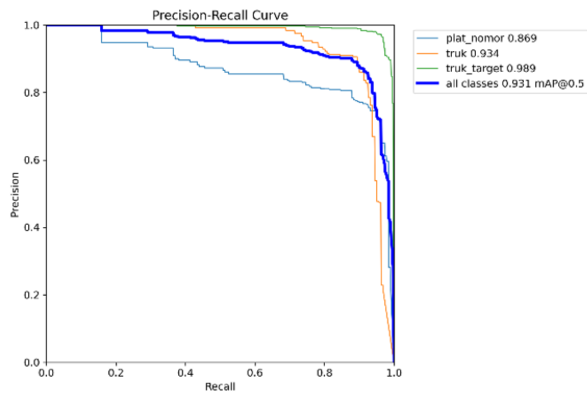


Figure 5. The precision-recall graph of the optimal model for each object class

Figure 5 illustrates that the model attains elevated precision and recall for larger objects, such as trucks and truck targets, signifying robust detection accuracy. Nonetheless, the performance for license plates is comparatively inferior, aligning with prior findings attributed to the reduced object size and increased detection challenges.

Figure 6. Confusion matrix of the optimized YOLOv8n model

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

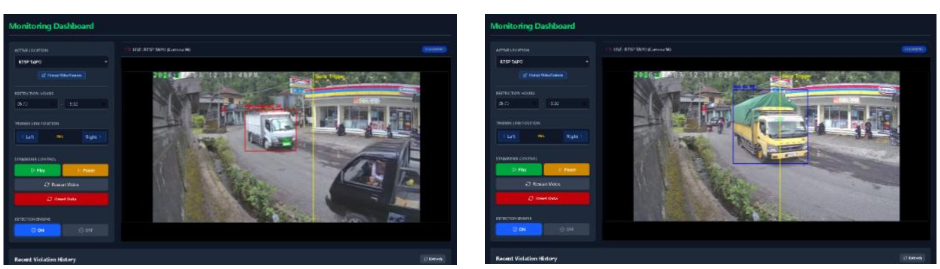
Figure 6 illustrates that the model accurately classifies the majority of cases, particularly for the truck_target class, which has the highest detection accuracy. Misclassifications have been noted within the license plate category, hence reinforcing the difficulties associated with identifying smaller items. The model exhibits robust classification accuracy with negligible confusion among primary object categories.

3.6. System Implementation

The most effective model was integrated into a real-time monitoring system utilizing video streams from CCTV (TP-Link Tapo C320WS) and recorded footage. The system incorporates YOLOv8n for object detection, DeepSORT for object tracking, and EasyOCR for license plate recognition.

In the integrated system, YOLOv8n is utilized to detect trucks and license plates from each video frame. The detected vehicles are subsequently processed using DeepSORT to maintain object identity consistency across consecutive frames, enabling stable tracking during vehicle movement. When a target truck crosses the predefined virtual line during restricted operational hours, the detected license plate region is cropped and forwarded to EasyOCR for text recognition. This integration enables the system to perform real-time object detection, tracking, and license plate identification simultaneously. However, OCR performance may decrease under challenging conditions such as motion blur, partial occlusion, low-light environments, or small license plate objects.

To optimize the tracking performance within the integrated system, DeepSORT was configured with a max_age of 30 frames to handle brief vehicle occlusions and an n_init of 3 frames to confirm new tracks, minimizing false positive trajectories. Regarding the EasyOCR component, its isolated performance was evaluated through functional testing during the real-time observation trials. As reflected in the minor misdetection rates (Table 6), the OCR successfully extracted license plate strings when YOLOv8n provided tightly cropped bounding boxes under clear daylight. The few observed system failures were primarily driven by OCR limitations, rather than detection misses, where text recognition degraded significantly when input regions suffered from motion blur or low-light conditions.



(a) (b)

Figure 7. Real-time detection and plate recognition: (a) detection of truck target and license plate and (b) detection of truck object

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

Figure 7 illustrates the real-time detection dashboard, wherein the system analyzes live video streams to identify moving cars. A temporal logic and virtual trigger line are employed to detect when objects traverse a certain boundary, facilitating automated surveillance of possible infractions.

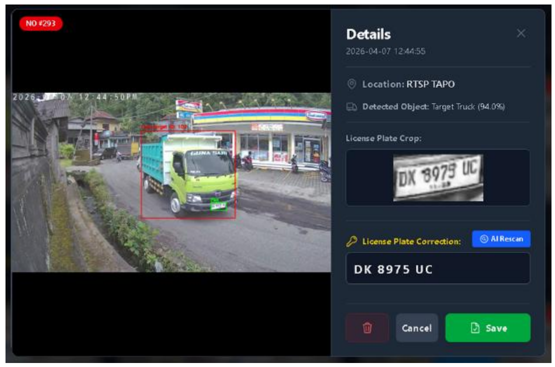


Figure 8. Detailed target truck detection result

Figure 8 displays the comprehensive detection outcomes, encompassing license plate identification with EasyOCR. The system extracts license plate numbers from identified vehicles and offers a manual verification option to enhance recognition precision and dependability.

To ensure the system's reliability, three separate live observation tests were conducted, each lasting 10 minutes. The averaged results of these real-time implementation trials are summarized in Table 6.

Table 6. Real-Time System Performance over Three Observation Trials

Parameter	Test 1 (10 min)	Test 2 (10 min)	Test 3 (10 min)
Hardware Specs	Intel i5, 16GB RAM, RTX 3050	Intel i5, 16GB RAM, RTX 3050	Intel i5, 16GB RAM, RTX 3050
Camera	15 FPS (Input)	15 FPS (Input)	15 FPS (Input)
Total Observed	34	29	53
Avg. FPS	14 – 23 (Avg: 19)	14 – 23 (Avg: 19)	14 – 23 (Avg: 19)
Success Objects Detected	32	29	52
Avg. Confidence Score	92.3%	89.96%	93%
Misdetections / Missed	2	0	1

Table 6 shows that the system achieved a processing speed of 14–23 FPS, exceeding the camera's 15 FPS input due to the high computational performance of the local GPU. The average confidence scores remained high (above 89%) because the vehicles were clearly visible under stable lighting conditions. However, minor misdetections in Test 1 and Test 3 were observed, primarily caused by objects partially occluding the target trucks or motion blur effects during rapid vehicle movement. Despite these challenges, the system consistently fulfills the requirements for real-time traffic monitoring.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1138

(a) (b)

Figure 9. Failure cases under challenging environmental conditions: (a) partial occlusion causing missed vehicle detection, and (b) low-light nighttime conditions reducing object visibility and OCR performance.

Several failure cases were observed during real-time deployment, particularly under challenging environmental conditions. As illustrated in Figure 9, detection performance may decrease due to partial occlusion and low-light nighttime conditions. Partial occlusion can cause missed vehicle detection when the target object is blocked by another vehicle, while nighttime environments reduce object visibility and OCR readability due to poor illumination and light glare.

4. CONCLUSION

This study evaluated the effectiveness of hyperparameter tuning on the YOLOv8n model for real-time material truck detection. Experimental results demonstrated that adjustments to hyperparameters, including epochs, batch size, optimizer, and learning rate, influence detection performance. The optimal configuration was achieved using 100 epochs, a batch size of 16, the Adam optimizer, and a learning rate of 0.001, resulting in a mAP@50 score of 0.9302 and a mAP@50–95 score of 0.7226. Although the optimized model only showed modest improvements over the baseline YOLOv8n model, hyperparameter tuning contributed to improved model stability and consistency across repeated experiments.

Further analysis indicated that object size significantly affected detection performance. Larger objects such as trucks and targeted material transport vehicles achieved more stable and accurate detection results, while smaller objects such as license plates remained more challenging, particularly in mAP@50–95 due to localization limitations. The optimized model was successfully deployed in a real-time monitoring system integrating object detection, tracking, and license plate recognition. Validation through multiple live observation trials demonstrated stable system performance with an average processing speed of 19 FPS, indicating suitability for real-world traffic monitoring applications.

This study still has several limitations. The dataset was primarily collected under daytime traffic conditions, causing nighttime and extremely low-light scenarios to remain relatively underrepresented. In addition, the detection of small objects such as license plates remains challenging under conditions involving motion blur, partial occlusion, and poor illumination, which may reduce localization accuracy and OCR readability.

Future research may focus on improving small-object detection performance and enhancing OCR robustness under challenging environmental conditions. Expanding the dataset with more diverse traffic scenarios, particularly nighttime conditions, may further improve model generalization. Additionally, future system development may explore deployment on edge-computing devices and integration with broader intelligent transportation systems for automated traffic monitoring and enforcement.

REFERENCES

[1] M. Kotthapalli, D. Ravipati, and R. Bhatia, "YOLOv1 to YOLOv11: A Comprehensive Survey of Real-Time Object Detection Innovations and Challenges," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2508.02067

[2] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475. [Online]. Available: https://arxiv.org/abs/2207.02696

[3] A. A. Murat and M. S. Kiran, "A comprehensive review on YOLO versions for object detection," Oct. 01, 2025, *Elsevier B.V.* doi: 10.1016/j.jestch.2025.102161.

[4] M. Bakirci, "Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems," *Traitement du Signal*, vol. 41, no. 04, pp. 1727–1740, Aug. 2024, doi: 10.18280/ts.410407.

[5] Z. Wang, K. Zhang, F. Wu, and H. Lv, "YOLO-PEL: The Efficient and Lightweight Vehicle Detection Method Based on YOLO Algorithm," *Sensors*, vol. 25, no. 7, Apr. 2025, doi: 10.3390/s25071959.

[6] Y. Li, Y. Huang, and Q. Tao, "Improving real-time object detection in Internet-of-Things smart city traffic with YOLOv8-DSAF method," *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-68115-1.

[7] J. Shen, Z. Zhang, J. Luo, and X. Zhang, "YOLOv5-TS: Detecting traffic signs in real-time," *Front. Phys.*, vol. 11, 2023, doi: 10.3389/fphy.2023.1297828.

[8] M. Bakirci, "Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications," *Digit. Signal Process.*, vol. 152, p. 104594, 2024, doi: https://doi.org/10.1016/j.dsp.2024.104594.

[9] J. W. Baek, J. Il Kim, M. H. Mun, and C. B. Kim, "Hyperparameter optimization to enhance the performance of deep learning models for the early detection of invasive turtles in Korea," *Sci. Rep.*, vol. 16, no. 1, p. 37636, Dec. 2026, doi: 10.1038/s41598-026-37636-2.

[10] H. Saputra, K. Muchtar, N. Chitraningrum, A. Andria, and A. Febriana, "Performance evaluation of hyper-parameter tuning automation in YOLOV8 and YOLO-NAS for corn leaf disease detection," *Sinergi (Indonesia)*, vol. 29, no. 1, pp. 197–206, 2025, doi: 10.22441/sinergi.2025.1.018.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

[11] F. Hidayat, N. Billy, N. Russel Permana, and M. Evans Hariady, "Penerapan You Only Look Once dan DeepSORT untuk Deteksi Plat Nomor Kendaraan," *Jurnal Telematika*, vol. 19, no. 2, pp. 53–59, 2025, doi: 10.61769/telematika.v20i2.676.
[12] Indra Bayu Pangestu, M. Maimunah, and Mukhtar Hanafi, "Traffic Congestion Detection Using YOLOv8 Algorithm With CCTV Data," *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, vol. 12, no. 2, pp. 435–444, Sep. 2024, doi: 10.33558/piksel.v12i2.9953.
[13] Saepudin, N. Sujana, M. M. Mutoffar, and A. A. Haryanto, "Analisis Kinerja YOLOv8 Optimalisasi Roboflow Untuk Deteksi Ekspresi Wajah Emosional Dengan Machine Learning," *Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika*, vol. 06, no. 2, pp. 115–124, 2024, doi: 10.53580/naratif.v6i2.292.
[14] P. Y. Putra, A. S. Arifianto, Z. E. Fitri, and T. D. Puspitasari, "Deteksi Kendaraan Truk pada Video Menggunakan Metode Tiny-YOLO v4," *JIP (Jurnal Informatika Polinema)*, vol. 9, no. 2, pp. 215–222, 2023, doi: 10.33795/jip.v9i2.1243.
[15] R. Apriliyanti, D. Kurniadi, D. Novaliendry, S. Rahmadika, and M. Farhan, "Real-Time Color Classification of Objects with an Improved MobileNetV2 CNN Model," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 18, no. 2, pp. 899–914, Aug. 2025, doi: 10.24036/jtip.v18i2.969.
[16] R. S. Saputri, A. Apriliani, R. S. Amar, and L. Y. Astri, "Deteksi Dini Microsleep Pada Pengemudi Kendaraan Roda Empat Melalui Citra Mata Menggunakan Algoritma YOLOv8," *Jurnal PROCESSOR*, vol. 20, no. 2, Oct. 2025, doi: 10.33998/processor.2025.20.2.2529.
[17] D. Novaliendry, F. Rizal, M. Anwar, and D. Irfan, "Aplikasi Web untuk Klasifikasi dan Deteksi Penyakit Daun Tomat Menggunakan Model CNN dan YOLO," *Jurnal Teknologi Informasi dan Pendidikan*, vol. 19, no. 1, pp. 1179–1192, Feb. 2026, doi: 10.24036/jtip.v19i1.1073.
[18] F. Wijaya and D. Hermanto, "Implementasi Metode YOLOv8 Mendeteksi Komputer Aktif dengan Subjek Layar Monitor," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 10, no. 3, pp. 319–330, 2025, doi: 10.14421/jiska.2025.10.3.319-330.
[19] G. D. Deepak and S. K. Bhat, "Optimizing YOLOv4 Hyperparameters for Enhanced Vehicle Detection in Intelligent Transportation Systems," *International Journal of Intelligent Transportation Systems Research*, vol. 23, no. 3, pp. 1372–1384, Dec. 2025, doi: 10.1007/s13177-025-00519-3.
[20] R. Fadilatul Fajriyah and Y. Sulistyo nugroho, "Analisis Tren Penelitian Hyperparameter Tuning dalam Software Engineering melalui Systematic Literature Review dan Bibliometric Analysis," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 8, pp. 2278–2294, Aug. 2025, doi: 10.52436/1.jpti.817.
[21] B. Irawan, N. Andono, and R. Suko Basuki, "Optimization of Yolov5 Hyperparameter Using Adam Optimizer in Vehicle Object Detection," 2024. doi: 10.62411/jais.v9i1.9244.
[22] H. Yu, Q. Luo, W. Peng, L. Zheng, J. Ju, and H. Zhuo, "PKD-YOLOv8: A Collaborative Pruning and Knowledge Distillation Framework for Lightweight Rapeseed Pest Detection," *Sensors*, vol. 25, no. 16, Aug. 2025, doi: 10.3390/s25165004.