JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. XI No 3, Juni 2025, hlm. 533 – 540 ISSN 2550-0201 (Online)
DOI: http://dx.doi.org/10.33330/jurteksi.v11i3.3871
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

TRAFFIC FLOW DETECTION USING YOLOV4 AND DEEPSORT ON NVIDIA JETSON NANO

Reny Medikawati Taufiq[1*], Syahril[1], Faris Abi Rafdi[1], Rahmad Firdaus[1], Sunanto[1], Putri Fadhilla Muarif[1]
[1]Informatic Engineering, Universitas Muhammadiyah Riau
email: [*]renymedikawati@umri.ac.id

Abstract: This study aims to develop a Deep Learning-based Traffic Flow Detector to automatically and accurately observe traffic flow. Conventional traffic observation is often conducted manually or via CCTV, but it is prone to human error and difficult to use for real-time trend analysis. In this study, the YOLOv4 method is used to detect four types of vehicles (cars, motorcycles, buses, trucks). To continuously track vehicle movement and address occlusion issues, the Deep SORT algorithm is implemented. The YOLOv4 model used is a pre-trained model and was tested on seven CCTV video recordings obtained from the official website of the Pekanbaru City Transportation Department. The system was implemented on a limited device, the Nvidia Jetson Nano, as a simulation of direct CCTV integration. Test results showed a highest precision of 98%, but the maximum accuracy achieved was only 26%. This low accuracy is influenced by several factors, including video resolution, detection model quality, and lighting conditions. Nevertheless, the system demonstrates potential to support future traffic management and engineering decisions but still requires further optimization, including improving video resolution and quality, retraining the model with a more representative local dataset, using lighter and more accurate detection models, and optimizing the tracking algorithm.

Keywords: deep learning; deepsort; NVIDIA Jetson NANO; traffic flow; YOLOv4

Abstrak: Penelitian ini bertujuan mengembangkan *Traffic Flow Detector* berbasis *Deep Learning* untuk mengobservasi arus lalu lintas secara otomatis dan akurat. Observasi lalu lintas konvensional sering dilakukan secara manual atau melalui CCTV, namun rentan terhadap *human error* dan sulit digunakan untuk menganalisis tren secara *real-time*. Pada penelitian ini digunakan metode YOLOv4 untuk mendeteksi empat jenis kendaraan (mobil, motor, bus, truk). Untuk melacak pergerakan kendaraan secara berkelanjutan dan mengatasi masalah *occlusion*, digunakan algoritma Deep SORT. Model YOLOv4 yang digunakan merupakan *pre-trained model* dan diujikan pada tujuh rekaman video CCTV yang diambil dari situs resmi Dinas Perhubungan Kota Pekanbaru. Sistem ini diimplementasikan pada perangkat terbatas Nvidia Jetson Nano sebagai simulasi penerapan langsung pada CCTV. Hasil pengujian menunjukkan presisi tertinggi mencapai 98%, namun akurasi tertingginya hanya sebesar 26%. Rendahnya akurasi dipengaruhi oleh beberapa faktor seperti resolusi video, kualitas model deteksi, serta kondisi pencahayaan. Meski demikian, sistem ini menunjukkan potensi untuk membantu pengambilan keputusan dalam manajemen dan rekayasa lalu lintas di masa depan, namun masih membutuhkan optimasi lebih lanjut, seperti peningkatan kualitas video input, pelatihan ulang model dengan dataset lokal, penggunaan model deteksi yang lebih ringan dan akurat serta pengoptimalan algoritma pelacakan.

Kata kunci: deep learning deepsort; Nvidia Jetson Nano; traffic flow; YOLOv4

INTRODUCTION

According to data from Badan Pusat Statistik (BPS) of Riau Province, the number of motorized vehicles of all types increased from 2019 to 2023. On average, 3,864,345 motorized vehicles (consist of passenger cars, buses, trucks, and motorcycles) were added each year. If this significant increase in vehicles is not balanced with sufficient road capacity as well as effective traffic flow management and regulation of vehicle volumes on road sections, congestion in urban areas will worsen.

Traffic management and planning in Pekanbaru City is the responsibility of the Dinas Perhubungan (Dishub) according to PP No. 32/2011, which comprises planning, regulation, engine ering, empowerment, and supervision. In carrying out its duties, Dishub identifies traffic problems, analyzes the flow and capacity of roads, and formulates traffic management policies. Observation of traffic flow is still conducted manually through direct observation or CCTV at several points in the city, therefore it remains incapable of performing automated and rapid analysis of traffic patterns and trends. As a result, traffic engineering such as closing or diverting he flow cannot be done responsively to overcome congestion. Despite, traffic distribution information is essential for forecasting future needs, evaluating road solutions, and designing infrastructure. Detecting and counting the number of vehicles on the road (such as cars, buses, trucks, motorcycles, ambulances) is also crucial for statistical data collection, congestion management, safety improvement, navigation, and optimal resource allocation [1] [2] [3].

Based on the problems mentioned, this research proposes the development of Intelligent Transportation Systems (ITS) based on deep learning that will be implemented on edge-computing devices using Nvidia Jetson Nano. This research will investigate how the Nvidia Jetson Nano is capable of observing traffic conditions. The use of edge-computing devices is intended so that traffic video captured by CCTV can be directly processed without sending it to a distant data center or cloud server, with the aim of reducing latency, increasing efficiency, and enabling real-time data processing. The use of deeplearning in intelligent systems to monitor vehicle activity on highways and urban roads has been implemented in several studies [4] [5] [6]. Some studies [7] [8] used Nvidia Jetson Nano as an ITS device.

YOLOv4 is used to detect four types of vehicles (cars, motorcycles, buses, and trucks). Vehicle movement is tracked using the Deep SORT algorithm, which is designed to effectively manage occlusion scenarios in which a vehicle is partially or fully obscured by other objects. Unlike object detection, which focuses solely on identifying and localizing objects within a single frame, object tracking entails the continuous association of those objects across multiple frames while preserving their unique identities over time [9].

The implementation of YOLO for vehicle tracking has been proposed in various studies and resulted in high accuracy, such as [10], [11], [12], [13] using YOLOv3 for vehicle tracking. YOLOv4 was proposed by [14] [15] [16] [17] [18] with various techniques and datasets.

[12] detected and counted three classes of vehicles, namely cars, buses, and trucks with YOLOv3, data sourced from unobstructed road CCTV cameras. The combination of the YOLO algorithm with Deep SORT results in better accuracy for detecting and tracking vehicles in a variety of different data sources, [13] uses the YOLOv3

DOI: http://dx.doi.org/10.33330/jurteksi.v11i3.3871
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

algorithm for vehicle extraction and detection and Deep SORT for tracking, [19] uses YOLOv4 and Deep SORT.

METHOD

The ITS simulation was carried out through the following sequential stages: (1) Preparation of input video data; (2) Configuration of the Traffic Flow Detector script to execute the YOLOv4–DeepSORT detection and tracking model; (3) Deployment of the Traffic Flow Detector model on the Nvidia Jetson Nano platform; (4) Performance testing of the model on the Nvidia Jetson Nano device; and (5) Extraction and analysis of vehicle count data.

The simulation process begins with the uploading of CCTV-recorded video to the NVIDIA Jetson Nano device. The video is subsequently processed using the YOLOv4 (You Only Look Once) algorithm for object detection and the Deep SORT (Simple Online and Realtime Tracking) algorithm for object tracking. This integrated detection and tracking process generates quantitative data related to traffic density levels and vehicle counts at the Labersa intersection.

The programming used is Python-based, using OpenCV and TensortFlow libraries. The specifications of the NVIDIA Jetson Nano used are 4 GB RAM, Quad-core ARM@ A57 @ 1.43 GHz CPU, 128-core NVIDIA Maxwell GPU.

Preprocessing Input Video

For the traffic flow detection simulation, this study utilized seven CCTV video recordings with a resolution of 1280 × 720 pixels, captured on different days. Each video, in .mp4 format, has a duration of approximately 60 minutes and was recorded at various time intervals, specifically at: 20:30–21:30, 06:30–07:53, 09:40–10:40, 14:42–17:07, 08:10–09:44, 16:46–17:54, and 11:11–12:21. Some of the recordings contained corrupted segments, which potentially affected system performance. Therefore, a pre-processing stage was conducted to trim or remove the defective portions of the videos to ensure more reliable input data for the detection process.

Pre-Trained Model

The pre-trained model utilized in this study is YOLOv4-tiny, which is based on the Darknet5 architecture and employs CSPDarknet53-tiny (CD53s-tiny) as the backbone and FPN-tiny as the neck component [20]. Compared to Darknet-53, CD53s-tiny has a significantly reduced number of layers, enabling faster computation. The model was initially trained on the MSCOCO object detection dataset, which contains 80 object classes. Pre-training was performed using the `yolov4.weights` file on an RTX 2070 GPU, achieving an inference speed of approximately 50–100 frames per second (FPS). To enable deployment on the NVIDIA Jetson Nano, the original YOLOv4-tiny model weights, initially formatted for Darknet, were converted into the TensorFlow model format. This conversion facilitates integration within the Jetson Nano ecosystem and supports object tracking through the Deep SORT algorithm.

Object Detection and Tracking

The Traffic Flow Detector functions to estimate the number of vehicles passing through a designated monitoring area. By knowing the number of vehicles passing through, the density of traffic flow on the route can be identified. The system operates in two primary phases: vehicle detection and vehicle tracking. In the first

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. XI No 3, Juni 2025, hlm. 533 – 540 ISSN 2550-0201 (Online)
DOI: http://dx.doi.org/10.33330/jurteksi.v11i3.3871
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

phase, the YOLOv4 object detection algorithm is employed to identify vehicles within video frames, generating bounding boxes around detected objects. At this stage, four vehicle categories are recognized: cars, motorcycles, buses, and trucks. In the subsequent phase, the Deep SORT algorithm is utilized to perform path tracking of vehicles that have been detected by YOLOv4

The YOLOv4 algorithm performs vehicle detection on each frame of a video or image sequence. However, vehicle detection systems continue to encounter several challenges, including occlusion—where a vehicle is partially or fully obstructed by another object—detection instability between consecutive frames due to rapid or inconsistent appearance and disappearance of objects (flickering), and varying lighting conditions that may affect detection accuracy [21]. These issues cause YOLOv4 to perform repeated detections on each new frame, which may result in a single vehicle being incorrectly identified as multiple distinct objects and consequently counted more than once. To address this limitation, the Deep SORT algorithm is integrated to associate detections across consecutive frames and preserve the unique identity of each object. Deep SORT enhances tracking stability under challenging conditions, including occlusion, changes in object appearance, and high traffic density [21].

Vehicle Counting

Two traffic lanes are monitored for traffic flow detection: Lane A, which covers the route from Jl. Datuk Setia Maharaja to Jl. Sudirman, and Lane B, which corresponds to the exit route from Jl. Kelapa Sawit, as shown in Figure 1. Traffic density is measured by counting vehicles that cross the respective lanes. Two marking lines are defined to represent each lane. When the centroid of a vehicle's bounding box—detected by YOLOv4—crosses one of the marking lines, the vehicle is counted. Vehicle detection is performed using YOLOv4, traffic flow tracking is handled by the Deep SORT algorithm, and vehicle counting is executed continuously until the final frame of the traffic video recorded at the Labersa intersection.

Figure 1. Intersection Lanes and Marking Lines

Information Extraction

Four types of vehicles detected along traffic lanes A and B are counted until the final frame, and the resulting data are stored in a `.csv` file. This data can be used for traffic analysis purposes in the management and planning of future traffic demand and highway usage.

RESULT AND DISCUSSION

The Traffic Flow Detector that has been installed on the NVDIA Jetson Nano is then tested with 7 input videos with a video resolution of 1280 x 720, 29 FPS. YOLOv4 and DeepSORT have successfully detected the number of vehicles passing on Lane A and Lane B. YOLOv4 successfully detects objects and predicts bounding boxes. Deep SORT successfully detects the path of objects that have previously been detected with

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. XI No 3, Juni 2025, hlm. 533 – 540 ISSN 2550-0201 (Online)
DOI: http://dx.doi.org/10.33330/jurteksi.v11i3.3871
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

the YOLOv4 algorithm. Deep SORT assigns a unique ID to each detected object and predicts its subsequent movement. When an object consistently appears across successive frames and crosses a predefined marking line, it is registered as a vehicle count on the corresponding lane. As illustrated in Figures 2–4, objects such as Car-1 and Car-3 were consistently tracked from frame 2 to frame 60. The Traffic Flow Detector demonstrated the capability to process video input at an average frame rate of 0.9 FPS on the NVIDIA Jetson Nano platform.

Figure 2. Detection on Frame 2

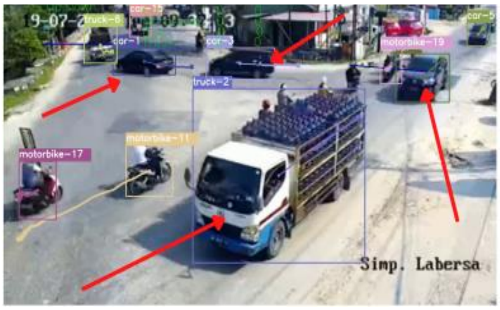

Figure 3. Detection on Frame 40

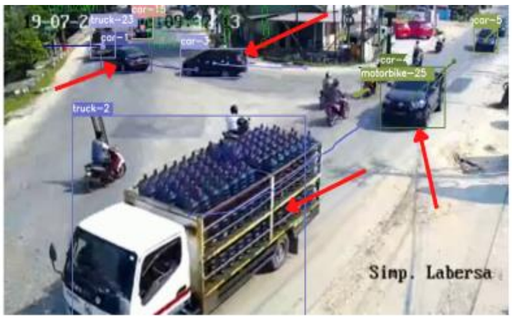

Figure 4. Detection on Frame 60

To evaluate the performance of the traffic flow detector system, accuracy and precision metrics are calculated based on a confusion matrix. The confu-sion matrix provides information on the comparison of the detection results carried out by the system with the actual results (ground truth) carried out manually by video observation. The confusion matrix format for the four vehicle classes is presented in Figure 5 [22]. To calculate accuracy and precision, it is essential to identify the predicted and actual values, represented by True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) metrics.

Confussion Matrix	Prediksi Mobil (1)	Prediksi Sepeda Motor (2)	Prediksi Bus (3)	Prediksi Truk (4)
Aktual Mobil (1)	TP1	FP12	FP13	FP14
Aktual Sepeda Motor (2)	FP21	TP2	FP23	FP24
Aktual Bus (3)	FP31	FP32	TP3	FP34
Aktual Bus (4)	FP41	FP42	FP43	TP4

Figure 5. Four Classes Confusion Matrix

The formula for calculating accuracy and precision values can be seen in Equations (1), (2), and (3) [23]. These calculations aim to evaluate the performance and reliability of the YOLOv4 and Deep SORT algorithms.

$$Akurasi\ Total = \frac{TP}{Total\ Dataset} \quad (1)$$

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

$$Presisi\ Total = \frac{\sum_{i=1}^{N} Presisi(i)}{\sum\ Kelas} \quad (3)$$

Figure 6 presents the confusion matrix for Video 1. Similar calculations were also performed for Videos 2 through 7. Based on the confusion matrix

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. XI No 3, Juni 2025, hlm. 533 – 540 ISSN 2550-0201 (Online)
DOI: http://dx.doi.org/10.33330/jurteksi.v11i3.3871
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

shown in Figure 6, the accuracy and precision values for Video 1 are as follows:

Table 1. Accuracy and Precision Calculation Results

Input Video	Recording	Manual Count (by video observation)				Overall Precision (%)	Overall Accuration (%)
		Car	Motorcycle	Bus	Truck		
Video 1	Night	251	415	1	21	24	5,6
Video 2	Morning	410	1213	2	31	73	13
Video 3	Morning	306	407	5	60	85	26
Video 4	Afternoon	381	600	2	42	68	16
Video 5	Morning	339	681	0	58	87	11
Video 6	Afternoon	549	711	3	43	98	11
Video 7	Afternoon	369	539	0	77	83	17

Accuracy $= \frac{39}{688} = 0{,}056 = 5{,}6\%$

Precision (Car) $= \frac{39}{39+1} = 0{,}975$

Precision (Motorcycle) $= 0$

Precision (Bus) $= 0$

Precision (Truck) $= 0$

Precision (Overall) $= \frac{0{,}975}{4} = 0{,}24 = 24\%$

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	39	0	0	1
	Sepeda Motor	0	0	0	0
	Bus	0	0	0	0
	Truk	0	0	0	0

Figure 6. Confussion Matrix for Video 1

The accuracy value obtained for Video 1 is 0.56%, while the overall precision is 24%. Video 1 was recorded at night under low lighting conditions, and the glare from vehicle headlights hindered effective detection by the Traffic Flow Detector system, resulting in a low number of detected vehicles. Equations (1) through (3) were also applied to calculate the total accuracy, precision, and overall precision for Videos 2 to 7, with the results presented in Table 1. The system achieved an

average precision of 74% and an average accuracy of 14.2

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	205	7	0	7
	Sepeda Motor	0	10	0	0
	Bus	0	0	0	0
	Truk	0	0	0	7

Figure 7. Confussion Matrix for Video 2

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	179	1	0	8
	Sepeda Motor	0	6	0	0
	Bus	0	0	2	1
	Truk	1	1	3	19

Figure 8. Confussion Matrix for Video 3

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	152	5	0	12
	Sepeda Motor	0	6	0	0
	Bus	0	0	0	0
	Truk	1	0	1	12

Figure 9. Confussion Matrix for Video 4

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	116	0	1	2
	Sepeda Motor	0	3	0	0
	Bus	0	0	0	0
	Truk	1	0	0	2

Figure 10. Confussion Matrix for Video 5

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	146	0	1	6
	Sepeda Motor	0	2	0	0
	Bus	0	0	1	0
	Truk	0	0	0	5

Figure 11. Confussion Matrix for Video 6

JURTEKSI (Jurnal Teknologi dan Sistem Informasi) ISSN 2407-1811 (Print)
Vol. XI No 3, Juni 2025, hlm. 533 – 540 ISSN 2550-0201 (Online)
DOI: http://dx.doi.org/10.33330/jurteksi.v11i3.3871
Available online at http://jurnal.stmikroyal.ac.id/index.php/jurteksi

		Prediksi			
		Mobil	Sepeda Motor	Bus	Truk
Aktual	Mobil	144	4	0	15
	Sepeda Motor	1	3	0	0
	Bus	0	0	0	0
	Truk	0	0	3	23

Figure 12. Confussion Matrix for Video 7.

CONCLUSION

This study successfully evaluated the implementation of an Intelligent Transportation System (ITS) using the YOLOv4-tiny and Deep SORT algorithms on the NVIDIA Jetson Nano, which demonstrated the capability to detect and track vehicle movements. Measurement results indicate that, overall, the system's accuracy is lower than its precision, with an average precision of 74% and an average accuracy of 14.2%, highlighting the need for improvements to reduce detection errors. Further optimization is necessary, particularly to enhance accuracy under low-light (nighttime) conditions and to reduce detection errors for high-density vehicle types, such as motorcycles. Recommended optimizations include improving the quality of video input, retraining the model using a more representative local dataset, adopting a lighter and more accurate detection model, and refining the tracking algorithm

BIBLIOGRAPHY

[1] A. M. Santos, C. J. A. Bastos-Filho, A. M. A. MacIel, and E. Lima, "Counting Vehicle with High-Precision in Brazilian Roads Using YOLOv3 and Deep SORT," *Proc. - 2020 33rd SIBGRAPI Conf. Graph. Patterns Images, SIBGRAPI 2020*, pp. 69–76, 2020.

[2] B. Y. Maram and A. Boukerche, "Context Prediction of Highways Based on The Vehicular Traffic Distribution," *IEEE Int. Conf. Commun.*, pp. 1–6, 2021.

[3] P. Khorramshahi, V. Shenoy, M. Pack, and R. Chellappa, "Scalable and Real-time Multi-Camera Vehicle Detection, Re-Identification, and Tracking," pp. 1–13, 2022.

[4] Y. Djenouri, A. Belhadi, G. Srivastava, D. Djenouri, and J. Chun-Wei Lin, "Vehicle detection using improved region convolution neural network for accident prevention in smart roads," *Pattern Recognit. Lett.*, vol. 158, pp. 42–47, 2022.

[5] Y.-L. Li, H.-T. Li, and C.-K. Chiang, "Multi-Camera Vehicle Tracking Based on Deep Tracklet Similarity Network," *Electron.*, 2022.

[6] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-Time Vehicle Detection Based on Improved YOLO v5," *Sustain.*, vol. 14, no. 19, 2022.

[7] M. I. Uddin, M. S. Alamgir, M. M. Rahman, M. S. Bhuiyan, and M. A. Moral, "AI Traffic Control System Based on Deepstream and IoT Using NVIDIA Jetson Nano," *Int. Conf. Robot. Electr. Signal Process. Tech.*, pp. 115–119, 2021.

[8] A. Saadeldin and M. M. Rashid, "Video-Based Vehicle Counting and Analysis using YOLOv5 and DeepSORT with Deployment on Jetson Nano," *Asian J. Electr. Electron. Eng.*, vol. 2, no. 2, pp. 11–20, 2022.

[9] F. Imanuel, S. K. Waruwu, A. Linardy, and A. M. Husein, "Review

Application of YOLO Algorithm for Detection and Tracking," *J. Comput. Networks , Archit. High Perform. Comput.*, vol. 6, no. 3, pp. 1378–1383, 2024.

[10] Al-qaness, M.A.A., Abbasi, A.A., and H. Fan, "An improved YOLO-based road traffic monitoring system," *Computing*, vol. 103, pp. 211–230, 2021.

[11] Z. Rahman, A. M. Ami, and M. A. Ullah, "A Real-Time Wrong-Way Vehicle Detection Based on YOLO and Centroid Tracking," *2020 IEEE Reg. 10 Symp. TENSYMP 2020*, no. June, pp. 916–920, 2020.

[12] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *Eur. Transp. Res. Rev.*, vol. 11, no. 1, 2019.

[13] Y. Zhao *et al.*, "A novel vehicle tracking id switches algorithm for driving recording sensors," *Sensors (Switzerland)*, vol. 20, no. 13, pp. 1–15, 2020.

[14] H. Park, S. H. Ham, T. Kim, and D. An, "Object Recognition and Tracking in Moving Videos for Maritime Autonomous Surface Ships," *J. Mar. Sci. Eng.*, vol. 10, no. 7, pp. 1–17, 2022.

[15] S. Park *et al.*, "Edge-computing Enabled Traffic State Estimation Based on Vehicle Tracking," *Soc. Sci. Res. Netw.*, 2022.

[16] S. Tak, J. D. Lee, J. Song, and S. Kim, "Development of AI-Based Vehicle Detection and Tracking System for C-ITS Application," *J. Adv. Transp.*, vol. 2021, pp. 78311–78319, 2021.

[17] X. C. Vuong, R. F. Mou, and T. T. Vu, "Vehicle Tracking and Speed Estimation Under Mixed Traffic Conditions Using Yolov4 and Sort: a Case Study of Hanoi," *Transp. Probl.*, vol. 17, no. 4, pp. 17–26, 2022.

[18] J. Zhao *et al.*, "Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4," *IEEE Access*, vol. 10, no. i, pp. 8590–8603, 2022.

[19] X. Liu and Z. Zhang, "A Vision-Based Target Detection, Tracking, and Positioning Algorithm for Unmanned Aerial Vehicle," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021.

[20] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. June, pp. 13024–13033, 2021.

[21] S. Kumar *et al.*, "Fusion of Deep Sort and Yolov5 for Effective Vehicle Detection and Tracking Scheme in Real-Time Traffic Management Sustainable System," *Sustain.*, vol. 15, no. 24, 2023.

[22] F. E. Purwiantono and A. Aditya, "Klasifikasi Sentimen Sara, Hoaks Dan Radikal Pada Postingan Media Sosial Menggunakan Algoritma Naive Bayes Multinomial Text," *J. Tekno Kompak*, vol. 14, no. 2, p. 68, 2020.

[23] F. Rofii, G. Priyandoko, M. I. Fanani, and A. Suraji, "Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis Yolov4 Deep Neural Networks," *Teknik*, vol. 42, no. 2, pp. 169–177, 2021.