Signal and Image Processing Letters
Vol. 7., No. 2, 2025, pp. 105-114
ISSN 2714-6677
https://simple.ascee.org/

Vehicle Detection and Tracking using Coarse-to-Fine Module and Spatial Pyramid Pooling–Fast with Deep Sort

Anita Nur Widdia Saputri [a,1], Aria Hendrawan [a,2,*], Rofiatul Khoiriyah [b,3]

[a] Faculty of Information Technology, Universitas Semarang, Jl. Soekarno Hatta, Semarang, Central Java 50196, Indonesia
[b] Master of Information Systems, Postgraduate School of Diponegoro University, Jl. Imam Bardjo SH No. 5, Semarang, Central Java 50135, Indonesia
[1] anitanurwiddias@gmail.com; [2] ariahendrawan@usm.ac.id; [3] rofiatulkhoiriyah@students.undip.ac.id
* corresponding author

ARTICLE INFO

Keywords
Traffic Challenges
Transportation System
Traffic Management
Vehicle Detection
Vehicle Tracking

ABSTRACT

Semarang City, a rapidly growing urban area in Indonesia, faces significant traffic challenges stemming from the widespread use of motorcycles, an inefficient public transportation system, and accelerated urban development. These factors contribute to congestion and complicate traffic management efforts. To address this issue and enhance monitoring capabilities, this study develops an automatic vehicle detection system utilizing the YOLOv8 algorithm, applied to CCTV footage obtained from TILIK SEMAR, a local traffic surveillance initiative. The research methodology encompasses several key stages: data collection from real-world traffic scenarios, meticulous annotation of vehicle types, model training using the YOLOv8 framework, and performance evaluation conducted at two distinct locations in Semarang—Banyumanik and Thamrin Pandanaran. The trained model achieved an impressive average accuracy, measured as mean Average Precision (mAP50), exceeding 97%, with a rapid processing time of 4.2 milliseconds per image, making it suitable for real-time applications. Among vehicle categories, the highest detection accuracies were recorded for buses at 99.3% and box trucks at 99.5%, reflecting the model's robustness for larger vehicles. However, motorcycles presented a challenge, with a lower mAP50-95 score of 64.3%, attributed to variations in shape, size, and lighting conditions. Overall, the system successfully identified 96.77% of 3,036 vehicles across the test dataset, demonstrating strong generalization across diverse traffic conditions. These findings validate YOLOv8 as an effective tool for real-time traffic monitoring in urban settings. Future enhancements will focus on expanding dataset diversity and improving performance under challenging environmental factors, such as adverse weather or low-light scenarios, to further refine the system's reliability.

1. Introduction

The city of Semarang faces complex traffic challenges due to the high usage of motorcycles, inefficient public transportation, and rapid urban development. According to Suseno *et al*. (2020), motorcycles accounted for 79% of total transportation modes in 2014, followed by private cars at 18% and other vehicles at 3%. Meanwhile, the Bus Rapid Transit (BRT) system in Semarang continues to struggle with various issues in meeting efficient service standards [1]. The rapid urban expansion has also led to increasing congestion, particularly in areas surrounding universities [2][3]. Additionally, the influence of Trans Semarang Bus routes on land development and improved accessibility has contributed to changes in traffic patterns [4].

To address traffic issues, the Semarang city government has implemented a CCTV-based traffic monitoring system known as TILIK SEMAR. However, this system still faces various limitations, such as funding and resource constraints that hinder the effective implementation of CCTV analytics technology [5]. Other challenges include difficulties in system integration with urban infrastructure, limitations in human monitoring, and environmental and operational constraints [6][7]. Furthermore, low-light conditions, privacy concerns, and ethical issues remain key considerations in the development of this system [8][9].

Artificial intelligence (AI)-based object detection has emerged as a potential solution to enhance the effectiveness of traffic monitoring. Object detection algorithms enable real-time monitoring, vehicle identification, and more accurate traffic analysis [10]-[12]. With the ability to optimize traffic signals and identify congestion points, this technology plays a crucial role in modern traffic management [13]. Additionally, AI-based systems can support automatic accident detection, improving road safety through faster and more precise responses [12].

Object detection models that are widely used in traffic monitoring or vehicle classification include YOLO (You Only Look Once) and CNN (Convolutional Neural Network). This algorithm offers high processing speed while maintaining good accuracy, making it an ideal solution for real-time applications [14]-[16]. YOLO has undergone multiple developments, from YOLOv3 to YOLOv8, each bringing improvements in accuracy and efficiency [16][17]. YOLOv8 [18]-[21], as the latest variant, adopts a more optimized architecture with enhanced detection of small objects and adaptability to complex environmental conditions [22]. The effectiveness of deep learning methods in vehicle classification using CNN has also been demonstrated in the classification of domestic car types in Indonesia [23].

Beyond object detection, vehicle tracking is a critical aspect of traffic monitoring systems. One effective tracking method is Deep SORT, which combines object detection with a Kalman filter and feature embedding to improve the accuracy of vehicle identification across frames [24]. The combination of YOLOv8 with Deep SORT has been proven to enhance stability and reliability in tracking dynamic objects in dense urban environments [16],[22].

To further improve detection and tracking performance, the Coarse-to-Fine Module and Spatial Pyramid Pooling – Fast (SPPF) approach has been applied. The Coarse-to-Fine technique enables more precise object identification, while SPPF enhances computational efficiency in feature extraction [16],[25]. Integrating these methods with YOLOv8 can result in a more adaptive system capable of handling various dynamic traffic conditions.

Considering the traffic challenges in Semarang, the implementation of YOLOv8-based vehicle detection combined with Deep SORT, the Coarse-to-Fine Module, and SPPF presents a promising solution. This technology not only enhances traffic monitoring effectiveness but also contributes to more efficient transportation management. Therefore, this study aims to develop a reliable and high-speed vehicle detection and tracking model for complex urban traffic scenarios, with a particular focus on implementation in the city of Semarang.

2. Research Method

Research method explains research chronological, including research design, research procedure (in the form of algorithms, pseudocode or other related things), how to test, and data acquisition process [26][27]. Any description related to research method should be supported by reference. The research method consists of several key stages to address the identified problems and develop an effective solution can be seen in Fig. 1. These stages include a literature review, needs analysis, data collection, implementation, and testing. The literature review focuses on identifying relevant theories and prior studies, particularly those related to the YOLO algorithm for object detection and vehicle classification using Traffic CCTV videos. This stage involves searching academic sources using Google Scholar with relevant keywords. The needs analysis determines the required software and hardware for the research. The hardware used includes a Dell Vostro 5402 i7-1165G7 laptop with 16GB RAM and a 14-inch monitor. The software consists of a 64-bit operating system, the Python programming language, and the OpenCV library.

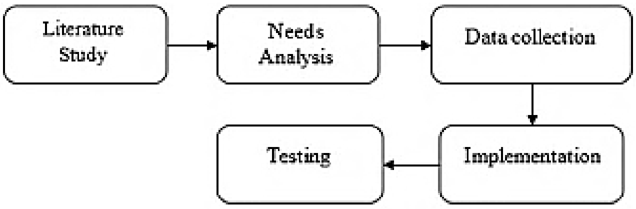


Figure 1. Research Method Stages

Data collection is conducted using Traffic CCTV footage obtained from the TILIK SEMAR website, which is managed by the Semarang City Government. This data serves as the primary material for the study [5]. During implementation, video data in .mp4 format is converted into image sequences. The object tracking process is then applied by assigning bounding boxes and labeling vehicle types such as motorbikes, cars, buses, and trucks. The YOLO algorithm is used to classify these objects. YOLOv8 [20], the latest version of YOLO released in 2020, introduces several improvements, including a more efficient backbone network, multi-scale prediction, and a new anchor system. The model comprises a backbone network for feature extraction, a neck with cross-layer connections for refinement, and a head for predicting bounding boxes and object classes. This version features Anchor-Free Detection, eliminating the need for predefined anchor boxes by predicting object centers directly. It also incorporates C2f (Coarse-to-Fine Module) to enhance feature extraction and SPPF (Spatial Pyramid Pooling - Fast) for improved multi-scale object detection and faster inference. These advancements make YOLOv8 a highly efficient choice for real-time vehicle detection.

3. Results and Discussion

In this study, the YOLOv8 algorithm was implemented to detect and classify vehicles from CCTV traffic video recordings obtained through the TILIK SEMAR portal in Semarang City. The primary objective of this research is to enhance the accuracy of vehicle detection based on type, including cars, motorcycles, buses, trucks, and other categories, while considering variations in lighting conditions and camera angles [1][2].

3.1. Data Collection

The initial stage of the study involved traffic video recording. Two video datasets with different durations and road locations were collected. The first dataset consists of an MP4-format video recorded on Jalan Thamrin Pandanaran, with a duration of 40 seconds, a resolution of 1280 × 720 pixels, and a frame rate of 30 frames per second (fps). The second dataset is an MP4-format video recorded in Banyumanik, with a duration of 69 seconds, a resolution of 640 × 480 pixels, and a frame rate of 18.59 fps. The corresponding video frames are shown in Fig. 2.

Figure 2. Thamrin Pandanaran Traffic Video Dataset

3.2. Dataset Frame Extraction

Each paragraph consists of one main sentence and several explanatory sentences. The explanations should be delivered systematically and provide information about how the authors do, related to data, methods, or stages that conducted.

Frame extraction is an essential step in preprocessing image datasets for object detection models like YOLOv8. In this study, the Roboflow platform was used for dataset processing. First, a new project was created in Roboflow, and the video dataset files were uploaded. After uploading, frames were extracted at predefined time intervals based on the frame rate (FPS) of each video. For YOLOv8 compatibility, an output resolution of 640 × 640 pixels was selected. Once the extraction settings were finalized, Roboflow automatically extracted frames according to the specified parameters. After processing both video datasets, a total of 2,223 images (frames) were generated for annotation and model development.

3.3. Data Anotation

Annotating extracted frames is a crucial step in preparing datasets for training object detection models like YOLOv8. After obtaining 2,223 frames, object classes were defined, including "Bus," "Car," "Motorcycle," "Pick-Up Car," "Truck," and "Truck Box." These predefined classes were used to label objects in each frame [4][5]. Bounding boxes were manually drawn around detected objects, and each bounding box was assigned a corresponding label. To speed up the annotation process, Roboflow’s auto-labeling feature was used, enabling automatic object detection and bounding box placement. These annotations were then reviewed and manually corrected to ensure accuracy. The dataset was then exported into a YOLO-compatible format (YOLO TXT), storing class ID, bounding box coordinates, and bounding box size relative to the image dimensions. A final verification step ensured that all frames were correctly annotated for training the YOLOv8 model [6][7].

3.4. Dataset Distribution

After frame extraction and annotation were completed, the dataset was divided into three subsets: training, validation, and test sets. The total dataset consisted of 2,223 images, which were split into 1,546 images for training, 450 images for validation, and 227 images for testing. The dataset was divided in a 70:20:10 ratio while considering the distribution of object classes.

3.5. Model Training

After preparing the data, the YOLOv8 model was trained on Google Colab by installing the Ultralytics package, importing the YOLO class, downloading the dataset from Roboflow with an API key, and using Albumentations for image augmentation. The model was trained for 25 epochs, followed by generating a confusion matrix and graphs to evaluate performance, and displaying predictions on validation data to classify vehicles like cars, motorcycles, buses, and trucks.

3.6. Model Validation

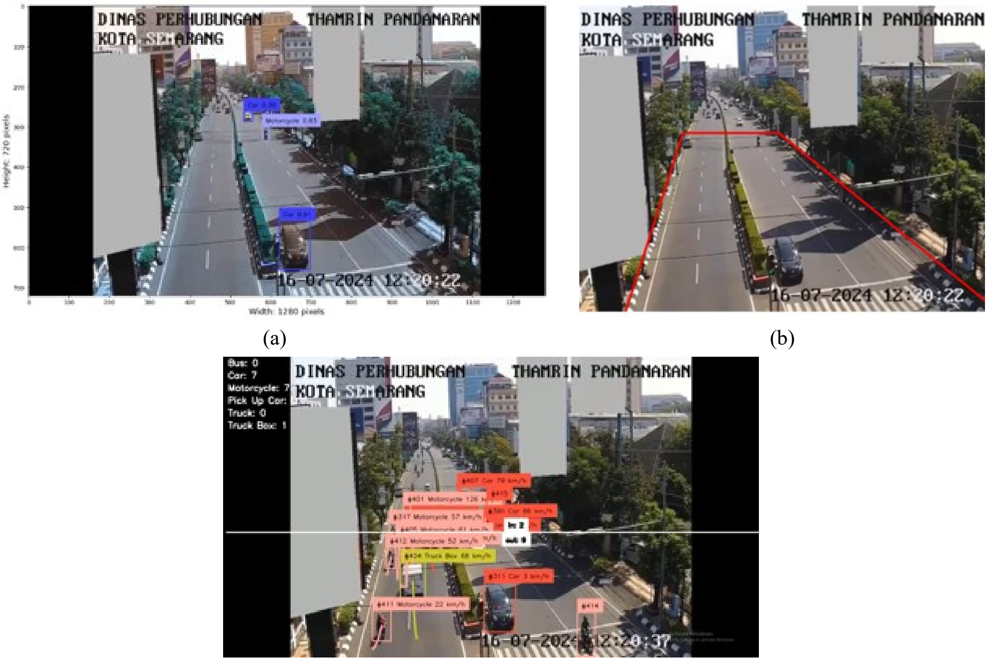
After training, the YOLOv8 model was evaluated on validation data using a dataset configuration defined in the data.yaml file. The results showed high performance, with mAP50 reaching 0.986 and mAP50-95 at 0.76. Precision and recall were high for all classes, but the motorcycle class had lower performance (0.643), indicating room for improvement. This evaluation helps determine whether the model is sufficient or requires further refinement.

3.7. Model Testing with Video

The trained YOLOv8 model was tested on live video data by installing the supervision library (version 0.18.0) for annotation and visualization, preparing the model for vehicle detection, mapping object classes to focus on relevant vehicles, and processing a video dataset (640 × 480 pixels, 18 fps, 1,294 frames) to incorporate object counting and speed detection features. Using this data, the video duration was calculated as:

$$Video\ Duration\ (s) = \frac{total_{frames}}{fps} = \frac{1294}{18} = 71.89\ s\ (about\ 1\ minute\ 12\ seconds) \quad (1)$$

Next, bounding box annotations were applied, as shown in Fig. 3(a). The annotation process continued with defining polygons and targets, drawing them on video frames in red with a thickness of 4 pixels, as illustrated in Fig. 3(b). Afterward, OpenCV was used to retrieve video width, height, and frame rate for coordinate extraction. The final step involved initializing object tracking, defining class names, creating bounding box annotations, tracing objects, and calculating object velocity. The processed video frames included object detection counts displayed in the top-left corner.



(a) (b) (c)

Figure 3. (a) is Preview of bounding box annotation on Banyumanik dataset. (b) is Preview of speed tracking polygons on the Thamrin Pandanaran dataset and (c) is Preview of the Thamrin Pandanaran video dataset results

3.8. Research Results

After testing the model against vehicle detection in traffic. From the dataset tested both on training, validation, and testing data produced almost the same object detection output. In this case, testing was carried out using 2 video datasets with different locations. The image shows in Table 1 and Table 2 that from the 2 video datasets from the model testing, it can detect predetermined objects and can display the speed of existing objects. After that, objects that cross the line will be counted as defined from top to bottom as "In" and from bottom to top as "Out". The following are the results of the analysis of several sample image datasets. Testing is also done to evaluate the performance of the developed system. Evaluation of system performance by calculating the accuracy value of the detection results and measuring the speed of the detection process. Based on the confusion matrix table from the classification results, the results are as in Table 3. The results showed strong detection performance for large vehicles like buses and box trucks with mAP50 over 99%. However, motorcycles had lower detection performance with mAP50-95 at 64.3%, due to occlusion and lighting conditions. Testing with 3,036 vehicles resulted in 2,938 correctly detected, achieving 96.77% accuracy. The system processed 1,294 frames at 4.2 ms/frame.

In a previous study by Jung Hyun Ki entitled "Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions" discussed the performance improvement of the YOLOv5 model for object detection using drone images in various environmental conditions. This study aims to improve the accuracy of object detection in situations such as illegal immigration, industrial and natural disasters, and searching for missing people or objects. After conducting experiments using datasets covering various environmental conditions, the improved YOLOv5 model showed significant performance improvements. This model achieved a Precision value of 90.7%, Recall of 87.4%, F1 Score of 88.8%, and mAP of 95.5%. The article written by Alvi Khan Chowdhury entitled "Oil Palm Fresh Fruit Branch Ripeness Detection Using YOLOV6 Algorithm" discusses the application of the YOLOv6 algorithm to automatically detect the ripeness level of oil palm Fresh Fruit Bunches (FFB). This study aims to improve efficiency in determining the optimal harvest time, which currently still relies on manual assessment by workers on plantations. The model was trained using two variants, namely YOLOv6s (small) and YOLOv6m (medium), with 100 training epochs. The evaluation results showed that the YOLOv6m model performed better than YOLOv6s, with Precision of 36.9%, Recall of 30%, F1 Score of 33.1%, mAP(50) of 36.9%, and mAP(50–95) of 16.5%. In the article written by Ardiansyah entitled "Detection and Classification of Diseases in Coffee Leaves Using Yolov7" discusses the use of YOLOv7 for detection and classification of diseases in coffee leaves. The results of the study using Google Colab devices with Tesla T4 GPUs showed that YOLOv7 provided excellent performance with Precision 0.926, Recall 0.932, mAP@IoU 0.5 of 0.956, mAP@IoU 0.5:0.95 of 0.927, and F1-Score 0.93 for all data classes. The best results for binary classification were, Precision 0.991, Recall 1, mAP@IoU 0.5 of 0.998, and F1-Score 0.99. Based on the YOLO version comparison Table 4, it can be concluded that YOLOv8 used in this study shows the best overall performance compared to other versions. This is evidenced by the highest mAP (mean Average Precision) value, which is 98.6%, as well as the highest Precision and Recall values (96.3% and 96.6%). While YOLOv5 and YOLOv7 show good performance.

Table 1. Sample Analysis of Detection Results of Video Dataset 1 Banyumanik

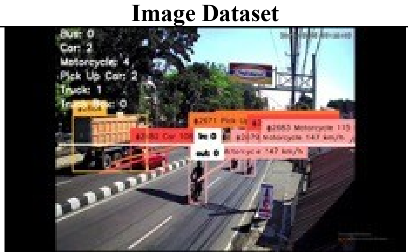
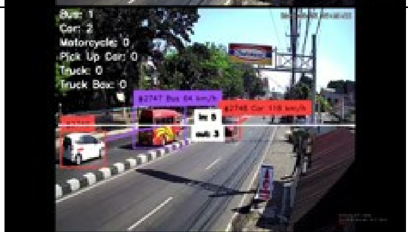
Image Dataset	Information	Status
	Sample video 1 second 1 shows that 4 motorcycles, 2 cars, 2 pick-up cars, and 1 truck are detected. And detecting the speed of each object	Success
	Sample video 1 second 15 shows that 2 Buses, 2 Cars are detected. And detects the speed of each object.	Success
	Sample video 1 second 38 shows that 2 motorcycles, 1 pick-up car, and 1 truck are detected. And detects the speed of each object.	Success
	Sample video 1 second 56 shows that 2 motorcycles and 2 cars are detected. And detects the speed of each object.	Success
	Sample video 1 second 71 shows that 4 motorcycles and 1 truck were detected. And detects the speed of each object.	Success

Table 2. Sample Analysis of Video Dataset Detection Results 2 Pandanaran

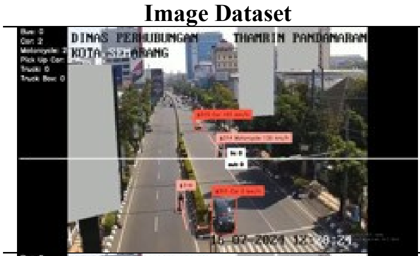
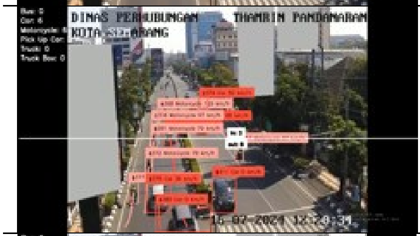
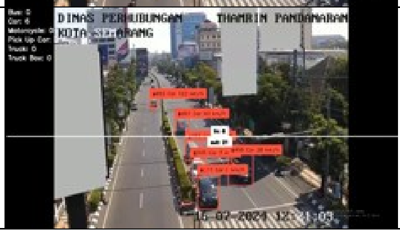
Image Dataset	Information	Status
	Sample video 2 seconds 1 shows that 2 motorcycles and 2 cars are detected. And detects the speed of each object.	Success
	The 2 second 11 video sample shows that 6 motorcycles and 6 cars are detected. And it detects the speed of each object.	Success
	The 20 second video sample shows that 6 motorcycles and 6 cars were detected. And it detects the speed of each object.	Success
	The 2 second 30 video sample shows that 4 motorcycles and 6 cars are detected. And it detects the speed of each object.	Success
	The 2 second 40 video sample shows that 4 cars are detected. And it detects the speed of each object.	Success

Table 3. Test Data Classification Results Based on Confusion Matrix

Data Classification Labels	Number of Detections (True)	Detection Results (actual)
Motorcycle	1358	1384
Car	868	890
Bus	38	42
Truck Box	35	38
Pick Up Car	138	152
Truck	501	530

Table 4. YOLO Version Comparison

YOLO version	Precision	Recall	mAP
YOLOv5	90.7%	87.4%	95.5%
YOLOv6m	36.9%	30%	36.9%
YOLOv7	92.6%	93.2%	95.6%
YOLOv8	96.3%	96.6%	98.6%

4. Conclusion

Based on the research conducted, YOLOv8 integrated with Deep SORT, Coarse-to-Fine, and SPPF modules proved effective for real-time vehicle detection and tracking. It achieved a mAP50 of 98.6%, mAP50-95 of 76%, and a counting accuracy of 96.77%. Motorcycle detection remains a challenge. Future research will focus on dataset diversity (weather, time), model optimization for edge devices, and enhancing small object detection under occlusion.

Despite these limitations, the model remains robust, even with imbalanced training data, where motorcycles dominate (46.279% of instances) while box trucks account for only 1.167%. The training and validation results confirm consistent performance across different IoU thresholds (50%-95%). Furthermore, vehicle counting evaluation demonstrated a 96.77% success rate, with 2,938 detected vehicles out of 3,036 total instances, reinforcing the model's effectiveness in real-world applications.

Given the significance of motorcycles in Semarang's traffic landscape, future improvements should focus on enhancing detection accuracy for smaller objects by incorporating advanced augmentation techniques, dataset balancing, and hyperparameter optimization. Additionally, integrating Deep SORT for tracking stability and Coarse-to-Fine Module with SPPF for computational efficiency could further refine traffic monitoring capabilities, making it a viable solution for urban traffic management in Semarang.

Acknowledgment

We express our gratitude to the Faculty of Information Technology, Universitas Semarang, for the support and resources provided during this research. Additionally, we appreciate the valuable feedback from colleagues and reviewers that helped improve this work.

References

[1] D. P. Suseno, A. B. Siswanto, M. A. Salim, and B. Rozak, “Mass transport rail based study for sustainable transportation in semarang,” *International Journal of Scientific and Technology Research*, vol. 9, no. 3, pp. 461–465, 2020, https://www.scopus.com/pages/publications/85083825087?inward.

[2] Y. Basuki and S. Rahayu, “Land use and trip production model in central and peri-urban Semarang to anticipate land use conversion in post pandemic era,” in *IOP Conference Series: Earth and Environmental Science*, vol. 1082, no. 1, p. 012015, 2022, https://doi.org/10.1088/1755-1315/1082/1/012015.

[3] O. R. Manullang, A. R. Rakhmatulloh, D. A. Sihaloho, and N. M. Samosir, “Changes of landuse in the campus area and their implications toward traffic condition,” in *IOP Conference Series: Earth and Environmental Science*, vol. 340, no. 1, p. 012034, 2019, https://doi.org/10.1088/1755-1315/340/1/012034.

[4] A. R. Rakhmatulloh, D. I. Kusumo Dewi, and D. M. K. Nugraheni, “Bus Trans Semarang toward Sustainable Transportation in Semarang City,” in *IOP Conference Series: Earth and Environmental Science*, vol. 409, no. 1, p. 012021, 2020, https://doi.org/10.1088/1755-1315/409/1/012021.

[5] D. S. Kencono and M. R. Ahsany, “The Use of CCTV Model Analysis for a Smart Environment in Semarang City,” in *E3S Web of Conferences*, vol. 440, p. 07006, 2023, https://doi.org/10.1051/e3sconf/202344007006.

[6] M. Abraham, N. Suryawanshi, N. Joseph and D. Hadsul, "Future Predicting Intelligent Camera Security System," *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pp. 1-6, 2021, https://doi.org/10.1109/ICITIIT51526.2021.9399597.

[7] A. Dabrowski, P. Matczak, A. Wójtowicz, and M. Leitner, “Identification of experimental and control areas for CCTV effectiveness assessment—the issue of spatially aggregated data,” *ISPRS Int J Geoinf*, vol. 7, no. 12, 2018, https://doi.org/10.3390/ijgi7120471.

[8] A. Hendrawan, R. Gernowo and O. D. Nurhayati, "Contrast Stretching and Contrast Limited Adaptive Histogram Equalization for Recognizing Vehicles Based on Yolo Models," *2023 International Conference on Technology, Engineering, and Computing Applications (ICTECA)*, pp. 1-6, 2023, https://doi.org/10.1109/ICTECA60133.2023.10490716.

[9] R. Surette, “The thinking eye: Pros and cons of second generation CCTV surveillance systems,” *Policing*, vol. 28, no. 1, pp. 152–173, 2005, https://doi.org/10.1108/13639510510581039.

[10] A. Bhowmik, R. Saha, S. Mishra, P. K. Pareek, and R. Garg, “Real-Time Object Detection in Road Traffic with Road Maintenance Capability,” in *Lecture Notes in Networks and Systems*, pp. 491–502, 2024, https://doi.org/10.1007/978-981-97-3817-5_35.

[11] H. Ghahremannezhad, H. Shi, and C. Liu, “Object Detection in Traffic Videos: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6780–6799, 2023, https://doi.org/10.1109/TITS.2023.3258683.

[12] K. S. Hansen, F. M. Bruun, F. Sermsar, M. Nygaard and M. Koca, "Comparative Analysis of SSD and Faster R-CNN in UAV-Based Vehicle Detection," *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1-6, 2024, https://doi.org/10.1109/IDAP64064.2024.10711057.

[13] K. Anitha, H. M. Manjula, H. P. Leelavathi and P. Swetha, "Automated Traffic Management Using Image Processing," *2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*, pp. 1-4, 2023, https://doi.org/10.1109/ICCAMS60113.2023.10525834.

[14] S. D. Alapati, M. Arunachalam, C. Chennamsetty, P. Dantam, and A. Dabbara, “Real-time object détection in video for traffic monitoring,” in *AI Tools for Protecting and Preventing Sophisticated Cyber Attacks*, pp. 166–179, 2023, https://doi.org/10.4018/978-1-6684-7110-4.ch008.

[15] M. Flores-Calero *et al.*, “Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review,” *Mathematics*, vol. 12, no. 2, 2024, https://doi.org/10.3390/math12020297.

[16] Y.-L. Chen, C. He, and B. Ren, “Traffic Scene Object Detection Based on YOLO Algorithm,” in *ACM International Conference Proceeding Series*, pp. 314–318, 2024, https://doi.org/10.1145/3687488.3687543.

[17] K. V. Mahalakshmi, S. Kalagara, A. Gudivada, S. Vankudoth and L. Neeli, "Vehicle Detection Using CNN and YOLOv3," *2023 4th IEEE Global Conference for Advancement in Technology (GCAT)*, pp. 1-6, 2023, https://doi.org/10.1109/GCAT59970.2023.10353257.

[18] Y. Swathi and M. Challa, “YOLOv8: Advancements and Innovations in Object Detection,” in *Lecture Notes in Networks and Systems*, pp. 1–13, 2024, https://doi.org/10.1007/978-981-97-1323-3_1.

[19] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pp. 1-6, 2024, https://doi.org/10.1109/ADICS58448.2024.10533619.

[20] I. V Prakash and M. Palanivelan, “A Study of YOLO (You Only Look Once) to YOLOv8,” in *Algorithms in Advanced Artificial Intelligence*, pp. 257–266, 2024, https://doi.org/10.1201/9781003529231-40.

[21] A. Gupta *et al.*, “Traffic Light Detection for Self-Driving Cars using the YOLOv8 architecture,” in *ACM International Conference Proceeding Series*, pp. 263–269, 2024, https://doi.org/10.1145/3660853.3660925.

[22] X. Hong, J. Huang, W. Zhao, H. Zou, Z. Lin, and Y. Chen, “Object Detection for Traffic Management Based on YOLO,” in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 13018, pp. 157-161, 2024, https://doi.org/10.1117/12.3024069.

[23] E. A. Nabila, C. A. Sari, E. H. Rachmawanto, and M. Doheir, “A Good Performance of Convolutional Neural Network Based on AlexNet in Domestic Indonesian Car Types Classification,” *Advance Sustainable Science, Engineering and Technology*, vol. 5, no. 3, p. 0230302, 2023, https://doi.org/10.26877/asset.v5i3.16854.

[24] J. Azimjonov and A. Özmen, “A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways,” *Advanced Engineering Informatics*, vol. 50, 2021, https://doi.org/10.1016/j.aei.2021.101393.

[25] A. Makhmutova, I. V Anikin, and M. Dagaeva, "Object Tracking Method for Videomonitoring in Intelligent Transport Systems," in *Proceedings - 2020 International Russian Automation Conference, RusAutoCon 2020*, pp. 535–540, 2020, https://doi.org/10.1109/RusAutoCon49822.2020.9208032.

[26] W. A. Kusuma and L. Husniah, "Skeletonization using thinning method for human motion system," in *2015 International Seminar on Intelligent Technology and Its Applications, ISITIA 2015 - Proceeding*, pp. 103–106, 2015, https://doi.org/10.1109/ISITIA.2015.7219962.

[27] A. E. Minarno, Y. Munarko, A. Kurniawardhani, and F. Bimantoro, "Texture Feature Extraction Using Co-Occurrence Matrices of Sub-Band Image For Batik Image Classification," in *Information and Communication Technology (ICoICT)*, pp. 249–254, 2014, https://doi.org/10.1109/ICoICT.2014.6914074.