International Journal of Informatics and Computation (IJICOM)
Vol. 6, No.1, August 2024
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Line Crossing Detector System for Real-Time Over-Taking Vehicle Detection

Ahmad Nanda Yuma Rafi[1], Mohamad Yusuf[2]

Abstract

This study introduces a novel method for detecting overtaking vehicles by integrating Virtual Line Detection with the YOLOv8n algorithm. The objective is to enhance road safety by accurately identifying and tracking vehicles as they overtake, which is crucial for preventing. The research demonstrates the effectiveness of this approach, achieving a detection accuracy rate of 80.95% using line-crossing detection techniques. This high level of accuracy underscores the potential of the system to reliably identify overtaking maneuvers in traffic conditions. Furthermore, this innovative method holds promising implications for enhancing safety riding by providing real-time alerts to drivers and preventing infrastructure loss resulting from traffic incidents. Our findings suggest that integrating advanced detection algorithms like YOLOv8n with virtual line detection can be a viable solution for modern traffic safety challenges.

Keywords:
YOLOv8n, Vehicle Detection, Line Crossing Detector, CNN

1. Introduction

In modern transportation systems, ensuring the safety of vehicles on highways is of paramount importance. One critical concern is the occurrence of over-taking incidents, which pose significant risks to both drivers and passengers [1]. These incidents not only result in potential loss of life but also cause substantial damage to infrastructure and vehicles. According to Halim et al (2020), There are four factors causing transportation accidents in Indonesia: the condition of transportation infrastructure, and human and natural factors. However, among these four factors, human negligence is the main factor causing the high number of traffic accidents [2].

Over-taking incidents refer to situations where a vehicle loses stability and tips onto its side or roof. These incidents can occur due to various factors, including high-speed turns, sudden maneuvers, adverse weather conditions, or improperly distributed loads. The accidents are particularly dangerous on highways, where vehicles are traveling at higher speeds and have limited maneuvering space. Furthermore, the urgency of addressing overtaking incidents cannot be overstated. As mentioned by Septianingtyas (2019) The presence of aggressive drivers, who tend to react emotionally to the behavior of other drivers, often leads them to attempt overtaking. This occurrence can certainly contribute to an increase in traffic accidents and also affect the severity of accident victims. Traditional vehicle safety features such as airbags and seat belts are crucial, but they primarily focus on minimizing injuries after an accident has occurred. A real-time warning system specifically designed for over-taking scenarios can provide a crucial layer of prevention,

Corresponding Author: Mohamad Yusuf (mhd.yusuf@mercubuana.ac.id)

giving drivers the information they need to make split-second decisions that can mean the difference between a safe passage and a catastrophic event [3].

According to data from the Traffic Corps of the Republic of Indonesia in 2021, the total number of traffic accidents increased to 103,645 in 2021, with 22,033 accidents occurring on toll roads and 1,275 on arterial roads. This represents a 3.62% increase compared to the previous year, which had 100,028 cases [4]. One of the solutions to this problem is using Line Crossing Detector (LCD) to monitor the vehicle behind the user-vehicle. As mentioned by MELDIYANA, Salma et al. (2022) LCD is one of the real-time situational awareness products capable of monitoring key areas (specifically designated as surveillance/clear areas) [1]. This system can detect objects crossing the virtual danger and/or warning lines that have been set, identify the objects, and send notifications when a specific object crosses the virtual danger and/or warning lines.

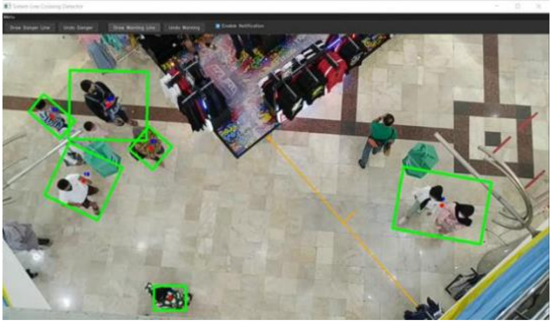

Fig 1. Line Crossing Detector in CCTV (MELDIYANA, Salma et al. (2022))

The primary theoretical framework underpinning this research revolves around the utilization of Line Crossing Detectors as a fundamental mechanism for monitoring and recording movements within specific areas, as highlighted by Ahmed et al. (2021). This theoretical concept forms the cornerstone of understanding the functionality and significance of line-crossing detection algorithms within the realm of surveillance and object movement tracking.

Line Crossing Detectors serve as critical components in surveillance systems, aiding in the observation and documentation of the entry and exit of objects within designated areas emphasizes the utility of these detectors in capturing and analyzing movements, thus laying the groundwork for their application in various domains, including traffic monitoring and vehicular movement analysis [5].

On the other hand, Real-time object detection continues to be difficult due to variations in object sizes, shapes, inference speed, and noise. This is particularly challenging for our application, as the moving vehicle can rapidly alter its location, scale, rotation, and trajectory. Moreover, detecting vehicle types such as cars, motorbikes, buses, and trucks from captured images refers to fitting the target object across consecutive frames. This can be a challenge if the object is in different conditions, such as variations in viewing angle, vehicle direction, lighting conditions, and variations in vehicle image details. This can result in incorrect feature extraction. Additionally, high computation prolongs the data training process, impacting the performance and speed of object detection [6].

To prevent accidents, especially those involving overtaking vehicles, we propose combining YOLOv8n with a line-crossing detector integrated into an IoT system. This approach leverages YOLOv8n's real-time object detection capabilities and the accuracy of line crossing detection to monitor and track vehicle movements effectively. By integrating this system with IoT technology, we can provide real-time alerts and enhance road safety, particularly in scenarios where vehicles are overtaking, thereby reducing the risk of accidents and improving traffic management.

2. Related Works

Line Crossing Detector has become a tool to monitor an object that is detected in some areas. The capability to combine with an alarm system makes this model perfect for detecting moving activity. Recently, MELDIYANA, Salma, et al. (2022) successfully implemented a Line Crossing Detector to monitor an area by using CCTV combined with YOLOv3 and COCO dataset as a media for detecting the object. The proposed system can detect objects crossing designated virtual danger or warning lines with a confidence level of 90% and a significance level of 10%. Moreover, it can identify objects after they cross these virtual lines and is capable of providing notifications when objects pass the established virtual danger or warning lines.

Furthermore, Ahmed et al. (2021) utilized the Single Shot Multibox Detector (SSD) model with MobileNetv2 as the base network for detecting people. The accuracy of the detection model was enhanced using a transfer learning approach. Two virtual lines were defined to count how many people were leaving and entering the scene. To assess performance, experiments were conducted using different video clips. The results indicated that transfer learning significantly improved the overall detection performance of the system, achieving an accuracy of 95%.

Samuel et al. (2017) researched counting vehicles using Line Detection. The proposed system employed multiple virtual lines as sensors to count moving objects and monitor live traffic flow. This method proved effective, achieving high accuracy in sufficient lighting conditions, such as daylight. The detection accuracy approached 100% due to individual line monitoring of specific screen areas. The system's ability to place multiple line sensors on the screen allowed for unlimited monitoring locations, provided the object size remained above a certain threshold. Moreover, the flexibility of virtual line placement, without strict positioning rules, enabled the system to count objects from various angles of traffic surveillance cameras effectively. This feature makes the application highly adaptable for existing surveillance systems with multiple cameras, eliminating the need to adjust camera positions.

In terms of object detection, YOLOv8 demonstrates enhanced performance in terms of object detection accuracy. The model has undergone rigorous training on several datasets encompassing a wide range of item categories, enabling it to detect distinct objects with a high degree of precision accurately [8]. Generally, deeper layers in CNNs extract more granular/complex/low-level feature representations. YOLOv8 incorporates this idea into its architecture by having repeating modules and multiple detection heads when making its prediction [9].

In our latest research, we explore the utilization of Transfer Learning when dealing with challenging datasets. However, there are trade-offs, particularly concerning precision and key metrics. The decision to freeze specific layers introduces a combination of factors that influence the overall effectiveness of the model [12]. In addition, YOLOv8 can integrate feature maps at various scales to learn about object shapes and textures, enhancing its accuracy in most object detection tasks. Reis et al. (2023) mentioned YOLOv8 backbone consists of four sections, each with a single convolution followed by a c2f module 27. Reis et al. (2023) also compared the YOLOv5s and YOLOv8 performance for detecting Aerial objects with some challenges such as detecting and classifying extremely small objects, identifying flying objects that blend into their background, and classifying different types of flying objects. As a result, YOLOv8 surpasses YOLOv5 in aspects including a better mAP. Along with a better mAP, this shows that YOLOv8 has fewer outliers when measured against the RF100 which is a 100-sample dataset from the Roboflow universe which is a repository of 100,000 datasets.

This supports the study's exploration of these architectural enhancements for improved vehicle detection capabilities, aligning with the research's objective of evaluating and enhancing algorithms for accurate vehicle detection and tracking.

3. Proposed Method

3.1 YOLOv8n

Ultralytics YOLOv8 is the latest iteration of the YOLO object detection and image segmentation model. As a state-of-the-art (SOTA) model, YOLOv8 builds on the achievements of previous versions, offering new features and enhancements for improved performance, flexibility, and efficiency [10]. It is designed with an emphasis on speed, size, and accuracy, making it an excellent choice for a range of vision AI tasks. YOLOv8 surpasses earlier versions by integrating innovations such as a new backbone network, a new anchor-free split head, and new loss functions. These improvements enable YOLOv8 to deliver superior results while maintaining a compact size and exceptional speed [11]. This would be good since the real-time system for detecting vehicles on the highway needs a good accuracy and speed detection model.

In our model, we proposed an algorithm capable of detecting and tracking vehicles in rapidly changing circumstances, such as on highways or country roads. To achieve this, we chose the YOLOv8n model due to its high detection speed and small size, making it suitable for running on compact CPUs like Raspberry Pi. Although YOLOv8n has lower accuracy compared to other models like YOLOv8m or YOLOv8s, it is sufficiently adequate for vehicle detection tasks.

3.2 i_region

In this research, several challenges were identified, including the possibility of false alarms when a vehicle passes by. Direct implementation of a bare Line Detection model detects all objects crossing the line, meaning even stationary objects passed by the user can trigger the alarm. To mitigate this, researchers proposed a model called i_region, which defines a specific rectangle area in the center of the camera's view. This ensures that only objects moving from behind and crossing the line trigger the alarm.

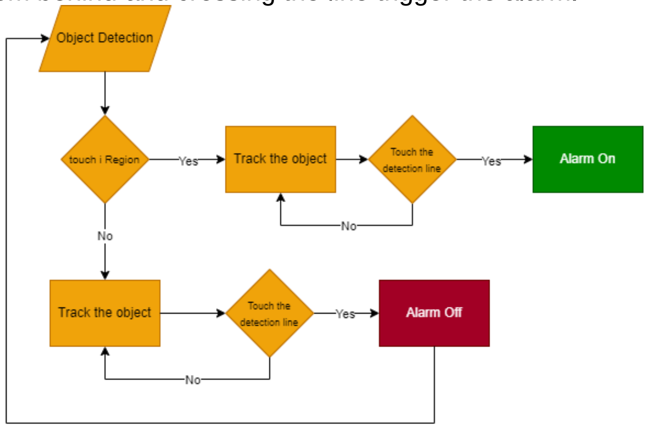

Fig. 2. System Flow

This is illustrated in Fig.1, where detected objects that are already in the i_region are assigned a value of 1, while those outside are assigned a value of 0 with the possibility to update the value into 1 when it enters the i_region area. By using this logic, false alarms caused by unmoving objects are prevented. Additionally, the model can accurately detect scenarios where a vehicle is initially overtaken by the user and then overtakes the user.

3.3 Line Detections

Since we need a detection model that can trigger an alarm when a tracked object passes by, we decided to use a Line Detection algorithm combined with an alarm warning system. This model is typically used for monitoring an area with CCTV as the visual input and corresponding lines to detect passing objects. We recommend this simple system because it does not require additional tracking components like GPS, speed sensors, or light sensors. We used two lines on each side of the camera, allowing for the detection of vehicles from the left or right. This setup can also serve as a warning system for drivers about vehicles in their blind spots.

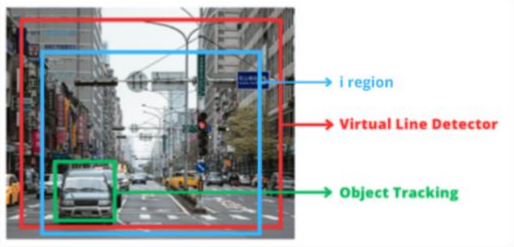

Fig. 3. Project Visualization

The Line Detection algorithm works by defining specific virtual lines within the camera's field of view. When an object crosses these lines, the system registers the event and triggers an alarm. This method is particularly effective for detecting moving objects in predefined areas, making it ideal for applications such as traffic monitoring and enhancing driver awareness. By strategically placing the lines on each side of the camera as shown in Fig.3, we ensure comprehensive coverage for detecting vehicles approaching from both directions. This is especially useful for identifying vehicles in blind spots, providing an additional layer of safety for drivers.

3.4 Performance Analysis

Some performance analysis will be conducted to ensure the effectiveness of the model. Accuracy, recall, and precision are the key metrics we will focus on to measure the model's performance. Accuracy measures the overall correctness of the detections, recall evaluates the model's ability to identify all relevant objects, and precision assesses the correctness of the detected objects. The formula for each parameter is:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

$$recall = \frac{tp}{tp+fn}$$

$$precission = \frac{tp}{tp + fp}$$

These metrics will help validate whether the model can reliably detect and track vehicles, especially during overtaking scenarios, thereby confirming its practical applicability and reliability.

4. Experimental Setup

4.1 Datasets

In this research, we decided to use the COCO dataset, which contains 80 classes and more than 200,000 labeled images. Since our project focuses on vehicle detection, we filtered the dataset to include only vehicle labels such as motorcycles, trucks, and cars. Additionally, we captured footage in an urban area (West Jakarta, Indonesia) to test our proposed model. This combination of the COCO dataset and real-world footage helps ensure the model's robustness and applicability in detecting vehicles in various environments.

4.2 Pseudocode

```
Loop:
Input data
Detect objects in the frame
Update object tracking

For each tracked object:
Draw a bounding box around the object
Calculate the center of the object

Check if the object is within the i region:
- If yes and the i_region value is 0 or not set:
Set i_region value to 1
- If no:
Set i_region value to 0

Check if the object crosses the line detection:
- For each tracked object with i_region value = 1:
Trigger the alarm and notification
Mark the object as passed
Reset i_region value to 0

Display the frame with the detected objects and
tracking information
```

4.3 Model Implementation

In our proposed model, after designing the system and running it with the test footage, we conducted a practical analysis to evaluate its effectiveness in detecting overtaking vehicles. Several key points emerged from this analysis. In urban areas with heavy traffic, the model successfully detected overtaking vehicles with 80.95% accuracy, 62.5% recall, and 83.33% precision, showcasing its potential in real-world scenarios.

Accuracy	**Recall**	**Precision**
80.95%	**62.5%**	**83.33%**

Table. 1. Performance Measure

However, there were instances of false alarms that the i_region logic could not resolve. These false alarms occurred when the user turned their vehicle to the left or right, causing the camera's point of view to change. As a result, objects detected in the i_region triggered the alarm, not because of overtaking, but due to the rotating camera following the user's vehicle position. This issue highlights the challenge of maintaining accurate detection in dynamic environments where the camera's perspective can shift significantly. Improving the model to account for these changes in viewpoint could enhance its reliability and reduce the occurrence of false alarms.

5. Result and Analysis

The test was conducted in West Jakarta, Indonesia, a bustling urban area with diverse traffic conditions, making it an ideal location for evaluating the performance of our object detection algorithm. The primary goal was to assess how well the algorithm could detect and track various objects, particularly vehicles, in a real-world environment. The results were promising, as the algorithm successfully detected and tracked the majority of objects, demonstrating its robustness and reliability in a dynamic setting.

Despite the false alarm caused by the rotating camera, the algorithm achieved a commendable accuracy rate of 80.95% in detecting overtaking vehicles. However, a noted challenge was the decrease in detection accuracy when vehicles were overtaking. During these maneuvers, the camera's point of view primarily captured the sides of the vehicles, which led to lower precision in detection (Fig.4 (b)). Even with this limitation, the application of i_region produced very good results. The precision value was high at 83.88%, indicating that the system was able to correctly identify and classify objects with a high degree of accuracy. This was particularly important in preventing false alarms, which can undermine the reliability and effectiveness of an object detection system as demonstrated in (Fig.4 (d)).

(a) (b) (c) (d)

Fig. 4. Result in True Positive (a), False Positive (b), True Negative (c), False Negative (d)

However, the recall value, which was 62.5%, revealed some areas for improvement. The recall metric measures the ability of the system to detect all relevant instances of objects, and a lower recall value suggests that some objects were not detected. In this case, the lower recall value was primarily due to a high true negative rate when many unmoving vehicles passed by the camera. This means that while the system was good at not flagging non-relevant objects (high precision), it also missed some relevant objects, resulting in a lower recall. This issue is understandable, given the context of the test environment where unmoving vehicles were prevalent, but it does highlight an area where the system could be improved.

6. Conclusion

The test that has been conducted confirmed the robustness and reliability of the real-time line-crossing detector for detecting overtaking vehicles in a dynamic urban environment. The algorithm successfully detected and tracked the majority of objects, with a commendable accuracy rate of 80.95% using line detection techniques. However, the precision decreased when detecting overtaking vehicles due to the camera's limited view of their sides, highlighting the need for a more comprehensive side vehicle dataset.

The application of i_region significantly improved the system's performance, achieving a high precision value of 83.88%. This high precision was crucial in preventing false alarms, thereby increasing safety and reliability in riding environments. The implementation of i_region also effectively reduced the number of false positives, as demonstrated in

(Fig.4.(c)). However, the recall value of 62.5% revealed some limitations in detecting all relevant objects, particularly in scenarios with many unmoving or passed vehicles.

Based on the findings of this study, several recommendations are proposed to enhance the effectiveness and reliability of the real-time line crossing detector for detecting overtaking vehicles. First, the dataset used for training the algorithm should be expanded to include a wider range of side profiles of vehicles to improve precision. Second, future research should consider using higher versions of YOLOv8, such as YOLOv8m, for potentially better results. Finally, integrating the system with IoT platforms using devices like Raspberry Pi is suggested to enable real-time alerts and monitoring capabilities. Implementing these recommendations will optimize the detector, improve traffic safety, and enhance the overall efficiency of traffic monitoring systems.

Acknowledgment

I extend my gratitude to Mohamad Yusuf, S.Kom., M.C.S, for his invaluable guidance and unwavering support as the supervisor of this second research project. His expertise, constructive feedback, and dedication significantly contributed to the successful completion of this study. Additionally, I express my deep appreciation to Universitas Mercu Buana for providing the essential facilities and lectures that enriched my academic journey. The conducive research environment and resources offered by the university played a crucial role in the development and execution of this research.

References

[1] S. Meldiyana, I. Fathina, R. Yuner, S. Rachmat, and M. Harika, “Line Crossing Detector System in Real-time Situational Awareness Using Spatial Sample Difference Consensus,” *JTT (Jurnal Teknologi Terapan)*, vol. 8, no. 1, 2022.

[2] H. Halim, I. Mustari, Mursalim, and Z. Saing, “Overtake Behavior Model of Motorcycle on Accident Risk,” *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 12556–12567, 2020.

[3] D. M. Septianingtyas, “Identification of Characteristics of Traffic Accident Victims in Sidoarjo in 2016,” *The Indonesian Journal of Public Health*, vol. 14, no. 1, pp. 50–59, Jul. 2019.

[4] Republic of Indonesia National Police Headquarters Traffic Corps, *Traffic Police in Figures for 2021*, Jakarta: National Police of the Republic of Indonesia Traffic Corps, 2021.

[5] I. Ahmed, M. Ahmad, A. Ahmad, and G. Jeon, “IoT-based crowd monitoring system: Using SSD with transfer learning,” *Computers and Electrical Engineering*, vol. 93, 2021, doi: 10.1016/j.compeleceng.2021.107226.

[6] E. Panja, Hendry, and C. Dewi, “YOLOv8 Analysis for Vehicle Classification Under Various Image Conditions,” *Scientific Journal of Informatics*, vol. 11, no. 1, Feb. 2024.

[7] S. Tedjojuwono, N. Widjaja, and A. Kurniawan, “Vehicle Identification Systems using Virtual Line Sensors and Speed Up Robust Features,” *IOP Conference Series: Materials Science and Engineering*, vol. 924, p. 012031, 2020, doi: 10.1088/1757-899X/924/1/012031.

[8] E. Panja, Hendry, and C. Dewi, “YOLOv8 Analysis for Vehicle Classification Under Various Image Conditions,” *Scientific Journal of Informatics*, vol. 11, no. 1, Feb. 2024.

[9] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-Time Flying Object Detection with YOLOv8,” *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2305.09972.

[10] J. Solawetz and Francesco, “What is YOLOv8? The Ultimate Guide,” *Roboflow Blog*, 2023. [Online]. Available: https://blog.roboflow.com/whats-new-in-yolov8/.

[11] S. Karakuş, M. Kaya, and S. A. Tuncer, “Real-Time Detection and Identification of Suspects in Forensic Imagery Using Advanced YOLOv8 Object Recognition Models,” *Traitement du Signal*, vol. 40, pp. 2029–2039, 2023, doi: 10.18280/ts.400521.