

Article

Not peer-reviewed version

---

# Parking Time Violation Tracking using Yolov8 and DeepSORT

---

[Nabin Sharma](#) <sup>\*</sup> , [Sushish Baral](#) , [May Phu Paing](#) , [Rathachai Chawuthai](#) <sup>\*</sup>

Posted Date: 11 May 2023

doi: [10.20944/preprints202305.0828.v1](https://doi.org/10.20944/preprints202305.0828.v1)

Keywords: DeepSORT; Object Detection; Parking Time Tracking; Parking Violation Detection; Vehicle Tracking, Yolov8



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Parking Time Violation Tracking using Yolov8 and DeepSORT

Nabin Sharma <sup>1</sup>, Sushish Baral <sup>1</sup>, May Phu Paing <sup>2</sup> and Rathachai Chawuthai <sup>1,\*</sup>

<sup>1</sup> Department of Robotics and AI, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

<sup>2</sup> Department of Biomedical Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

\* Correspondence:rathachai.ch@kmitl.ac.th

**Abstract:** In Thailand, parking time violation is a major problem, especially for mini-marts. At present the task of detecting parking time violation is mainly conducted manually using Closed-Circuit Television (CCTV). This method requires additional human labour to track incoming and outgoing vehicles. Therefore, low cost time violation tracking is needed. To the best of our knowledge, there has not been any research for parking violation detection and tracking conducted for parking time limits. This paper introduces a novel parking time violation detection algorithm using the Yolov8 and DeepSORT tracking algorithms to track vehicles in consecutive frames. The presented parking violation tracking algorithm can provide a guideline for research in parking time violation detection.

**Keywords:** DeepSORT; object detection; parking time tracking; parking violation detection; vehicle tracking; Yolov8

## 1. Introduction

The rapid increase in car registration in Thailand has led to parking problems in cities throughout Thailand[1]. In order to meet the need for parking spaces, shopping malls and mini marts are trying to effectively manage the limited number of parking spaces that are accessible to the public. In Thailand, mini marts such as Tesco Express, 7-11, and Mini Big C, face a substantial parking problem due to the limited availability of parking spaces. In a fast paced environment such as 7-11 with limited parking facilities, it is essential to track the parking time of each vehicle in the parking lot. The issue of parking violations has long attracted the attention of the public. Parking violation includes activities such as parking a vehicle in a restricted area, and time limit violations. Figure 1 shows an instance of implementation of a time restriction parking rule in a mini-mart in Thailand.



Figure 1. Parking time limit of 15 min for consumers.

Most shopping malls and mini marts use Closed Circuit Television(CCTV) in parking lots to observe parking time violations. Currently parking violations are manually checked by the concerned authorities. However, these methods are expensive and require high labour costs due to the constant monitoring required to track incoming and outgoing vehicles. Therefore parking violation detection solutions which can reduce manpower and significantly save costs, are in high demand. To address the parking space problem, various existing parking solutions have used IOT devices to provide drivers with real time parking details [2]. However, using sensors and hardware, while providing accuracy, also requires constant maintenance, making it a non-ideal solution for mini-marts. A low cost parking violation detection is still required.

The majority of currently available research focuses on assisting drivers to locate parking spots. Numerous models have been developed to assist drivers with nearby parking spaces in real time [3]. However, there are very few studies that have been conducted to help authorities to manage parking spots effectively. Very few studies have been conducted regarding parking violations and time restrictions.

Over the last few years, mini-marts in Thailand have adopted the method to allocate a parking time of 15 minutes in their stores, in order to keep up with parking demand. However, this method has not been effective as additional manpower is required to record the arrival time and departure time of each car.

Recently, considerable effort have been made to use CCTV cameras to identify illegal parking practices. Many research studies have employed a Gaussian Mixture Model based image segmentation approach to retrieve vehicle information [4,5]. Akhawaji *et al.* [4] further utilized a Kalman filter to eliminate false alarms and enhance efficient vehicle tracking. However, the effectiveness of this strategy might be affected if the lighting conditions in the operating area change. Apart from image processing algorithms, many deep learning algorithms have been deployed to detect parking violations. For instance, networks such as the Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO) have been implemented to identify parking violations [6] [7] [8].

In this paper, a novel real time parking time violation algorithm using Closed Circuit Cameras is proposed. A state of the art object detection algorithm, Yolov8 has been implemented to detect vehicles in a parking lot. Consequently, each detected vehicle is assigned an ID in the parking lot, using DeepSORT to check for parking time violations. Optimal parameters have been chosen for the algorithms to have greater advantage during different levels of illumination, weather conditions and short term occlusion. Since, this algorithm does not require prior information about Region Of Interest (ROI), this algorithm can be implemented in many parking lot scenarios. To the best of our knowledge, this is the first research paper implementing parking violation detection in time-series. In conclusion the major contributions of this paper are as follows:

- 1) The proposed algorithm achieves good vehicle detection in many weather settings with accurate vehicle tracking over a prolonged period of time.
- 2) Scenarios such as short term occlusion do not affect the ID of the vehicle, making the model robust, accurate and practical in different parking lot settings.
- 3) For the first time, parking violation detection based on time has been proposed.

As a limitation and privacy concern, mini-mart data has not been used for inference. Parking lot video data from King Mongkut's Institute of Technology Ladkrabang (KMITL) has been used to demonstrate the novel algorithm. The remaining sections of this paper are organised as follows. Section 2 reviews the background and literature related to object detection and object tracking. Section 3 describes the details of the proposed parking time violation process, including the experimental setup and experimental settings of parameters for the algorithms. Section 4 presents the results and discussion.

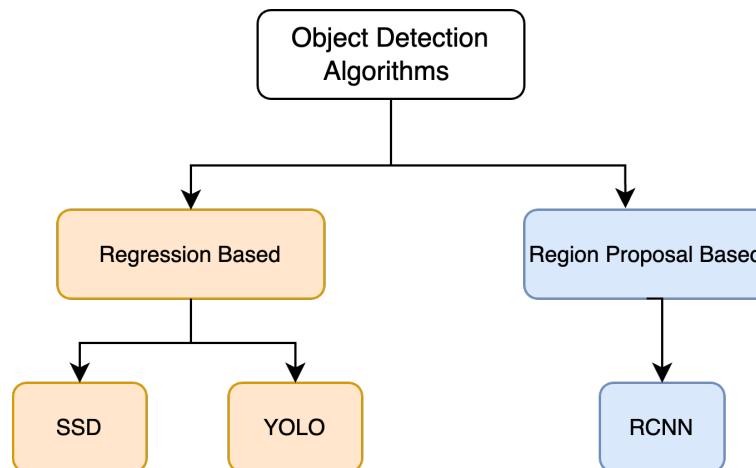
## 2. Related Work

Deep Learning applications have received significant traction due to their extensive research and development. Deep Learning based methods have been applied in many tasks, such as Classification [9], Object Detection [10], Object Tracking [11], and Healthcare [12] [13] [14]. The objective of parking time violation tracking is to identify parking time violation by vehicles especially in mini marts which have a limited parking time. The task involves implementation of object detection and object tracking. Therefore, the current literature on Object Detection and Object Tracking is discussed accordingly.

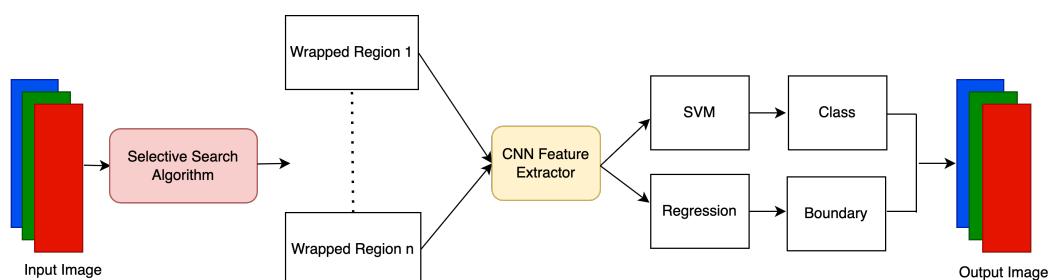
### 2.1. Object Detection

In its early years, the machine learning-based object detection pipeline consisted of two primary steps: ROI extraction and object classification [15]. Based on [Zhao et al.](#) [16], object detection models consists of three phases: selection of region of interest, extraction of features and finally classification of objects. One common ROI extraction technique uses sliding windows that move various scales across an image. Exhaustive sliding is, however, very challenging to apply in practice due to its great processing complexity.

Deep Learning algorithms are extensively implemented for object detection tasks. They can mainly be categorized into two types as shown in Figure 2. You Only Look Once (YOLO) [17] and Single Shot MultiBox Detector (SSD) [18] consider detection tasks as a regression problem and are one stage networks. On the other hand, algorithms such as Region-Based CNN (RCNN) [10] firstly locate the region of interest, which is further classified into classes. As depicted in Figure 3, The R-CNN model uses a selective search algorithm to determine the number of candidates for bounding box object regions and then features are fed into CNN that act as a feature extractor. Support Vector Machine (SVM) is used to classify if the object is present within that candidate region by utilizing the retrieved features. R-CNN can perform well in various object detection tasks, but training these models takes significant time and the speed of detection is limited [19].

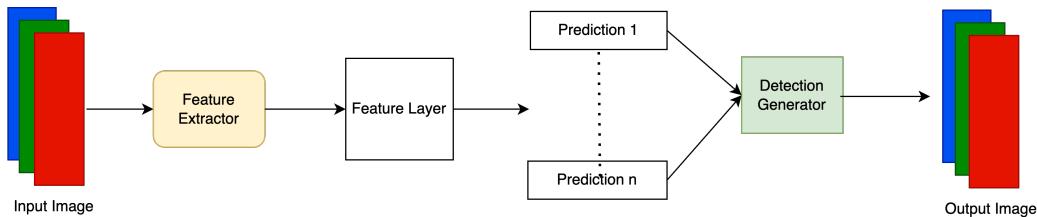


**Figure 2.** Object Detection Algorithms



**Figure 3.** Flowchart of Region-Based CNN

To improve the speed of training and inference, one-stage detectors such as SSD and YOLO have been introduced for object detection tasks. Figure 4 shows the architecture of SSD Models. In SSD, CNN based feature extractors are used. At the end of feature extractor, convolutional feature layers generate predictions at multiple scales. Since, SSD does not use region based proposals, SSD models have increased detection speed as opposed to two-stage detectors like R-CNN. In this research work, YOLO has been implemented for the object detection task. The details of the YOLO model are discussed accordingly in the Methodology section.



**Figure 4.** Flowchart of Single Shot MultiBox Detector

One of the first attempts to tackle the issue of parking lot monitoring using machine learning utilized color vector features on an SVM classifier to distinguish parking spaces inside a parking lot in 2002 [20]. Based on a 3D model of parking spaces, [Huang et al.](#) [21] proposed a day and night operating parking space detection system using a Bayesian hierarchical framework. Apart from deep learning models, many image processing techniques have been implemented to solve parking space detection problems. For instance, [Menéndez et al.](#) proposed temporal analysis of parking areas video frames to identify vacant spaces. This method includes the process of background subtraction using the Gaussian mixture to identify and track vehicles in the parking lot. In addition, a transience map was created to observe incoming and outgoing vehicles. [Xie et al.](#) proposed an optimized SSD to detect illegal vehicle parking from a video stream in a robust environment achieving 99% accuracy.

Recently, [Patel and Meduri](#) [23] presented an automatic parking space detection algorithm that comprised of two steps. The first step included vehicle detection with Faster R-CNN and Yolov4, while vehicle tracking was used to differentiate between moving and stationary vehicles. Based on [Patel and Meduri](#)'s research, this technique reduced the amount of human effort required by up to 90%. [Grbić and Koch](#) [24] proposed occupancy classification and a parking space detection algorithm, whereby parking spaces were determined as occupied or empty using a trained ResNet34 deep classifier, extensively evaluating this approach on publicly known parking datasets like PKLot [25] and CNRPark+EXT [26].

## 2.2. Object Tracking

Object tracking is a technique for detecting objects across frames by utilizing their spatial and temporal characteristics. In its simplest form, the method of obtaining the initial set of detections, giving them distinct IDs, and following them over frames is the essence of object tracking. Over the last decade, object tracking methods have gained popularity in the fields of deep learning and computer vision.

Single Object Tracking (SOT) and Multiple Object Tracking (MOT) are the two categories into which object tracking can be subdivided. [27]. Multiple Object Tracking algorithm's primary responsibility is to identify multiple objects in a frame, assign and preserve their identities, and follow the object's trajectory in an input frames. Object tracking has been used in many domains such as pedestrian tracking [28], vehicle tracking [29], and player tracking [30]. [Parico and Ahamed](#) [31] implemented Yolov4 for pear detection and a multiple object tracking algorithm, DeepSORT, for pear tracking and counting.

### 3. Proposed Parking Time Violation Algorithm

The proposed algorithm utilizes a state of the art object detection algorithm called Yolov8 [32] for detecting vehicles. For tracking vehicles, Deep SORT(Simple Online Real Tracking) was implemented [11]. In depth discussion of the algorithm is included in the subsection below.

#### 3.1. Vehicle Detection

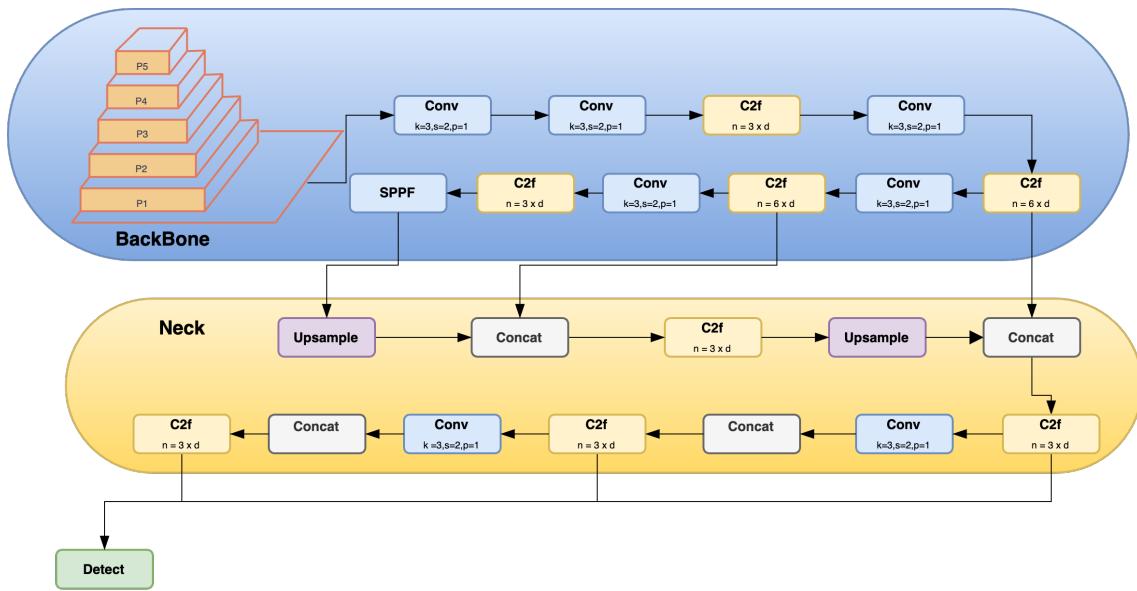
In 2015, Joseph Redmon and Ali Farhadi from the University of Washington created the cutting-edge object recognition algorithm called YOLO (You Only Look Once) [33]. YOLO outperforms many other object detection algorithms such as R-CNN and DPM [33]. YOLO sees the entire image during training and testing, in contrast to sliding window and region proposal-based approaches, implicitly capturing contextual information about classes as well as their appearance. The initial version of YOLO had the ability to quickly recognize objects in images, but it had trouble locating smaller objects precisely. Since, the problem domain does not contain very small images, there is no problem using YOLO in this study. YOLO divides input images to a grid, say  $M \times M$ . For instance, A grid cell is in charge of detecting an object if its center falls inside that grid cell. Each grid cell predicts the bounding box and offers a confidence score for the associated boxes. In YOLO, confidence is described as  $Pr(\text{Object}) * \text{IOU}$  where  $Pr(\text{Object})$  represents the probability of the presence of an object; IOU represents the Intersection over Union (IOU), the overlap area between inference and ground truth. Each grid cell generates 5 predictions ( $x, y, w, h$  and a confidence score). Additionally, each grid produces  $p$  conditional class probabilities, expressed as  $Pr(\text{Class} | \text{Object})$ . Equation 1 below demonstrates how to obtain class-specific confidence scores for each box during the test phase.

$$\text{Class}_i \times \text{IOU}_{\text{pred}}^{\text{truth}} = \text{Class}_i | \text{Object} \times \text{Object} \times \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

The final layers predict both the coordinates of their bounding boxes and their associated class probabilities. Then, the bounding boxes are normalized to fall between 0 and 1. All further layers increase non-linearity by using the leaky rectified linear activation function, as described in Equation 2, with an exception of the final layer, which employs a linear activation function.

$$f(x) = \begin{cases} x, & \text{if } x > 0. \\ 0.1x, & \text{otherwise.} \end{cases} \quad (2)$$

In 2016, YOLOv2 was released, improving on the original model by including batch normalization, anchor boxes, and dimension clusters [34]. Similarly, YOLOv3 was released in 2018 and improved the model's performance by employing a more efficient backbone network, incorporating a feature pyramid, and employing focal loss [35]. The initial models of YOLO have many successors including YOLOv4, YOLOv5, YOLOv6, and YOLOv7. In 2023, YOLOv8 was released by Ultralytics [32]. Figure 5 shows the detailed architecture of YOLOv8.



**Figure 5.** Yolov8 Architecture

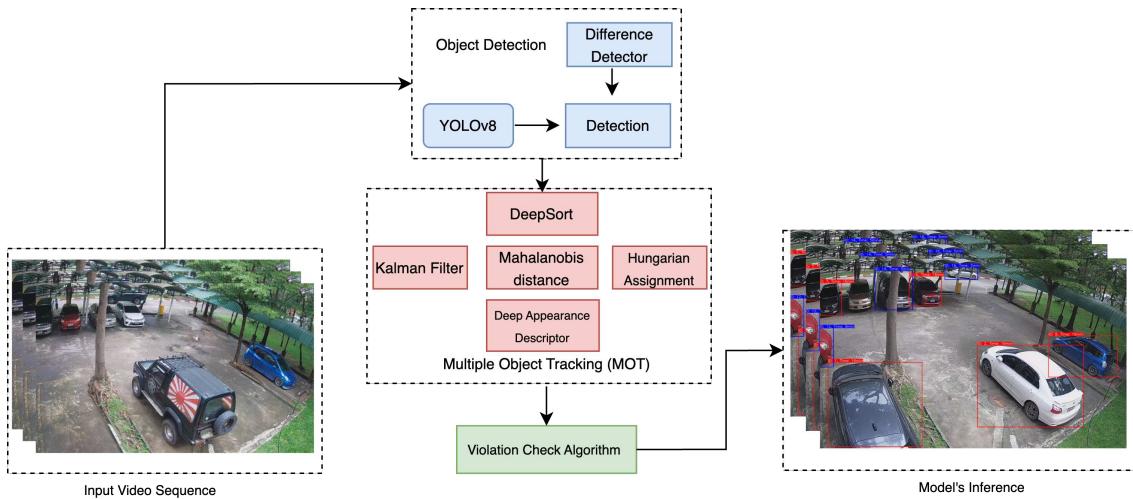
In the Backbone architecture, the C2f module based on Cross Stage Partial (CSP) is used in Yolov8, as opposed to the C3 Module used in Yolov5. The architecture of CSP enhances the learning capacity of CNN and decreases the computational effort of the model. The C2f module comprises of two Conv Module and n BottleNeck connected through Split and Concat. The remainder of the backbone park is the same as that of Yolov5. At the final layer of the backbone, the SPPF Module is used.

### 3.2. Movement Tracking

Once we are successfully able to detect vehicles in a parking lot, object tracking algorithms can be deployed to track the vehicle. The DeepSORT algorithm was implemented to track each vehicle throughout the frame. DeepSORT is an extension of SORT (Simple Online Realtime Tracking) [11]. DeepSORT uses appearance descriptors to minimize identity shifts, increasing the effectiveness of tracking. For problems involving the prediction of temporal or time series data, Kalman Filtering is the used algorithm.

### 3.3. Time Violation

After successful integration of vehicle detection followed by vehicle tracking that assigns a unique ID to each unique vehicle, a simple time violation algorithm was deployed in tracking every car in the frame. Every minute, the presence of the vehicle was checked. If the vehicle did not exist in the next 10 consecutive frames then the ID of the vehicle was discarded. On the other hand, if the ID was present in 15 consecutive frames the vehicle was determined to have violated the time restriction. Figure 6 shows the workflow of the proposed algorithm. In the figure, a blue label is associated with no violation, while a red label is associated with violation.



**Figure 6.** Workflow of our Proposed Work

### 3.4. Experimental Setup

This section discusses the experimental platform, providing details of the in-depth flow of the algorithm with the chosen parameters. The workstation used for the experiment was Ubuntu Linux. All experiments were conducted in a 3.6 GHz Intel Xeon Quad-Core processor with 8GB RAM and NVIDIA Quadro P4000 graphics card.

A Mean Average Precision (mAP) of 53.9 was achieved by YOLOv8x which was higher than all other YOLO versions. Therefore, Yolov8x was chosen for vehicle detection. A pretrained model was used as the class was already trained on the MS COCO dataset. The resolution of the Video Feed was 1080p (1920 x 1080) with a frame rate of 15fps. The dataset was considered balanced as it contained 50% of each class, violation and no violation. For Yolov8, the image-size parameter was set to 640. The model resizes the longest dimension to be 640 i.e. size 1920 becomes 640, while maintaining the aspect ratio. Therefore, the resized images were close to 640 x 360. Classes were filtered to be [2,5,7] i.e. car, bus, and truck, as the violation algorithm was checking for big vehicles. The confidence threshold was set to 0.5. The probability of the class occurring in the bounding box was evaluated using the confidence score. The maximum object detection parameter was set to 100.

For tracking vehicles, DeepSORT was used due to its superiority over its predecessors. The max-age parameter was set to 10. Max Age preserves the ID of a vehicle for a threshold number of frames before deleting the ID. The n\_init parameter was set to 2. The n\_init parameter refers to the number of objects detected before initializing the tracking of the object. max\_cosine\_distance was set to 0.3. Max Cosine Distance was the threshold to identify vehicle similarity by the DeepSORT algorithm. The trajectory parameter was set to False as the trajectory of the vehicle over time was not required.

### 3.5. Validation Criteria

Object detection algorithms like Yolov8 use Mean Average Precision (mAP) to validate object detection models. The objective of this work was to identify if a vehicle has violated a parking time restriction or not. Therefore, parking time violation tracking can be treated as a binary classification problem. Therefore, a confusion matrix can be used to evaluate the model. Firstly, it is essential to define TP, FP, FN and TN. Table 1 shows the detailed definitions. A False Negative can occur when the detection algorithm cannot detect the car in the frame due to various reasons such as the illumination condition, occlusion by trees, or reflection in a vehicle mirror. There was 1 instance of False Positive in this experiment. The Results and Discussion sections provide a detailed discussion of the observed results.

**Table 1.** Description validation criteria.

Definition	Description
True Positive (TP)	Vehicle parks for more than threshold time and the algorithm says Violated
True Negative (TN)	Vehicle parks for less than threshold time and the algorithm says Not Violated
False Positive (FP)	Vehicle leaves the parking space within threshold time and the algorithm says Violated
False Negative (FN)	Vehicle parks for more than threshold time and the algorithm says Not Violated

#### 4. Results and Discussion

From Table 2 it can be observed that the total number of actual violations is 13, out of which 11 were detected by the model. It can also be seen that 2 vehicles detected as "not violated" where in fact "violated" in terms of exceeding the time threshold. This happened as the object detection model was not able to detect the vehicle in Frame 1 but could detect the vehicle in Frame 5. Therefore, the counter of the vehicle began from Frame 5.

**Figure 7.** Observed Frame 1 to 30 of Violation Detection**Table 2.** Prediction summary shown in Confusion Matrix

Model Prediction	True Label	
	Positive	Negative
Positive	11	1
Negative	2	12

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

**Table 3.** Performance of our model

Precision	Recall	Accuracy
0.91	0.84	0.88

There was 1 instance of False Negative. This happened because the blue vehicle with ID 7 left the parking space before the violation time threshold, but entered the parking lot after 5 minutes. Based on the ID Deletion scheme, if a vehicle is not present for 10 frames, then the ID of that particular vehicle is discarded. This problem can be easily solved by discussion with concerned authorities before implementing the algorithm. The yellow car was not tracked in Frame 15, but started from frame 17. This is because the `n_init` parameter was set to 2. This means the tracking only begins if the same object has been detected in two consecutive frames. In addition, this parameter allows the user to provide a buffer of 2 minutes for incoming and outgoing vehicles. Based on the observations, metrics like Precision, Recall and Accuracy can be calculated. Precision refers to the ratio of the total correct parking violation predictions over the total number of violation predictions. Equation 3 shows the formula for Precision. A precision of 91% was achieved. Recall is also known as the True Positive Rate. Equation 4 shows the formula for Recall. The True Positive Rate was 84%. Accuracy refers to the Total number of correct parking violation predictions over the total number of predictions. The formula for accuracy is shown in Equation 5. An accuracy of 88% was achieved. Since, no research has focused on Time metrics for parking violation detection. Our proposed methodology using artificial intelligence can be valuable enhancement for sensor technologies.

## 5. Conclusion

Manual parking time violation tracking is a not ideal due to its requirements for manpower and high cost. In this research, a parking time violation tracking algorithm was successfully demonstrated; this can reduce additional manpower and labour cost. The algorithm can be implemented in many parking lot settings. The Yolov8 algorithm was used to detect vehicles in the parking lot, while DeepSORT was used to track the vehicle throughout the frame. Precision, recall and accuracy of 91%, 84% and 88% were achieved respectively. The performance of the detection model might be increased by custom training the model with vehicles. However, this is not necessary but may help the algorithm to be even more robust. Future studies can continue to explore synchronization of the algorithm with two cameras to detect parking time violations, allowing the algorithm to be scaled to situations where multiple CCTV cameras are used.

**Author Contributions:** Conceptualization, N.S,S.B and R.C; Methodology N.S,R.C,S.B; Validation,N.S.,M.P.P. and R.C.; Analysis, S.B.; Investigation, S.B.; Resources, N.S.; Data collection, R.C,N.S., S.B.; Draft Preparation, N.S.; Review and Editing, R.C,N.S., S.B.; Visualization, N.S.; Supervision, R.C, S.B., M.P.P.; Project Administration, R,C,S.B., M.P.P. All authors have read and agreed to the published version of the manuscript

**Funding:** This research did not receive any external funding.

**Institutional Review Board Statement:** Not Applicable

**Informed Consent Statement:** Informed consent was obtained from KMITL to use the dataset for research purpose.

**Data Availability Statement:** The data presented in the research paper is available on request as it is private parking lot data from KMITL.

**Acknowledgments:** We would like to acknowledge our sincere gratitude to the office of Information Technology of the School of Engineering, King Mongkut's Institute of Technology Ladkrabang for video data.

**Conflicts of Interest:** The authors declare no conflict of interest

## References

1. Flex, C. Thailand number of registered vehicles. *CEIC* **2023**.
2. Dinh, T.; Kim, Y. A Novel Location-Centric IoT-Cloud Based On-Street Car Parking Violation Management System in Smart Cities. *Sensors* **2016**, *16*. doi:10.3390/s16060810.
3. JOSEPH, J.; PATIL, R.; NARAHARI, S.; DIDAGI, Y.; Bapat, J.; Das, D. Wireless Sensor Network Based Smart Parking System. *Sensors & Transducers* **2014**, *162*, 5–10.
4. Akhawaji, R.; Sedky, M.; Soliman, A.H. Illegal Parking Detection Using Gaussian Mixture Model and Kalman Filter. 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), 2017, pp. 840–847. doi:10.1109/AICCSA.2017.212.
5. Sarker, M.M.K. Detection and recognition of illegally parked vehicles based on an adaptive gaussian mixture model and a seed fill algorithm. **2015**, *13*, 97–204. doi:10.6109/jicce.2015.13.3.197.
6. Chin Kit, N.; Cheong, S.; Yap, W.; Foo, Y.L. Outdoor Illegal Parking Detection System Using Convolutional Neural Network on Raspberry Pi. *International Journal of Engineering & Technology* **2018**, *7*, 17. doi:10.14419/ijet.v7i3.7.16197.
7. Xie, X.; Wang, C.; Chen, S.; Shi, G.; Zhao, Z. Real-Time Illegal Parking Detection System Based on Deep Learning. *CoRR* **2017**, *abs/1710.02546*, [\[1710.02546\]](#).
8. Tang, H.; Peng, A.; Zhang, D.; Liu, T.; Ouyang, J. SSD Real-Time Illegal Parking Detection Based on Contextual Information Transmission. *Computers, Materials & Continua* **2019**, *61*, 293–307. doi:10.32604/cmc.2020.06427.
9. Tamang, T.; Baral, S.; Paing, M.P. Classification of White Blood Cells: A Comprehensive Study Using Transfer Learning Based on Convolutional Neural Networks. *Diagnostics* **2022**, *12*. doi:10.3390/diagnostics12122903.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014, [\[arXiv:cs.CV/1311.2524\]](#).
11. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. *CoRR* **2017**, *abs/1703.07402*, [\[1703.07402\]](#).
12. Paing, M.P.; Pintavirooj, C. Adenoma Dysplasia Grading of Colorectal Polyps Using Fast Fourier Convolutional ResNet (FFC-ResNet). *IEEE Access* **2023**, *11*, 16644–16656. doi:10.1109/ACCESS.2023.3246730.
13. Paing, M.P.; Cho, O.S.; Cho, J.W. Histopathological Classification of Colorectal Polyps using Deep Learning. 2023 International Conference on Information Networking (ICOIN), 2023, pp. 472–477. doi:10.1109/ICOIN56518.2023.10048925.
14. Keakultanes, R.; Paing, M.P.; Pintavirooj, C. Automatic Cardiopulmonary Resuscitation System. 2022 14th Biomedical Engineering International Conference (BMEiCON), 2022, pp. 1–5. doi:10.1109/BMEiCON56653.2022.10012076.
15. Janai, J.; Güney, F.; Behl, A.; Geiger, A. Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art. *CoRR* **2017**, *abs/1704.05519*, [\[1704.05519\]](#).
16. Zhao, Z.Q.; Zheng, P.; tao Xu, S.; Wu, X. Object Detection with Deep Learning: A Review, 2019, [\[arXiv:cs.CV/1807.05511\]](#).
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection, 2016, [\[arXiv:cs.CV/1506.02640\]](#).
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016*; Springer International Publishing, 2016; pp. 21–37. doi:10.1007/978-3-319-46448-0\_2.
19. Mao, Q.C.; Sun, H.M.; Liu, Y.B.; Jia, R.S. Mini-YOLOv3: Real-Time Object Detector for Embedded Applications. *IEEE Access* **2019**, *7*, 133529–133538. doi:10.1109/ACCESS.2019.2941547.
20. Dan, N. Parking Management System and Methods, 2002. US Patent 10/066,215.
21. Huang, C.C.; Tai, Y.S.; Wang, S.J. Vacant Parking Space Detection Based on Plane-Based Bayesian Hierarchical Framework. *IEEE Transactions on Circuits and Systems for Video Technology* **2013**, *23*, 1598–1610. doi:10.1109/TCSVT.2013.2254961.
22. Menéndez, J.M.; Postigo, C.; Torres, J. Vacant parking area estimation through background subtraction and transience map analysis. *IET Intelligent Transport Systems* **2015**, *9*. doi:10.1049/iet-its.2014.0090.

23. Patel, R.; Meduri, P. Car detection based algorithm for automatic parking space detection. 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020, pp. 1418–1423.
24. Grbić, R.; Koch, B. Automatic vision-based parking slot detection and occupancy classification. *Expert Systems with Applications* **2023**, p. 120147.
25. Almeida, P.; Soares de Oliveira, L.; Jr, A.; Jr, E.; Koerich, A. PKLot - A Robust Dataset for Parking Lot Classification. *Expert Systems with Applications* **2015**, 42. doi:10.1016/j.eswa.2015.02.009.
26. Amato, G.; Carrara, F.; Falchi, F.; Gennaro, C.; Meghini, C.; Vairo, C. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications* **2017**, 72, 327–334.
27. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artificial intelligence* **2021**, 293, 103448.
28. Sun, Z.; Chen, J.; Chao, L.; Ruan, W.; Mukherjee, M. A Survey of Multiple Pedestrian Tracking Based on Tracking-by-Detection Framework. *IEEE Transactions on Circuits and Systems for Video Technology* **2021**, 31, 1819–1833. doi:10.1109/TCSVT.2020.3009717.
29. Hou, X.; Wang, Y.; Chau, L.P. Vehicle Tracking Using Deep SORT with Low Confidence Track Filtering. 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–6. doi:10.1109/AVSS.2019.8909903.
30. Buric, M.; Ivasic-Kos, M.; Pobar, M. Player Tracking in Sports Videos. 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2019, pp. 334–340. doi:10.1109/CloudCom.2019.00058.
31. Parico, A.I.B.; Ahamed, T. Real Time Pear Fruit Detection and Counting Using YOLOv4 Models and Deep SORT. *Sensors* **2021**, 21. doi:10.3390/s21144803.
32. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics, 2023.
33. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *CoRR* **2015**, *abs/1506.02640*, [\[1506.02640\]](https://arxiv.org/abs/1506.02640).
34. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *CoRR* **2016**, *abs/1612.08242*, [\[1612.08242\]](https://arxiv.org/abs/1612.08242).
35. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *CoRR* **2018**, *abs/1804.02767*, [\[1804.02767\]](https://arxiv.org/abs/1804.02767).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.