

Vehicle Surveillance System Using Yolov11 and Deepsort Algorithms

Olusola Theophilus Faboya

Department of Computing and Information Science, Bamidele Olumilua University of Education, Science and Technology, Ikere-Ekiti, Nigeria.

Abstract

The act of parking vehicles in unauthorised locations, especially in public places, has led to issues such as blocking pathways and access during emergencies. Parking on green lawns has negatively impacted the ecosystem. Although traditional deterrent methods and enhanced surveillance devices have been used, many are ineffective for managing dynamic situations. This paper presents an intelligent surveillance system that combines a cutting-edge machine learning algorithm, You Only Look Once (YOLOv11), with DeepSORT algorithm to address challenges of occlusion and identity switches for real-time vehicle recognition and monitoring. The system uses YOLO to capture images and videos of vehicles as input. It tracks their movements across frames with DeepSORT, identifying them based on predefined zone mappings of classified areas. Notifications are sent via email to designated personnel for prompt action. Testing in a controlled environment demonstrated high detection performance with a Mean Average Precision (mAP) of 99% at an Intersection over Union (IoU) threshold of 0.5. It also maintained consistent object identities across frames, providing reliable tracking at an average processing speed of 28 frames per second, with an ID F1 score of 93%. In conclusion, the combined strengths of YOLOv11 for precise detection and DeepSORT for persistent tracking have yielded a robust solution for real-time vehicle detection, indicating an improved vehicle tracking system in dynamic scenarios.

Keywords: Vehicle, Surveillance, YOLOv11, DeepSORT

1.0 Introduction

Unmonitored parking arrangements around public places, particularly within a university campus, have been a serious issue that requires attention. Such acts have resulted in blocking access paths, parking on green lawns, and general negative effects on the ecosystem. Parking vehicles on pedestrian pathways and access routes creates significant obstructions, hindering the free flow of both vehicles and pedestrians. From an environmental perspective, the act destroys green vegetation and ornamental shrubs (Lee & Power, 2013). It can also contribute to soil compaction and disruption of the natural ecosystem, contributing to ecological degradation. Over time, these destructions reduce the aesthetic appeal of the environment and could also lead to high costs of recovering and re-landscaping (Khachatryan et al., 2020).

Security personnel have played a crucial role in managing vehicle parking and providing surveillance in public places. Their presence has always served as a deterrent against unauthorised parking and enhanced safety. Physical barriers such as gates and bollards have also been used to restrict access to specific areas (Kovacich & Halibozek, 2013). These methods are resource-intensive and often insufficient for managing dynamic parking needs. The reliance on human enforcement can result in inconsistencies in monitoring effectiveness, particularly during peak hours or in high-traffic areas (Ratnam et al., 2024). Besides this, such operations can lead to frustration among users, especially when access is restricted during peak times or when barriers malfunction (Kirkpatrick, 2024; Alif, 2024; Godwin, 2022).

Apart from the conventional methods, there have been other advanced methods, including stationary cameras for monitoring parking areas (Alkudhayr, 2025). These methods provide a visual record of activities; they still require constant manual oversight, which can lead to delayed responses in emergencies. Consequently, the effectiveness of these devices can be compromised by poor lighting conditions or obstructions that block the camera's view. The need for manual review of recorded footage can hinder real-time decision-making (Ratnam et al., 2024).

However, machine learning techniques, particularly Support Vector Machines (SVM) and K-Nearest Neighbours (KNN), have been used for vehicle classification within monitoring solutions. SVMs, in particular, have been widely used in early parking management systems due to their effectiveness in binary classification tasks (Boateng et al., 2020). Its reliance on hand-crafted feature extraction limits its scalability to complex tasks such as multi-class vehicle detection or dynamic parking scenarios. The models also struggle with real-time processing, as they require significant computational resources for large-scale datasets or high-dimensional input, making them less practical for modern parking systems (Shdefat et al., 2024).

In recent times, some advanced technologies and solutions, including Convolutional Neural Network (CNN), Regional Convolutional Neural Network (R-CNN), You Only Look Once (YOLO), and Single-Shot Detector (SSD), have been employed to achieve improved performance. While the Region-based detectors, such as the R-CNN family, adopt a two-stage pipeline where a region proposal network first generates candidate bounding boxes before a classifier identifies objects, which yields high localization accuracy but with a slower processing speed (Shah & Tembhurne, 2023). The single-shot detectors such as YOLO and SSD eliminate the region proposal step by reframing object detection as a single regression problem by directly predicting bounding boxes and class probabilities from the full image in one evaluation and thereby enable real-time inference speeds. This architectural efficiency allows YOLO to maintain high frame rates while preserving competitive accuracy, making it particularly suitable for time-sensitive applications like surveillance systems (Shah & Tembhurne, 2023).

As the world is pushing for environmental sustainability. Goals 11 and 15 of the Sustainable Development Goals focus on reducing the environmental impact of urban areas through better waste management, protecting and restoring terrestrial ecosystems, halting biodiversity loss, and providing access to green public spaces while maintaining accuracy and accommodating large datasets.

Consequently, this paper introduces a novel surveillance system for monitoring vehicle parking within a university. It leverages the advancement in the capabilities of You Only Look Once (YOLO) for a robust vehicle detection process, integrated with the DeepSORT algorithm to provide an efficient monitoring system for improved performance in a dynamic environment. The proposed system seeks to provide enhanced detection accuracy and speed with robust tracking capabilities, particularly for handling occlusions and identity switches, for real-time vehicle recognition and monitoring in an environment.

YOLO is a state-of-the-art, unified deep learning model for computer vision tasks. A variant of CNN with object detector architecture, best known for its high speed and accuracy in object detection, instance segmentation, image classification, pose estimation, and oriented object detection (Magdy et al., 2025). The DeepSORT component supports the system in identifying vehicles based on predefined zone mappings of unauthorised areas. The combination of the two technologies is to provide a robust and environmentally friendly vehicle monitoring system that aligns with the Sustainable Development Goals.

The remaining part of the paper is structured as follows: Section 2 reviews related works, while Section 3 details the method used in the proposed system, Section 4 covers the implementation, Section 5 presents the results and discussion, and Section 6 discusses the conclusion and future work.

2.0 Related Work

Vehicle surveillance systems have experienced significant advancement, evolving from traditional methods to AI-driven solutions to meet the increasing complexities of modern traffic management. Early systems often depended on the Global Positioning System (GPS) and the Global System for Mobile Communication (GSM) for vehicle tracking and monitoring, offering a dependable, cost-effective tracking system.

The use of GPS, GSM and GRPS

Shinde (2015), proposed a vehicle anti-theft tracking system that utilizes GPS and location prediction techniques to track vehicles' location, even in situations where GPS is turned off. The study employs a time

series prediction algorithm based on historical location data to predict the vehicle's future locations. VasanthaAzhagu et al. (2014) developed a system using Global Navigation Satellite System (GNSS) for positioning and tracking vehicles to reduce parents' waiting time at the bus stop. The system is capable of sending information about the location of the buses to the parents through SMS. Kulkarni et al.(2017) used GPS and GSM to develop a user-friendly GUI anti-theft vehicle monitoring system in their study. The GUI displayed the vehicle positions on a map with their status, such as Ignition ON/OFF, Door Open/Close, etc. The system was able to track vehicles and store data about the vehicles' locations and where they travelled over a period of time.

Other studies provide insights into various aspects of vehicle tracking, using GSM, GPS, and GPRS and focused mainly on transportation, especially in the development of anti-theft mechanisms, real-time tracking, passenger waiting time and the development of intelligent tracking systems (Teja et al., 2023).

B. Vehicle Surveillance and Object Detection Algorithms

Recent researches have indicated that advancements in machine learning and deep learning have significantly improved the performance of object detection and tracking algorithms (Ratnam et al., 2024; Alif, 2024;Shen et al., 2025). Among the earlier approaches for object detection were those based on classic machine learning techniques, such as Support Vector Machine (SVM). Velazquez-Pupo et al. (2018) presented a high-performance vision-based system with a single static camera for traffic surveillance for moving vehicle detection with occlusion handling, tracking, counting, and One Class Support Vector Machine (OC-SVM) classification. The result of the extensive experiment showed that in eight real traffic videos with more than 4000 ground truth vehicles, the system ran in real time under an occlusion index of 0.312 and classified vehicles with a global detection rate or recall, precision, and F-measure of up to 98.190%, and an F-measure of up to 99.051% for midsize vehicles.

However, Kumar et al. (2023) observed that the advent of computer vision, particularly the rise of deep learning, has revolutionized the field. It has enabled the development of solutions capable of real-time vehicle detection, classification, and tracking with unprecedented accuracy and speed. Kit Ng et al.(2018) developed an outdoor low-cost illegal parking detection system based on convolutional neural networks using a Raspberry Pi for the detection of inappropriately parked vehicles. The system was able to deliver real-time alerts and achieved a precision rate of 1.00 and a recall rate of 0.94 across lighting scenarios. Cai et al (2018) created an enhanced CNN model to categorize vehicles in traffic monitoring systems, with potential for use in monitoring parking lots. Having the ability to classify vehicles is important for detecting unauthorized vehicles or making sure certain parking areas are used by the intended vehicle categories. Abu-alsaad (2023) proposed a smart parking system using CNN technology to identify empty and occupied parking spots via CCTV cameras. This system presented great precision and was seen as a more affordable option compared to sensor-based systems, which are infrastructure-intensive.

The current literature emphasizes CNNs' efficiency in vehicle detection assignments; yet, numerous systems encounter difficulties when performing in complex scenarios and under environmental variability, such as alterations in lighting or weather conditions, which may impact detection accuracy. Furthermore, some systems have been developed to operate in real-time; these systems require a substantial amount of computational resources, which restricts their ability to grow and be useful in bigger settings. To enhance performance against the limitations of existing machine learning and deep learning algorithms. The YOLO Framework was developed to handle real-time object detection (Miral et al., 2024). YOLO is a cutting-edge family of algorithms that was developed for real-time object detection applications. Since its release in 2016, the YOLO framework has undergone multiple revisions, all of which have improved upon the speed, accuracy, and processing efficiency of their predecessors.

Miral et al.(2024) explained that the YOLO framework is implemented in four phases for object detection. The images are segregated into squares in the first phase, while the object detection algorithms is applied to each of the squares in the second phase. YOLO adopts non-maximum suppression techniques in the third phase to nullify overlapping bounding boxes. In the last phase, YOLO tags the labels for the remaining bounding boxes. The metrics for evaluation of the object detection algorithm are presented in the form of intersection over union (IoU).

The Intersection over Union is defined as the ratio of the Area of Intersection and Area of Union, as represented as:

$$IoU = \text{Area of Intersection} / \text{Area of Union}$$

$$\text{Eq.....(1)}$$

The IoU threshold usually lies between 0 and 1. If the IoU is greater than the threshold, then it can be considered that the actual bounding box and predicted bounding box are quite similar. If the IoU is less than the threshold, it follows that the actual bounding box and the predicted bounding box are not close to each other.

The most recent version, YOLOv11, marks a substantial advancement in the field of object identification on both high-end GPUs and more advanced systems, making it adaptable for deployment in a variety of institutional settings. It introduces architectural enhancements designed to boost detection speed, accuracy, and resilience in complex environments, particularly for discerning smaller and occluded vehicles (Alif, 2024). These optimizations make YOLOv11 particularly proficient in handling the diverse shapes and sizes of vehicles encountered in real-world traffic scenarios by integrating multi-scale feature fusion for robust detection (You et al., 2024; Shen et al., 2025).

Object Tracking Algorithms

DeepSORT (Deep Simple Online Realtime Tracking) enhances the original SORT (Simple Online and Realtime Tracking) algorithm and incorporates deep learning-based appearance features for better tracking accuracy (Ratnam et al., 2024). It therefore provides more robust performance, especially in complex environments with occlusions, temporary disappearances, and similar-looking objects, by maintaining unique object identities over time. This DeepSORT's ability to maintain identity over time will significantly reduce identity switches and enhance the reliability of long-term vehicle tracking (Ratnam et al., 2024).

Hence, the integration of YOLOv11 with DeepSORT in the proposed system will specifically mitigate the system in identifying vehicles based on predefined zone mappings of unauthorised areas. The combined capabilities of YOLOv11 for precise detection and DeepSORT for persistent tracking offer a robust framework for real-time applications for detecting vehicles and maintaining vehicle identity through occlusions.

3.0. Methodology

The proposed system architecture is discussed in detail in Figure 1 below.

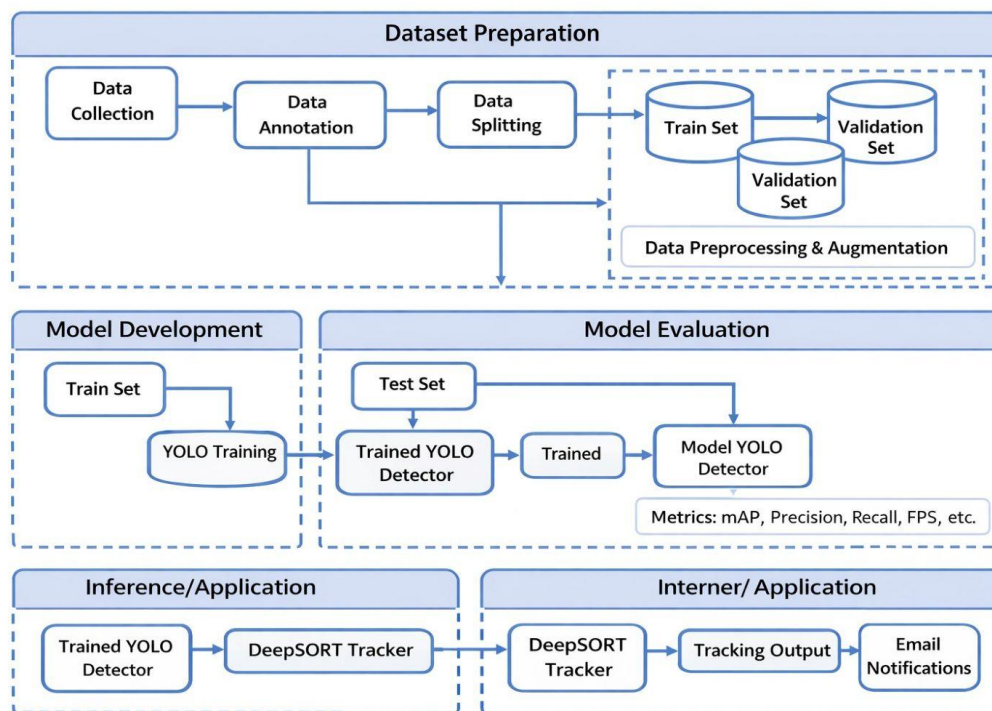


Figure 1: System Architecture: Step-by-step explanation

Dataset Preparation

Video footage of parking scenarios captured by CCTV, and other video footage from the controlled environment, simulating institutional parking scenarios of various vehicle types, lighting conditions, and parking situations were captured. The captured video feeds were converted into a sequence of images using Python OpenCV. To ensure sufficient diversity in vehicle poses and parking scenarios, the video feeds were processed at an average speed rate of 28 frames per second (FPS), ensuring near real-time analysis. Bounding boxes and classifications were generated within 10 seconds of capturing the video frames to create a dataset of still images that are YOLO ready .png files.

The extracted images from the previous stage were uploaded to Roboflow for annotation. Roboflow is a web-based annotated tool where bounding boxes were manually drawn around the detected vehicles. Each bounding box was labeled as either “Authorized Parking” or “Unauthorized Parking” based on predefined zone mappings. The datasets were augmented using techniques that include flipping, rotation, scaling, and brightness adjustment to enhance model generalization. The annotated images were stored internally in a normalized form by Roboflow, and automatically converted the annotations to the YOLO format, ensuring compatibility with the training pipeline.

Pseudocode for the YOLO-DeepSORT Detection and Tracking Pipeline

```
begin:
  //Data Collection:
    Collect images and video from the environment.

  //Data Annotation:
    Annotate the collected data with object class labels.

  //Data Preprocessing and Augmentation:
    Preprocess the data and split into train, validation, and test sets
    FOR each image in the dataset DO
      Standardize the image.
      Apply random crop/zoom (0% to 30%).
      Adjust Saturation of datasets ( $\pm 25\%$ ).
      Adjust brightness ( $\pm 25\%$ ).
    END data preprocessing and augmentation

  //YOLO-v11 Training
  //Initialize YOLO-v11 model with pre-trained COCO weights
  Begin Training:
    Train custom YOLO-v11 detector use YOLO-v11 architecture.
    Set training parameters:
      image size:  $640 \times 640$  pixels
      batch size: vary as 5, 10, 20
      epochs: vary as 16, 32, 64, 100, 300, 500
      weights: use pre-trained COCO dataset
    Train YOLO-v11 on the training set
    Validate the model on the validation set
    Save the best trained model
  END Training

  //YOLO-v11 Interference
  Run YOLO-v11 inference on test images:
    Obtain object detection results (bounding, boxes, classes, scores)
  END YOLO-v11 detection

  //DeepSORT Tracking
  Begin DeepSORT tracking
    FOR Each frame in video sequence DO
      Load YOLO-v11 detection results to DeepSORT
      Track objects using DeepSORT
    END FOR
    Output tracked objects
  END DeepSORT tracking

  //Send email notification:
    Compose a summary of detection and tracking results
```

Model Training

To improve the model’s detection accuracy, the annotated dataset was split into training, validation, and testing subsets as follows: 70% of the dataset for training; 20% for validation, and 10% for testing. This ensured that the model was evaluated on unseen data during testing. Then YOLOv11 was initialized with pretrained weights. Custom configurations, which include the number of output classes, input image size, and anchor box dimensions, were adjusted to align with the custom dataset. The model was trained using a GPU-enabled environment for faster processing, and Hyperparameters such as learning rate, batch size, and number of epochs were tuned to optimize performance.

Model Validation

The model’s performance was monitored on the validation set using metrics that include loss, precision, recall, and mAP. Early stopping was employed to prevent overfitting.

The following metrics were used to measure the performance of the system:

$$Precision = \frac{True\ Positive\ Cases}{Total\ Positive\ Predictions} \quad Eq. 2$$

Where: Precision is the proportion of correctly detected unauthorized parking instances.

$$Recall = \frac{True\ Positive\ Cases}{Total\ Cases} \quad Eq. 3$$

Where: Recall is the proportion of actual unauthorized parking instances correctly identified.

The mean Average Precision (mAP) evaluates overall detection accuracy across all classes. It is computed for the value recall over 0 to 1 as follows:

$$mAP = \int_0^1 P(R) dR \quad Eq. 4$$

Training Evaluation

The YOLOv11 model achieved a training mAP of 99% on the custom dataset, indicating its ability to detect vehicles accurately in various parking scenarios. Roboflow’s augmentation techniques improved model generalization, enabling the model to handle diverse parking scenarios.

System Integration

Incorporating DeepSORT into the system complemented the object detection process by establishing and maintaining a consistent identity for detected objects across sequential frames. The results obtained from training YOLOv11 on the custom dataset were integrated into DeepSORT for real-time vehicle tracking. DeepSORT tracked detected vehicles across frames, assigning unique IDs to monitor their movements. The detected and tracked vehicles were compared against predefined parking zones to flag unauthorized parking. This is vital for understanding trajectories and interactions in complex traffic scenarios, and it reduces identity switches and maintains consistent object identities significantly.

The Schematic diagram overview of the system with the YOLO architecture Model Training:

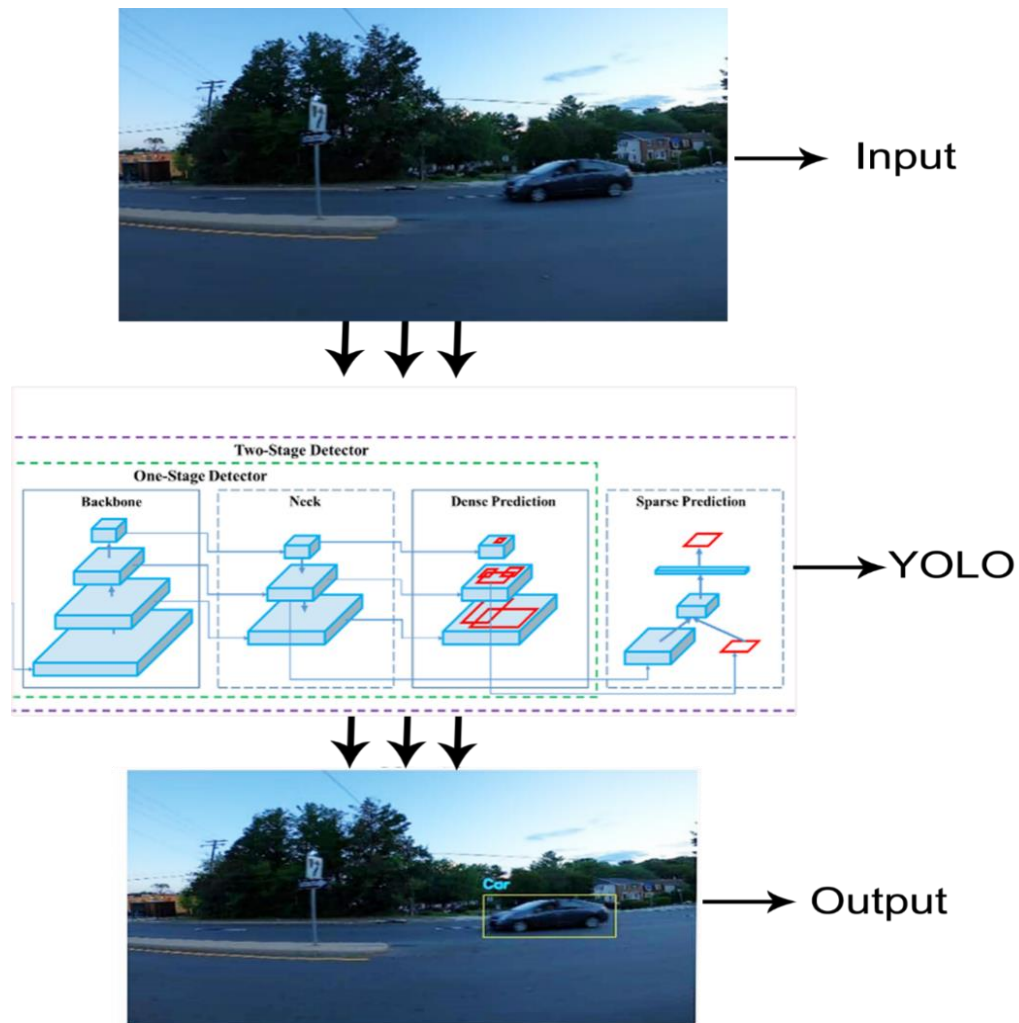


Figure 2: Schematic diagram overview of the system with YOLO architecture

How the System Works

Cameras installed in the marked places captured continuous video streams of cars in the designated areas, such as lawns, green spaces, or reserved areas marked as unauthorized parking zones.

The YOLO system consisted of components that include input layer, backbone, neck, dense prediction, and sparse prediction to enhance object detection capability.

- (a) The input layer stored the image fed into the system. The YOLO resized the input image to a set size (e.g., 640x640 pixels) to ensure uniformity of data samples.
- (b) The backbone extracted relevant features from an input image. YOLOv11 has integrated E-ELAN (Efficient Layer Aggregation Network), which improves feature extraction and network performance. For this work, E-ELAN enables the network to focus on crucial elements such as car outlines and forms that distinguish them from other objects in the scene.
- (c) The Neck is an intermediary layer that handles the Backbone's retrieved features. It is intended to improve the model's capacity to identify items at various scales, including sizes and shapes of vehicles parked in designated or undesigned places. This guarantees reliable identification performance in real-world parking settings.

- (d) The Dense Prediction component creates bounding boundaries and predicts class probabilities for observed items. It directly outputs the positions and sizes of items in the image, and their confidence scores. In this work, the Dense Prediction layer detected cars parked in areas marked as reserved or no-parking areas and produced bounding boxes around identified vehicles and confidence scores for classification (e.g., car, truck, bus).
- (e) To reduce noise in the datasets and improve precision, the sparse prediction employed the Non-Maximum Suppression (NMS) technique to remove overlapping bounding boxes. This is to ensure that only the most probable bounding boxes are considered for accurate object localization. This feature assisted in detecting closely parked vehicles that might result in overlapping bounding boxes and ensured that individual cars are appropriately identified and labelled without confusion. This is one of the reasons for employing YOLOv11 for this work.

4.0 Implementation

Validation Performance

On the validation set, the model achieved a precision of 100%, a recall of 100%, and an mAP of 99%, demonstrating strong generalization.

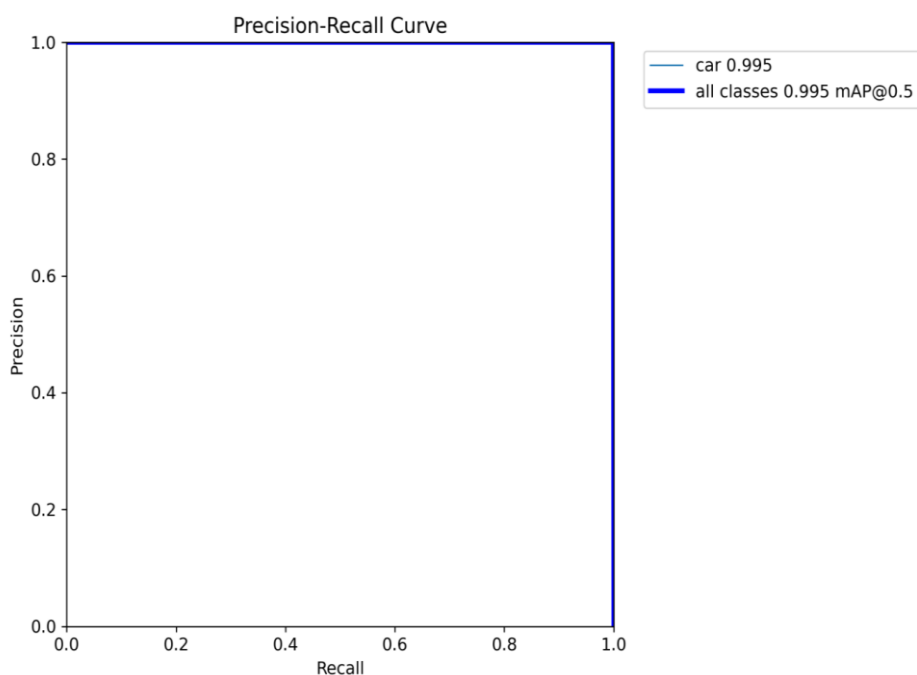


Figure 3: Precision-Recall Curve

Figure 3 shows the Precision-Recall curve, which is the trade-off between detection sensitivity and prediction reliability. The large area under the PR curve confirmed the model's effectiveness in handling class imbalance and demonstrated that the model has a precision across a wide range of recall values.

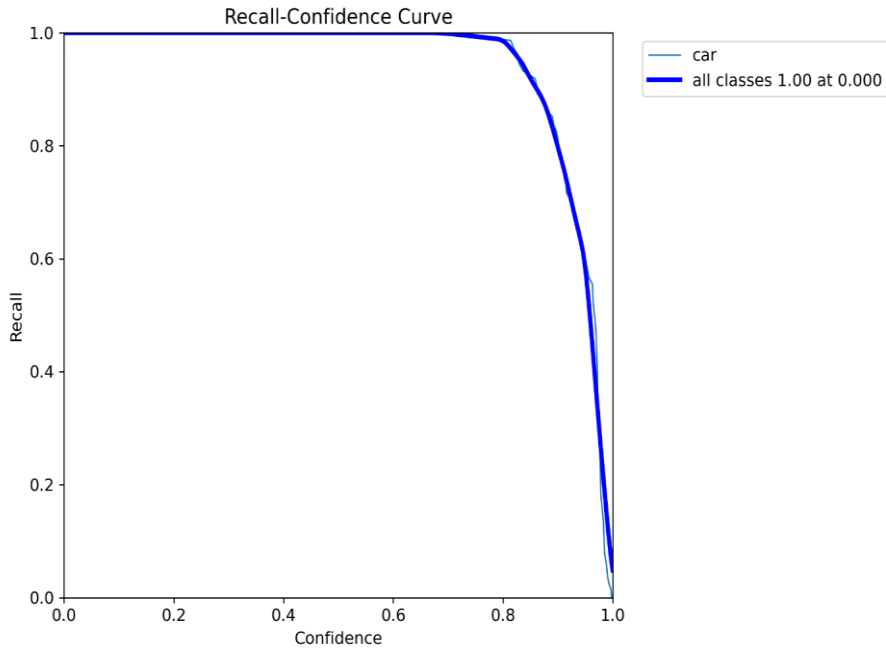


Figure 4: Recall-Confidence Curve

Figure 4 presents the Recall-Confidence curve, that showed that the model has a recall value of 0.995 at a confidence threshold of 0.8. The curve representation showed that the model assigned strong confidence scores to most true positive detections. The gradual decline in the recall at higher confidence thresholds suggested effective confidence calibration.

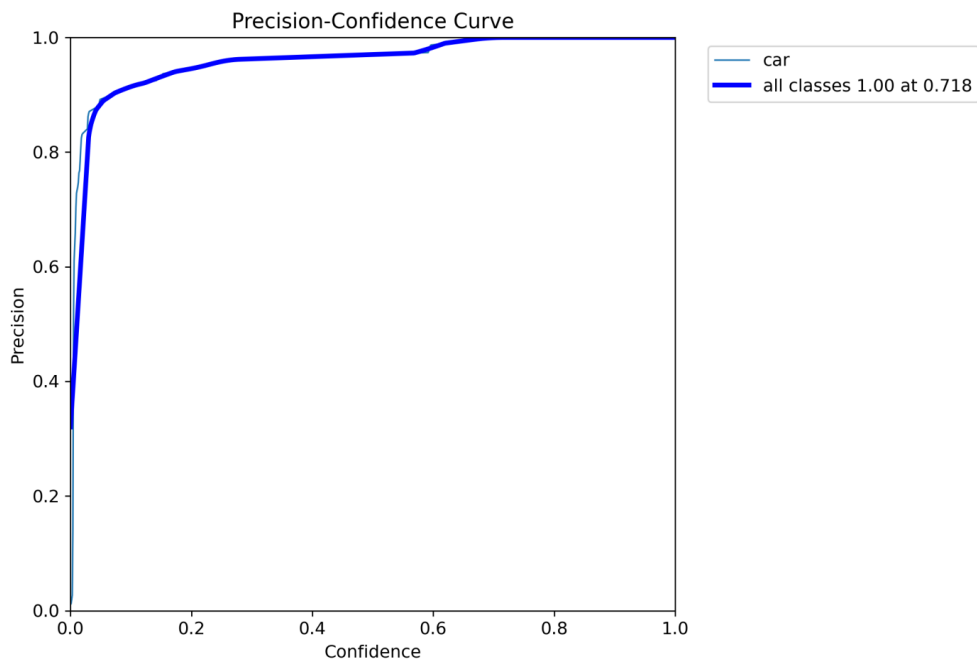


Figure 5: Precision-Confidence Curve

Figure 5 shows the Precision-Confidence curve, that indicated that the precision increases consistently with a higher confidence threshold, an indication that higher-confidence predictions are more reliable. This behavior supported the suitability of the model for vehicle detection applications under occlusion situations.

The summary of the performance presented in the three curves is that the Precision-Recall curves confirmed that the proposed model achieved a strong balance between sensitivity and prediction accuracy under varying thresholds. The Recall-Confidence and Precision-Confidence curves further demonstrated that the model's confidence scores are well calibrated; high-confidence predictions retain high precision while

recall degrades gradually. Together, these results validated the robustness and reliability of the model across different operating points.

Confusion Matrix Analysis

Figure 6 is the Confusion Matrix that provides a detailed breakdown of the system’s classification performance on the model’s ability to identify vehicles parked in an unauthorized place after a threshold of 10 seconds.

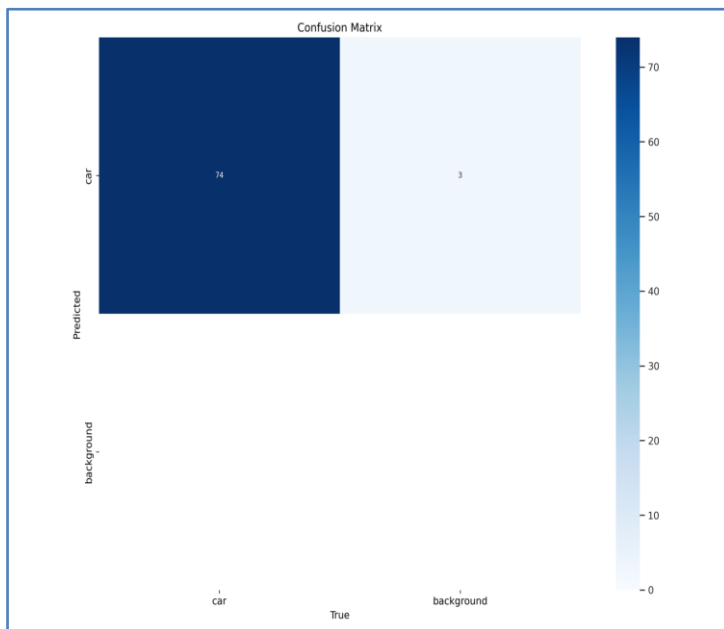


Figure 6: Confusion Matrix

The high values along the diagonal indicate strong classification performance, while the off-diagonal elements highlight areas of misclassification. The model demonstrated strong performance in both detecting unauthorized parking and correctly identifying authorized parking zones. The system achieved a detection accuracy of 95% on the test set, with false positives (3%) primarily due to vehicles near zone boundaries and false negatives (2%) caused by occlusions. Most errors occur between closely related classes, suggesting overlapping feature characteristics.

Confusion Matrix Metrics:

Parameter (Measure)	Result	Interpretation
Total Predicted		
True positive	93%	93% of the unauthorized parking cases were correctly detected.
False positive	3%	3% of authorized parking cases were misclassified as unauthorized, often due to vehicles near zone boundaries.
False negative	2%	2% of unauthorized parking instances were missed, primarily caused by occlusions or low-resolution frames.
True negative	95%	Instances where the system correctly identified authorized parking

		95% of the authorized parking instances were accurately identified.
--	--	---



Figure 7: Sample of a Car Detected in an unauthorized place

For the DeepSORT evaluation of tracking performance, the IDF1 Score is used to measure how well tracked IDs were maintained across frames. DeepSORT integration maintained ID consistency across frames, with an IDF1 score of 93%, ensuring reliable tracking, and the system processed video feeds at an average of 28 frames per second (FPS), allowing for near real-time performance.

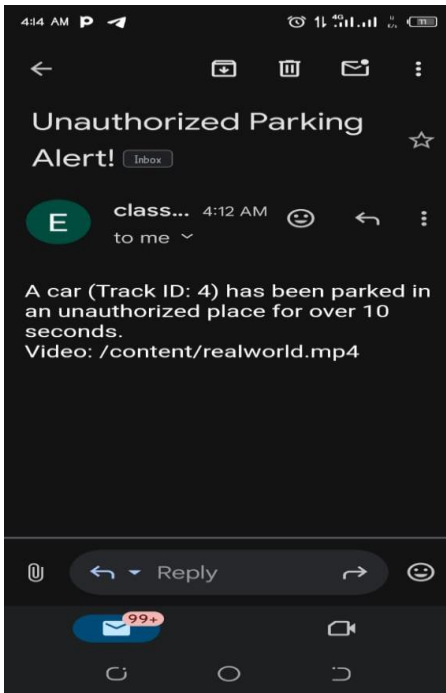


Figure 8: Email Infringement Message to Monitoring Personnel

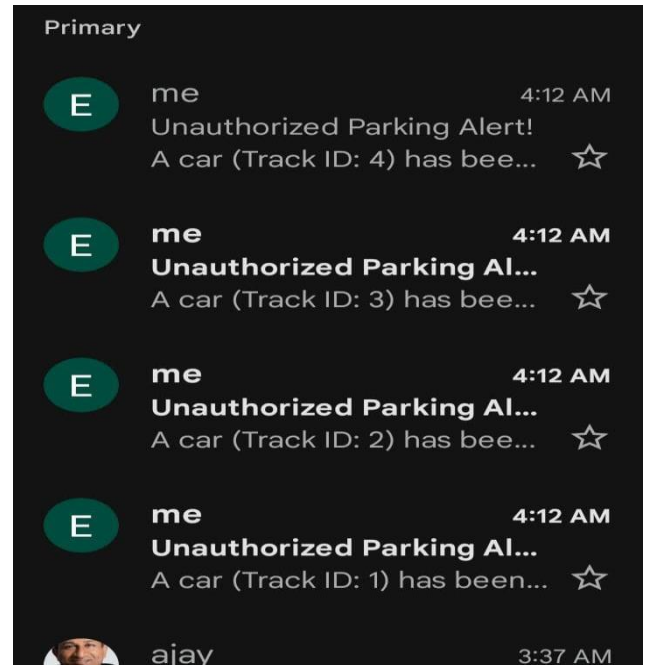


Figure 9; Email Notifications

Figures 8 and 9 are samples of infringement messages generated for a specified Gmail account of designated security personnel reporting a case of a car that has been parked in an unauthorized place for over 5 minutes.

5.0 Results and Discussion

Several experiments carried out showed that the proposed system has effectively mitigated issues such as occlusion and false classification by leveraging advanced features of YOLOv11 and DeepSORT. The technologies provide re-identification mechanisms and robust trajectory prediction, thereby ensuring continuous tracking even in challenging visual environments. The system demonstrated near real-time processing, making it suitable for live monitoring scenarios. It achieved an overall detection accuracy of 95%, correctly identifying vehicles and their compliance with parking regulations. The False Positives (3%) near boundaries were due to unauthorized zones being occasionally misclassified, and the False Negatives (2%) were due to some vehicles not detected as a result of occlusions or poor visibility in the captured footage. The overall results show that the model successfully identified a variety of vehicles with a high performance in daylight and well-lit environments, though its accuracy dropped slightly in low-light settings. The output evaluation indicates clear outputs, with bounding boxes around detected vehicles and labels specifying whether they were in authorized or unauthorized zones.

6.0 Conclusion

The integration of YOLOv11 with DeepSORT algorithms has demonstrated resilience across diverse operational conditions. The model proposed in this paper has effectively processed the captured video and identified vehicles with high accuracy and speed. The result of the evaluation process showed that the proposed system represents a significant advancement in vehicle surveillance, offering enhanced accuracy and efficiency crucial for intelligent vehicle detection systems. However, some areas requires further investigation in the future; these include conducting the model test in a real-world institutional environment rather than in a simulated environment and testing in actual parking areas to evaluate the system’s scalability and robustness in real-world conditions. The system development could also be automated with virtual maps to improve accuracy and efficiency. The alert system could further be expanded to include automatic walkie-talkie vibration whenever an infringement is detected. Investigation into whether the use of infrared cameras or additional lighting could enhance detection accuracy in low-light environments is required.

References

1. Abu-alsaad, H. A. (2023). CNN-Based Smart Parking System. *International Journal of Interactive*

- Mobile Technologies*, 17(11), 155–170.
2. Alif, M. A. R. (2024a). *YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems*. <http://arxiv.org/abs/2410.22898>
 3. Alif, M. A. R. (2024b). YOLOV11 FOR VEHICLE DETECTION: ADVANCEMENTS, PERFORMANCE, AND APPLICATIONS IN INTELLIGENT TRANSPORTATION SYSTEMS. *ArXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2410.22898>
 4. Alkhudhayr, H. (2025). Internet of things based parking slot detection and occupancy classification for smart city traffic management. *Engineering Applications of Artificial Intelligence*, 152, 110802. <https://doi.org/10.1016/j.engappai.2025.110802>
 5. Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. *Journal of Data Analysis and Information Processing*, 08(04), 341–357. <https://doi.org/10.4236/jdaip.2020.84020>
 6. Cai, Y., Liu, L., Ren, H., Ma, J., & Yang, J. (2018). Vehicle Type Classification from Video Using Deep Neural Networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(11), 3467–3478.
 7. Godwin, O. M. (2022). Global Positioning System Vehicle and Tracking System : a Review. *International Journal of Advanced Academic Research* |, 8(11), 138–145.
 8. Khachatryan, H., Rihn, A., Hansen, G., & Clem, T. (2020). Landscape Aesthetics and Maintenance Perceptions: Assessing the Relationship between Homeowners' Visual Attention and Landscape Care Knowledge. *Land Use Policy*, 95, 104645. <https://doi.org/10.1016/j.landusepol.2020.104645>
 9. Kirkpatrick, S. (2024). Expand Your Parking Paradigm to Meet Diverse Needs. <https://www1.appa.org/FacilitiesManager/article.cfm?ItemNumber=672&parentid=264>.
 10. Kit Ng, C., Nyeane Cheong, S., Wen-Jiun Yap, W., & Loo Foo, Y. (2018). Outdoor Illegal Parking Detection System Using Convolutional Neural Network on Raspberry Pi. *International Journal of Engineering & Technology*, 7(3.7), 17. <https://doi.org/10.14419/ijet.v7i3.7.16197>
 11. Kovacich, G. L., & Halibozek, E. P. (2013). Physical Security. *Effective Physical Security*, 339–353. <https://doi.org/10.1016/B978-0-12-415892-4.00021-3>
 12. Kulkarni, R., Hojage, A., Kakade, B., Pawar, P., & Khopade, R. (2017). Real Time Vehicle Tracking and Traffic Analysis. *Journal of Engineering, Technology And Innovative Research*, 2(2), 24–32.
 13. Kumar, S., Singh, S. K., Varshney, S., Singh, S., Kumar, P., Kim, B. G., & Ra, I. H. (2023). Fusion of Deep Sort and Yolov5 for Effective Vehicle Detection and Tracking Scheme in Real-Time Traffic Management Sustainable System. *Sustainability (Switzerland)*, 15(24). <https://doi.org/10.3390/su152416869>
 14. Lee, M. A., & Power, S. A. (2013). Direct and indirect effects of roads and road vehicles on the plant community composition of calcareous grasslands. *Environmental Pollution*, 106–113. <https://doi.org/10.1016/j.envpol.2013.01.018>
 15. Magdy, A., Moustafa, M. S., Ebied, H. M., & Tolba, M. F. (2025). Lightweight faster R-CNN for object detection in optical remote sensing images. *Scientific Reports*, 15(1), 1–14. <https://doi.org/10.1038/s41598-025-99242-y>
 16. Miral, D., Hiren, M., Pires, I. M., & Roy, S. (2024). Evaluating the Performance of the YOLO Object Detection Framework on COCO Dataset and Real-World Scenarios. *Procedia Computer Science*, 251, 157–163.
 17. Ratnam, V., Maddigunta, P., Annapureddy, L. S., Kolli, V. K., & Gurralla, S. (2024). Intelligent Traffic System: Yolov8 and Deepsort in Car Detection, Tracking and Counting. *International Journal for Research in Applied Science and Engineering Technology*, 12(3), 1838–1842. <https://doi.org/10.22214/ijraset.2024.59201>
 18. Shah, S., & Tembhurne, J. (2023). Object detection using convolutional neural networks and transformer-based models: a review. *Journal of Electrical Systems and Information Technology*, 10(1). <https://doi.org/10.1186/s43067-023-00123-z>
 19. Shdefat, A. Y., Mostafa, N., Al-Arnaout, Z., Kotb, Y., & Alabed, S. (2024). Optimizing HAR Systems: Comparative Analysis of Enhanced SVM and k-NN Classifiers. In *International Journal of Computational Intelligence Systems* (Vol. 17, Issue 1). Springer Netherlands.

<https://doi.org/10.1007/s44196-024-00554-0>

20. Shen, Y., Jian, R., & Zhao, J. (2025). Enhancing Traffic Surveillance Through YOLOv8 and ByteTrack: A Novel Approach to Real-Time Detection, Classification, and Tracking. In J. Shen, Y., Jian, R., Zhao (Ed.), *Proceedings of the 3rd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM2024)*. https://doi.org/https://doi.org/10.1007/978-981-96-2468-3_23
21. Shinde, P. A. (2015). *Advanced Vehicle Monitoring and Tracking System based on Raspberry Pi*.
22. Teja, P. S., Shiva, N., Reddy, S., & Pandugu, P. K. (2023). *Smart Vehicle Monitoring And Tracking System*. 01099, 1–7.
23. VasanthaAzhagu, A. ., Vijayalashmy, N., Yamuna, V., Rupavani, G., & Jeyalakshmy, G. (2014). GNSS Based Bus Monitoring And Sending SMS To The Passengers. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(1), 2502–2506.
24. Velazquez-Pupo, R., Sierra-Romero, A., Torres-Roman, D., Yuriy, V. S., Santiago-Paz, J., Gómez-Gutiérrez, D., Robles-Valdez, D., Reynoso, F. H., & Misael, R.-D. (2018). Vehicle Detection with Occlusion Handling, Tracking and OC-SVM Classification: A High Performance Vision-Based System. *Sensors*, 18(374), 1–23. <https://doi.org/10.3390/s18020374>
25. You, L., Chen, Y., Xiao, C., Sun, C., & Li, R. (2024). *Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack*.