

Intelligent Parking Management System for Smart Cities Using YOLOv8-Based Computer Vision

Basit Ali

Student, Department of Electrical Engineering, UET Peshawar enrbasitalai@gmail.com

Muhammad Hanif

Student, Department of Electrical Engineering, UET Peshawar
20pwele5695@uetpeshawar.edu.pk

Unmole Bakhtawar

Student, Department of Electrical Engineering, UET Peshawar
20pwele5690@uetpeshawar.edu.pk

Zar Afshan Iqbal

Student, Department of Electrical Engineering, UET Peshawar
zarafshanzafar918@gmail.com

Muhammad Farooq

Lecturer, Department of Electrical Engineering, UET Peshawar
muhammad.farooq@uetpeshawar.edu.pk

Bilal Ur Rehman

Lecturer, Department of Electrical Engineering, UET Peshawar.
bur@uetpeshawar.edu.pk

Kifayat Ullah

Lecturer, Department of Electrical Engineering, UET Peshawar.
kifayat.bangash@uetpeshawar.edu.pk

Muhammad Arshad

Lecturer, Department of Industrial Engineering, UET Peshawar.
marshad@uetpeshawar.edu.pk

Muhammad Kashif Khan

Lecturer, Department of Electrical Engineering, UET Peshawar.
kashifkhanmarwat@uetpeshawar.edu.pk

Wasim Habib

Lecturer, Department of Electrical Engineering, UET Peshawar.
wasimhabib@uetpeshawar.edu.pk

ABSTRACT

The advent of smart cities marks a significant leap towards creating urban environments that are more efficient, sustainable, and livable. A pivotal aspect in the evolution of smart cities is the integration of Artificial Intelligence (AI) with computer vision technologies. This research outlines the role of AI-driven computer vision in advancing the functionalities and efficiencies of smart cities. Smart cities use a multitude of data sources and technologies to optimize city management and improve the quality of life for their inhabitants. AI and computer vision play crucial roles in this ecosystem by enabling real-time data processing, analysis, and decision-making. These technologies facilitate numerous applications, including traffic management, public safety, environmental monitoring, and urban planning. One major challenge in modern cities is the search for unoccupied parking spaces. The increase in the number of vehicles, combined with limited parking spaces and low-speed parking searches, worsened traffic congestion, creating a time-consuming and polluted environment. The Intelligent Parking Management System for Smart Cities project aims to address the growing need for efficient parking solutions in urban areas. Traditional parking systems are plagued by inefficiencies, such as a lack of real-time information, difficulty in finding available parking spaces, and challenges in detecting illegal parking. In this proposal, we outline a comprehensive plan to create an intelligent parking management system using computer vision and YOLOv8, which will significantly enhance parking management in smart cities. The system successfully detects and displays Occupied and Available parking lots, as well as indicated right-parked and wrong-parked vehicles in real-time using a stable camera in the parking lot. We have employed a combination of cameras, image processing techniques, and real-time data analysis.

Keywords: Intelligent Parking, YOLOv8, Machine Learning, computer vision

Introduction

A city or municipality that employs information and communication technology (ICT) to enhance operational efficiency, disseminate information to the public, and improve the quality of government services and citizen welfare is known as a smart city [1]. The primary objective of a smart city is to optimize urban functions and foster economic development while also elevating citizens' quality of life using innovative technologies and data analysis. The actual value lies in how this technology is applied rather than merely the quantity of technology available. Smart city initiatives can also address climate change and air pollution issues, as well as manage waste and sanitation, through internet-connected garbage collection, bins, and fleet management systems [2].

Artificial intelligence deals with the development of systems that can perform tasks that generally require human intelligence, such as making decisions, recognizing speech, and identifying patterns [3]. AI and Machine Learning are now ubiquitous, playing a crucial role in various industries and serving as the backbone for newly established ones. Due to its importance and vast applications, it is used in building smart cities.

Computer vision is an area of artificial intelligence that equips computers to

comprehend and make sense of the visual environment [4]. By utilizing digital images from cameras, videos, and deep learning algorithms, computers can effectively recognize and categorize objects, responding to what they "observe." Major tasks within computer vision encompass image classification, object detection, image segmentation, and facial recognition. This technology finds extensive applications in fields such as self-driving cars, healthcare imaging, surveillance systems, and augmented reality [5].

YOLO, which stands for You Only Look Once, was introduced in 2015 with a pioneering research paper named "**You Only Look Once: Unified, Real-Time Object Detection**." YOLO marked a significant improvement in real-time object identification, presenting a cohesive framework that transformed the realm of computer vision [6]. The Evolution of YOLO is shown in Table 1.

YOLO Version	Year	Main Advancements
YOLO v1	2015	2015 Introduced real-time object detection using a grid-based approach.
YOLO v2	2016	Integrated anchor boxes, feature pyramid networks, and multi-scale predictions. .
YOLO v3	2018	Boosted both accuracy and speed by incorporating Darknet-53 and various detection scales.
YOLO v8	2021	Featured leading-edge enhancements in real-time object detection with improved accuracy and speed.

Table 0Error! No text of specified style in document. **The Evolution of YOLO Table**

Related Works

Smart Parking Systems Using IoT and Sensors:

This article focuses on developing smart parking systems and their relationship with the Internet of Things (IoT) and sensors. Most initial SP applications depend on IoT elements and sensors, including ultrasonic, infrared, and RFID, for sensing existing vehicles. Such systems often send information to a central hub where decisions are made or data are analyzed. These systems are helpful for occupancy detection; however, they pose problems of infrastructure and maintenance. They also underperform, particularly in complex scenarios where multiple vehicles are present or where environmental factors may impact the sensors. Your work expands this by incorporating computer vision, which may provide finer detections and would not rely on physical sensors [7].

Object Detection and Segmentation in Parking Systems:

In parking systems, object detection algorithms such as YOLO and SSD have been used in the detection of vehicles. However, such techniques usually provide bounding boxes around objects, which can be less accurate in complex surroundings. Although being fast, bounding box-based methods are ambiguous as far as occlusions and overlapping objects are concerned, which are typical for parking lots. This is a limitation of Faster R-CNN. In contrast, Mask R-CNN performs better in this regard by segmenting pixels to distinguish between different vehicles, even when they are

crowded. This ability to segment vehicles is critical in situations that involve multi-storey car parking, where some parts of one vehicle may be hidden by another [8].

Optimized Multi-Stage License Plate Recognition in Smart Parking Systems Using a YOLOv8 and CNN Framework:

This method integrates YOLOv5 for detecting license plates, YOLOv8 for recognizing characters, and a novel convolutional neural network design to enhance character recognition. We demonstrate that our method surpasses the single-stage model, achieving an overall accuracy of 96.1% compared to 83.9% for the single-stage model. Additionally, the approach is incorporated into an online dashboard that allows for real-time tracking and statistical evaluation of parking space usage and vehicle movements, maintaining efficient response times [9].

Enabling smart parking for smart cities using Internet of Things (IoT) and machine learning:

Our suggested solution aims to address this pressing issue, which is broken down into two parts: the first part includes a camera positioned at the entrance of the parking facility that employs the Automatic Number Plate Recognition (ANPR) technology to detect the license plate. The camera scans the license plate and applies optical character recognition (OCR) to obtain the plate number. Following this, the retrieved information is compared with records from an existing database to facilitate a seamless entry verification process. This method eliminates the need for security staff to manually verify vehicle identities or check for stickers, thereby enhancing the efficiency of access procedures. The second part entails the installation of a camera within the parking area that identifies whether parking spaces are available or occupied, storing the information collected by the camera in a central database, which allows for real-time updates of the closest available parking spots on digital displays at the entrances, significantly reducing the time and effort spent searching for parking spaces [10].

Smart Parking Management System for Cars Using IoT:

Smart Parking utilizes Ultrasonic sensors, an Arduino Uno, an ESP8266-01 Wi-Fi Module, and a Cloud server. The IoT-driven parking platform allows us to connect, assess, and automate the data collected from devices, making smart parking a reality. This intelligent parking system facilitates real-time monitoring of vehicle occupancy and the management of available parking spaces, thereby decreasing environmental pollution [11].

Implementation Challenges and Solutions in Real-world Parking Systems

The problem of utilizing computer vision-based parking systems in real-life contexts includes lighting conditions, weather interference, and restrictions due to real-time processing on edge devices. Past studies have proposed methods such as preprocessing methods for lighting equalization, processing at the edge so that computations are done on-site, and the employment of models that are resilient to variations in lighting conditions. It is in this regard that your research using Mask R CNN contributes to this discourse by offering solutions to these challenges through

improving upon image segmentation and providing a framework that can likely tackle real-world variability better than basic models [12].

Proposed Work

This research aims to enhance the smart parking system by leveraging computer vision with YOLOv8. The proposed technique aims to support the process of recognizing and categorizing parking spaces, enabling the correct indication of vehicles parked incorrectly or correctly in real-time, thereby improving the efficiency and satisfaction levels of parking users.

Dataset and Experimental Setup

We have generated a dataset of approximately 5000 images from various angles, which is used to label and develop a custom model for our research. The methodology is structured as follows

Prototype Setup

Materials: A small prototype page with printed parking slots and several toy cars

Image Capturing:

Capture images of the parking slot when unoccupied, occupied, and with the total slots visible.

Place a toy car correctly in the center of the slot and take pictures from different angles, labeling them as 'Right parked'.

Slightly displace the toy car and take pictures from different angles, labeling them as 'wrongly parked'

Dataset Size

We have collected 5000 images.

Each image represents different angles and orientations relevant to research.

Data Labeling:

Manual Annotation:

Divide the dataset into four groups.

Each group

Categorize images based on key characteristics and angles.

Used the labeling scheme to allow effective analysis of the data

Used “Roboflow” for labeling the images.

The labeling of images are shown in Figure 01



Figure 01 Car labelling

Data Preprocessing:

Image Resizing:

- Resize all images to a uniform size (e.g., 416x416 or 640x640) to ensure consistency for model training.

Normalization:

- Normalize the pixel values of images to the range [0, 1] or [-1, 1] to facilitate faster and more stable training.

Data Augmentation:

For improving diversity and robustness of the dataset apply various augmentation techniques:

Rotation: Randomly rotate images to simulate different camera angles.

Translation: Slightly shift images horizontally and vertically.

Scaling: Scale images up and down.

Flipping: Horizontally flip images to increase variation.

Brightness Modification: Alter the brightness of images in a random manner.

Contrast Modification: Change the contrast of images at random.

Noise Introduction: Introduce random noise to images to mimic real-world condition.

Splitting the Dataset:

Split the dataset into training, validation, and test sets (e.g., 70% training, 20% validation, 10% test).

Bounding Box Adjustment:

Ensure bounding boxes are adjusted and normalized according to the resized image dimensions.

Result and Discussion

First, trained the model on a car dataset using the YOLOv8 small model. This initial phase of training focused exclusively on cars. The car dataset consisted of images captured under controlled conditions with varying angles, lighting, and positions to simulate realistic parking scenarios.

- **Dataset Details:** The dataset included a total of 1800 images, divided into training (1260 images), validation (360 images), and test sets (180 images). These images were manually annotated to label Car using the Roboflow tool.
- **Data Preprocessing:** Images were resized to 416x416 pixels and normalized to ensure consistency. Data augmentation techniques, including rotation, translation, scaling, flipping, brightness and contrast adjustment, and noise addition, were applied to enhance the dataset's diversity and robustness.

Model Configuration: The YOLOv8 small model was trained with a batch size of 16, over 20 epochs, using the Adam optimizer with a learning rate of 0.001. The

respective results are shown in Table 02.

Table 02 Car Data Set Training Results

Yolov8 algorithm used on Car dataset	Obtained result (%)
Accuracy	96
Precision	95
Recall	93
F1 score	93.5

Enhanced Parking Slot Detection:

After training the model, I proceeded to draw the coordinates of each parking slot within the prototype. The key steps involved:

Coordinate Mapping: Defining the boundaries of each parking slot by drawing coordinates on the prototype page.

Center Targeting: Implementing a mechanism to target the center of the detected car within each slot.

Occupancy Detection: When a car enters a slot, the system detects its presence by targeting its center within the defined coordinates of the slot. If the center falls within the boundaries of the slot, it is marked as occupied.

System Implementation:

Total Slots: The prototype includes a total of 12 parking slots.

Occupancy Status: The system continuously monitors the parking slots. When a car is detected in a slot, it updates the status to show that the slot is occupied.

Performance Evaluation:

Real-Time Detection: The model successfully detected car occupancy in real-time, updating the slot status accordingly.

Accuracy: The occupancy detection was accurate, with minimal false positives or false negatives, ensuring reliable status updates

Second Phase of Training: Wrong and Right Parked Dataset with YOLOv8 Nano Model

Building on the success of the first phase, the next step is to train the model using the labeled dataset that includes both correctly and incorrectly parked cars with the YOLOv8 nano model. The results are illustrated in the Figures below. Figure 02 presents a Precision-Recall Curve for the model aimed at classifying parked cars. This curve illustrates the balance between precision and recall for the classes "Right-Parked" and "Wrong-Parked." The nearly rectangular shape of the curves, where precision stays close to 1 as recall increases, indicates the model's high performance. The mAP (mean Average Precision) for both classes is notably high, with values of 0.982 for "Right-Parked" and 0.991 for "Wrong-Parked," leading to an overall mAP of 0.987 at an IoU threshold of 0.5, showcasing outstanding precision and recall for each class.

Meanwhile, Figure 03 depicts the F1-Confidence curve, which illustrates the F1 score (y-axis) about the confidence threshold (x-axis) for classifying "Right-Parked" and

"Wrong-Parked" vehicles. The highest F1 score across all classes reaches 0.97 at a confidence threshold of 0.421. The points out the ideal threshold for achieving a balance between precision and recall in the model's predictions. The resemblance in the shapes of the curves for both classes in Figures 02 and 03, respectively, suggests that the model demonstrates consistent performance across them.

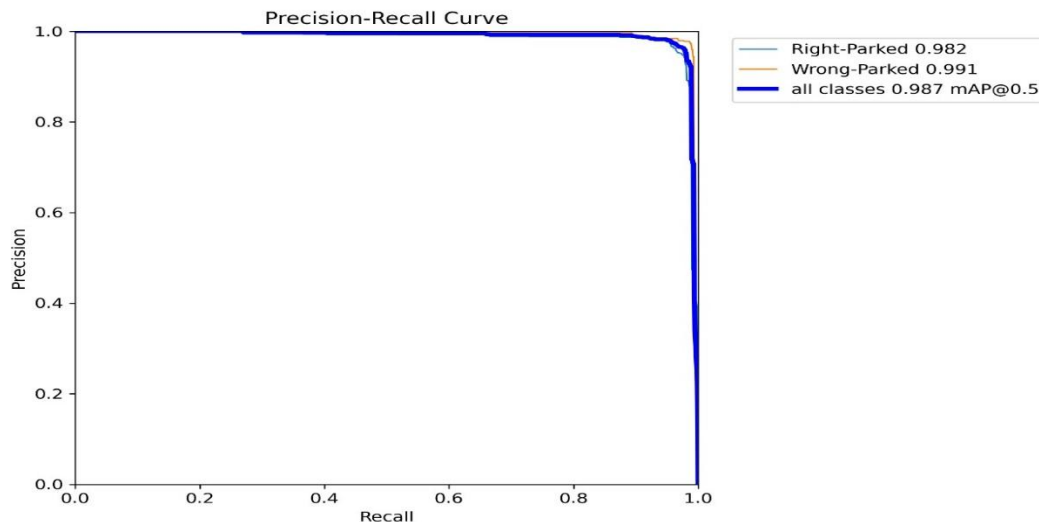


Figure 02 YOLO v8(Nano) precision recall curve.

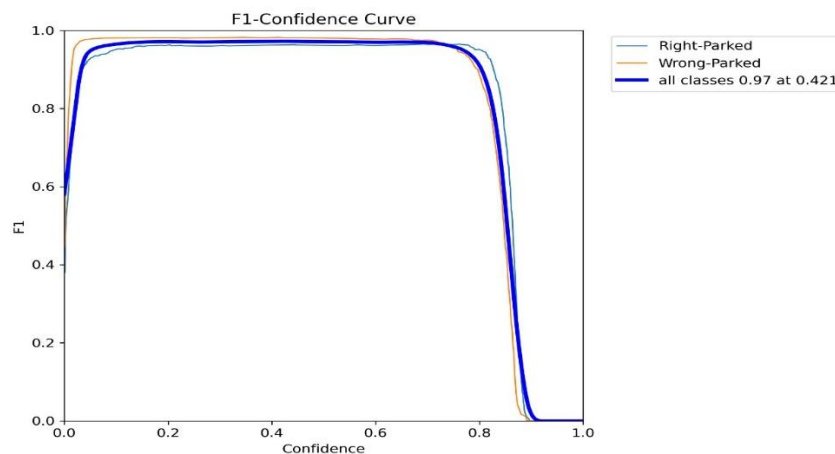


Figure 03 YOLO v8 (Nano) F-1 Confidence Curve.

Figure 04 shows a confusion matrix for a smart parking system using computer vision. The matrix compares the predicted classifications (rows) with the actual classifications (columns) for three categories: "Right-Parked," "Wrong-Parked," and "Background." High values on the diagonal (0.97 for Right-Parked, 0.99 for Wrong-Parked, and close to zero for Background) indicate accurate predictions. Misclassifications are represented by off-diagonal values, such as 0.74 where Right-Parked is incorrectly predicted as Background.

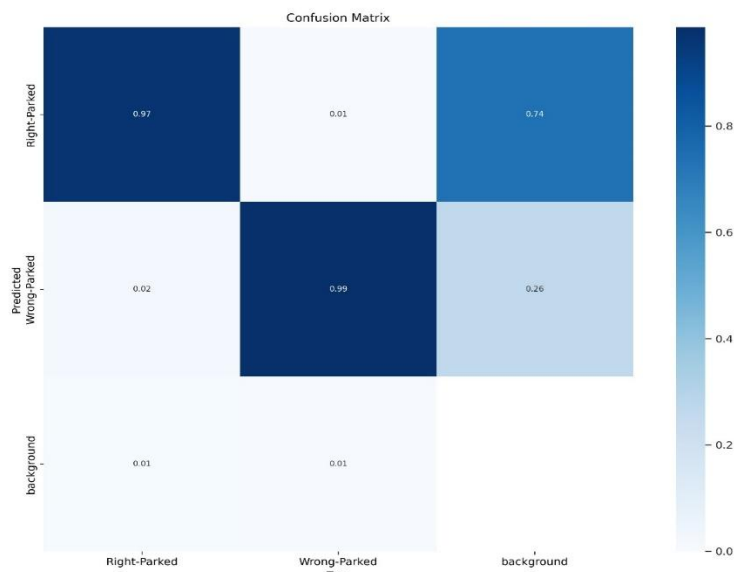


Figure 04 YOLO v8 (Nano) Confusion Matrix.

Figure 05 shows various training and validation metrics over epochs for YOLOv8 Nano model. All losses (box, objectness, and classification) decrease steadily, indicating that the model is learning effectively and generalizing well. Both precision and recall increase and stabilize at high values (~0.98 for precision and ~0.99 for recall), showing the model's high accuracy and ability to detect true positives. Overall, the model shows good performance with low losses and high precision, recall, and mAP values.

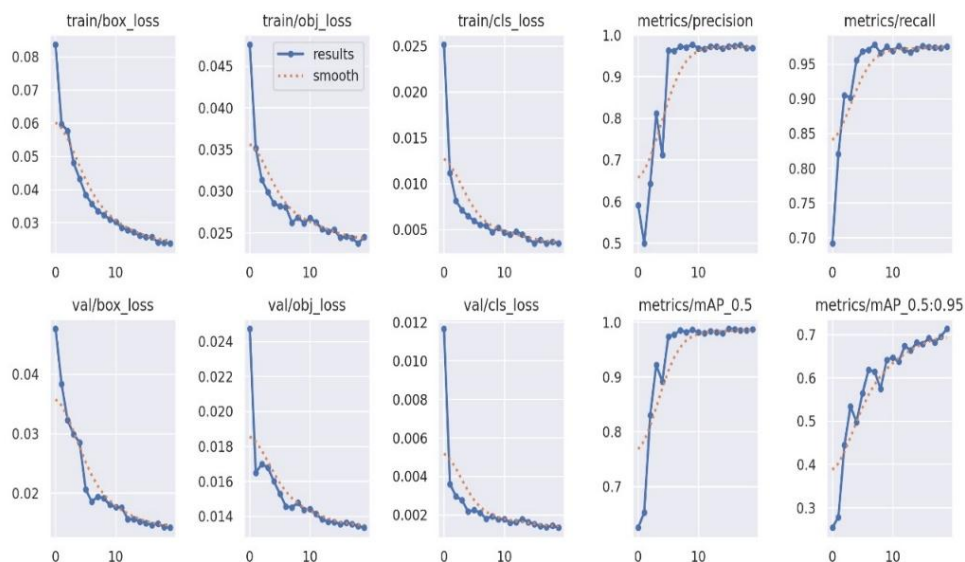


Figure 05 YOLO v8 (Nano) Losses Curves

The Figure 06 presents a Recall-Confidence Curve for a classification model differentiating between "Right-Parked" and "Wrong-Parked" vehicles. The x-axis represents the confidence score, while the y-axis indicates recall. The blue line shows the recall for all classes combined, achieving perfect recall (1.00) at a confidence threshold of 0.000. The light blue and orange lines correspond to the recall for the

"Right-Parked" and "Wrong-Parked" classes, respectively. The curves indicate high recall for both classes at lower confidence scores, with a sharp decline as the confidence threshold approaches 1.0.

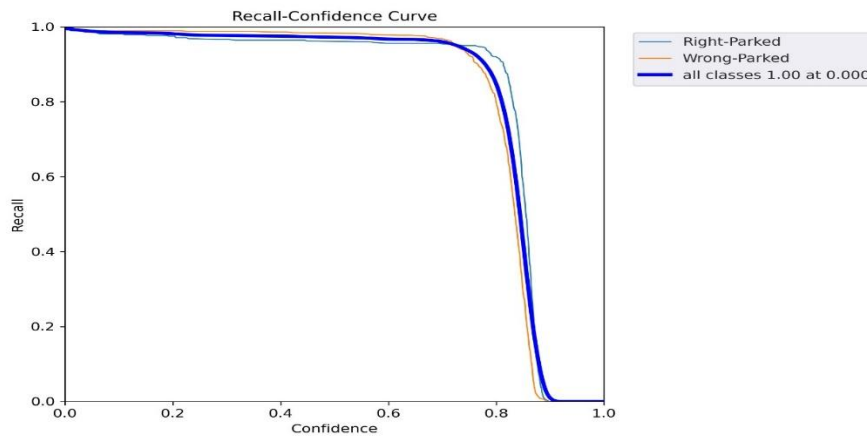


Figure 06 YOLO v8 (Nano) Recall - Confidence - Curve.

Figure 07 shows a Precision-Confidence Curve for a classification model distinguishing between "Right-Parked" and "Wrong-Parked" vehicles. The x-axis represents the confidence score, while the y-axis indicates precision. The blue line represents the performance of all classes combined, achieving perfect precision (1.00) at a confidence threshold of 0.893. The light blue and orange lines depict the precision for the "Right-Parked" and "Wrong-Parked" classes, respectively. The curves suggest high precision for both classes across a wide range of confidence scores

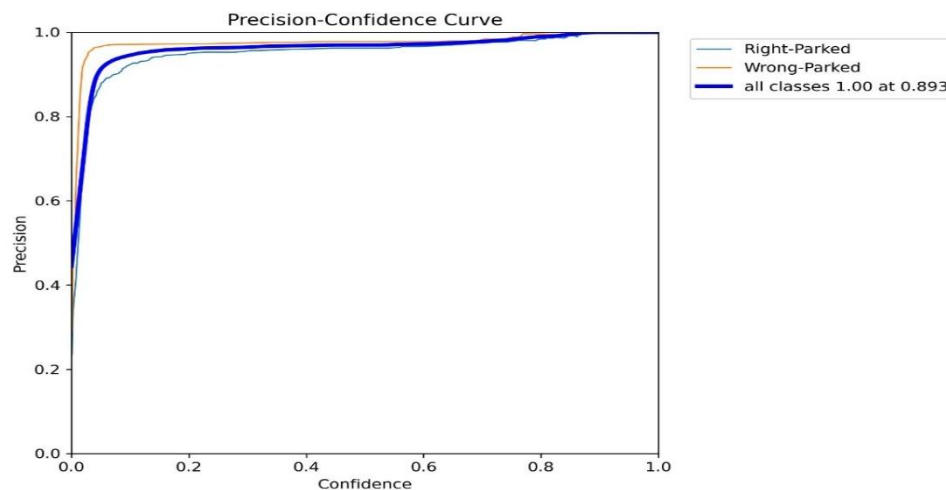


Figure 07 YOLO v8 (Nano) Precision- Confidence – Curve

Figure 08 shows the real time detection of YOLOv8 Nano model.

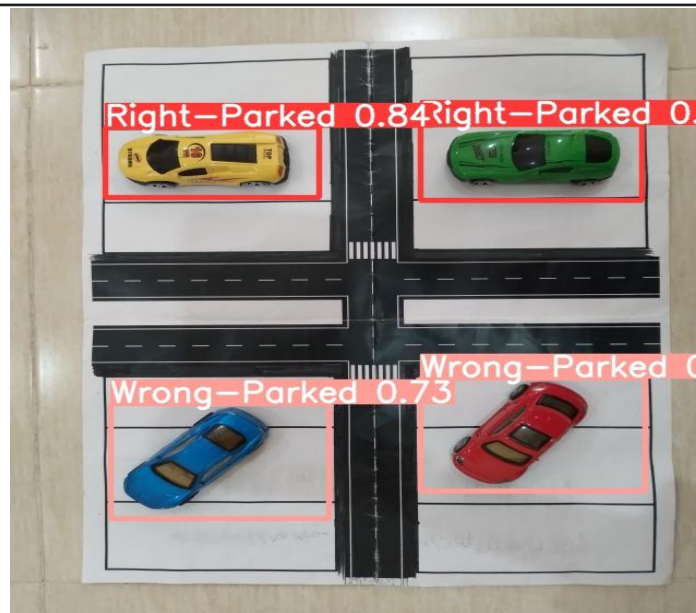


Figure 08 YOLO v8 (Nano) OUTPUT.

The following Table 03 shows the precision, Recall and F1- score of YOLOv8 Nano model.

Table 03 YOLO v8 (Nano) Results.

Class	Precision (%)	Recall (%)	F1 - score (%)
Right parked	97	56	77
Wrong parked	96	77	86
Background	20	92	33

Training: Wrong and Right Parked Dataset with YOLOv8 Large Model to improve Accuracy

Building on the success of the initial phase on **Nano** model, I then trained the model on the labeled dataset of correctly and wrongly parked cars using the YOLOv8 **large** model, for the sake of improvement. The accuracy of the previous Nano model was 79%. The results of the YOLOv8 large model are shown below figures. Figure 9 displays a confusion matrix for a parking classification system with three categories: "Right-Parked," "Wrong-Parked," and "Background." The matrix shows that 460 right-parked cars were correctly identified, with 8 mislabeled as wrong-parked and 18 as background. For wrong-parked cars, 565 were accurately detected, with 11 misclassified as right-parked and 15 as background. The background category had 7 right-parked and 4 wrong-parked cars incorrectly classified.

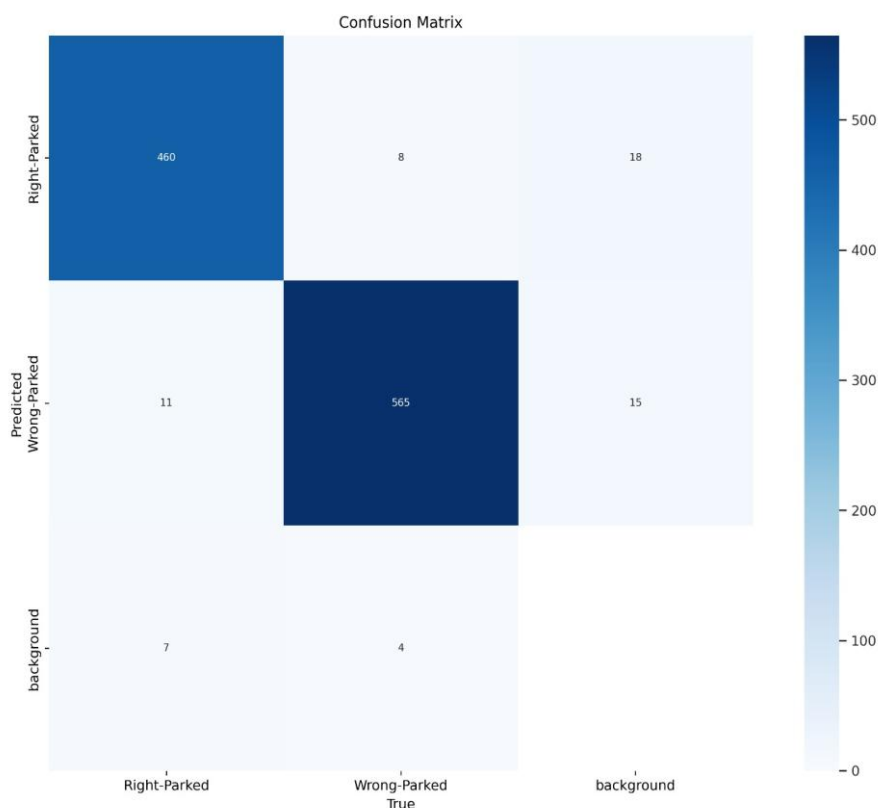


Figure 09 YOLO v8 (Large) Confusion Matrix

Figure 10 displays a precision-confidence curve for a model distinguishing between "Right-Parked" and "Wrong-Parked" vehicles. The x-axis represents confidence levels, while the y-axis shows precision. The blue line represents the overall performance, with an average precision of 1.00 at a confidence level of 0.826. The orange line indicates the performance for "Wrong-Parked," and the light blue line for "Right-Parked." The curves demonstrate that the model maintains high precision as confidence increases, peaking near a confidence level of 1.0.

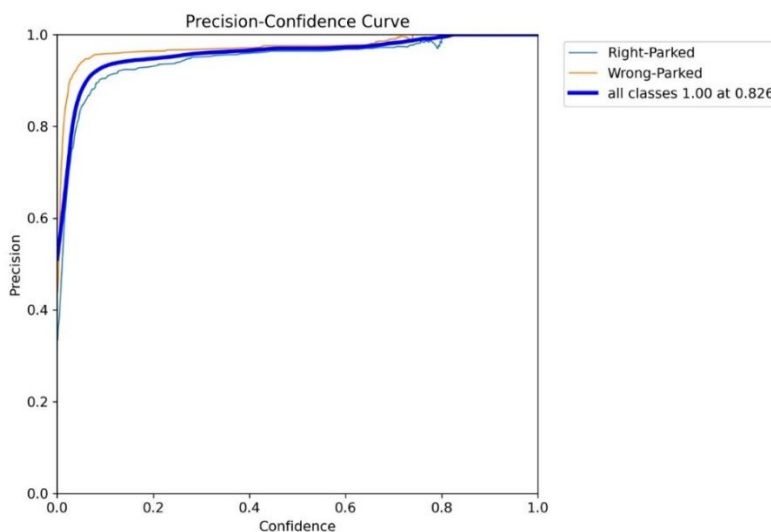


Figure 10 YOLO v8 (Large) Recall - Confidence - Curve.

Figure 11 below shows an F1-confidence curve for a model that classifies "Right-

Parked" and "Wrong-Parked" vehicles. The x-axis represents confidence levels, and the y-axis shows the F1 score. The blue line represents the overall performance, with an average F1 score of 0.97 at a confidence level of 0.450. The orange line shows the performance for "Wrong-Parked," and the light blue line for "Right-Parked." The curves indicate that the model maintains high F1 scores up to a confidence level of about 0.8, after which the F1 score drops sharply.

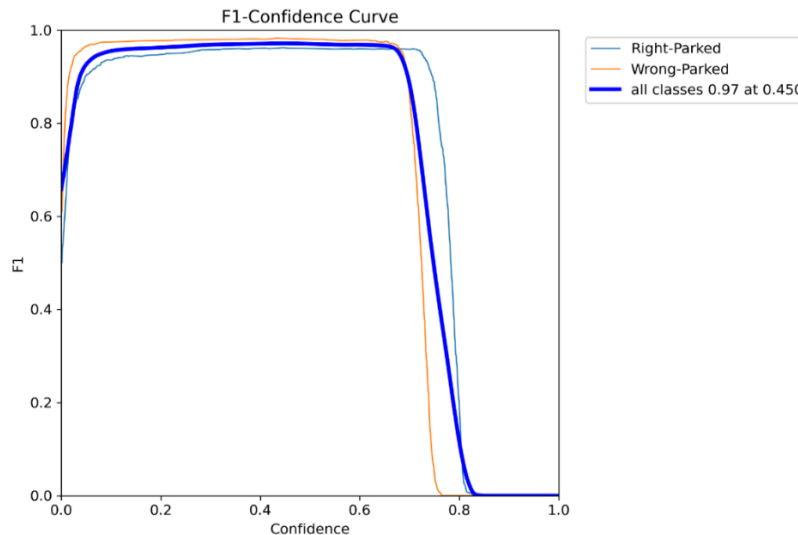


Figure 11 YOLO v8 (Large) F1- Confidence - Curve.

Figure 12 Precision-Recall Curve visualizes the performance of a classification model for two classes: "Right-Parked" and "Wrong-Parked." The curves indicate the trade-off between precision and recall for each class, showing that the model performs very well, with high precision and recall values close to 1.0 for most thresholds. The mean Average Precision (mAP) at a threshold of 0.5 is 0.982, indicating excellent overall performance for both classes.

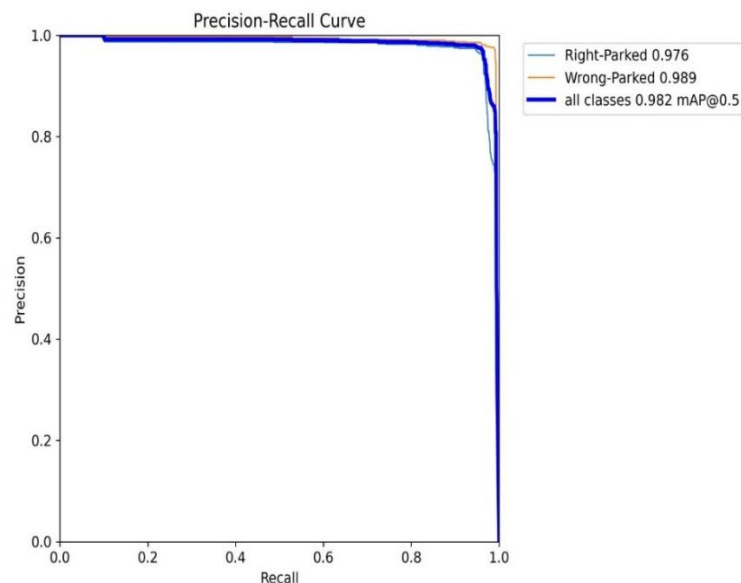


Figure 12 YOLO v8 (Large) F1- Confidence - Curve.

Figure 13 shows the final result of YOLOv8 Large model which correctly detect right

and wrong park cars.



Figure 13 YOLO v8 (Large) F1-Confidence - Curve.

Table 04 below shows the precision, recall, and F1-score of the YOLO V8(Large) model. Hence, the accuracy has improved to **94%**.

Table 04 YOLO v8(Large) Results

Class	Precision (%)	Recall (%)	F1-score (%)
Right parked	97	96.2	96.9
Wrong parked	97	98.1	97.8
Background	68	78.9	73.2

CONCLUSION

In conclusion, the Intelligent Parking Management System for Smart Cities project is a promising solution to the pressing challenges of urban parking. By harnessing the power of computer vision and real-time data analysis, we aim to revolutionize parking management in modern cities. Our innovative approach will not only ease the burden on drivers in their quest for available parking spaces but also contribute to reduced traffic congestion, decreased environmental pollution, and enhanced overall urban mobility. With a user-friendly interface and efficient detection of illegal parking, our system will offer a seamless and convenient experience for both residents and visitors. We envision a future where parking is no longer a time-consuming and frustrating endeavor, but rather a well-organized and sustainable part of smart city living. This proposal paves the way for a more connected, efficient, and environmentally responsible urban landscape.

References

- [1] "What is a Smart City? A Comprehensive Introduction." Accessed: Jul. 26, 2025. [Online]. Available: <https://www.opendatasoft.com/en/blog/what-is-a-smart-city-a-comprehensive-introduction/>
- [2] "What is a Smart City? – Definition and Examples - TWI." Accessed: Jul. 26, 2025. [Online]. Available: <https://www.twi-global.com/technical-knowledge/faqs/what-is-a-smart-city#SmartCityTechnologies>
- [3] "What Is Artificial Intelligence? Definition, Uses, and Types | Coursera." Accessed: Jul. 26, 2025. [Online]. Available: <https://www.coursera.org/articles/what-is-artificial-intelligence>
- [4] "Supervised Machine Learning - GeeksforGeeks." Accessed: Jul. 26, 2025. [Online]. Available: <https://www.geeksforgeeks.org/machine-learning/supervised-machine-learning/>
- [5] "What Is Computer Vision? | Microsoft Azure." Accessed: Jul. 26, 2025. [Online]. Available: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision#object-classification>
- [6] "YOLOv8: Object Detection Explained | Keylabs." Accessed: Jul. 26, 2025. [Online]. Available: <https://keylabs.ai/blog/under-the-hood-yolov8-architecture-explained/>
- [7] A. Khanna and R. Anand, "IoT based smart parking system," *2016 International Conference on Internet of Things and Applications, IOTA 2016*, pp. 266–270, Sep. 2016, doi: 10.1109/IOTA.2016.7562735.
- [8] R. Nithya, V. Priya, C. Sathiya Kumar, J. Dheeba, and K. Chandrababha, "A Smart Parking System: An IoT Based Computer Vision Approach for Free Parking Spot Detection Using Faster R-CNN with YOLOv3 Method," *Wirel Pers Commun*, vol. 125, no. 4, pp. 3205–3225, Aug. 2022, doi: 10.1007/S11277-022-09705-Y.
- [9] M. Safran, A. Alajmi, and S. Alfarhood, "Efficient Multistage License Plate Detection and Recognition Using YOLOv8 and CNN for Smart Parking Systems," *J Sens*, vol. 2024, no. 1, p. 4917097, Jan. 2024, doi: 10.1155/2024/4917097.
- [10] M. Alymani, L. A. Almoqhem, D. A. Alabdulwahab, A. A. Alghamdi, H. Alshahrani, and K. Raza, "Enabling smart parking for smart cities using Internet of Things (IoT) and machine learning," *PeerJ Comput Sci*, vol. 11, p. e2544, Jan. 2025, doi: 10.7717/PEERJ-CS.2544/SUPP-2.
- [11] A. Gupta, S. Kulkarni, V. Jathar, V. Sharma, and N. Jain, "Smart Car Parking Management System Using IoT," *American Journal of Science, Engineering and Technology 2017, Volume 2, Page 112*, vol. 2, no. 4, pp. 112–119, Nov. 2017, doi: 10.11648/J.AJSET.20170204.13.
- [12] Saravanan G, Brettlee salomon, Cibi S, Dharun T, and Hari Krishnan P, "Smart Parking System with Computer Vision," *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 2, no. 12, pp. 2816–2820, Dec. 2024, doi: 10.47392/IRJAEH.2024.0389.