ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 06, June 2025

Automated Car Parking System

[1] Dr. Virender Ranga, [2] Rishi Raj, [3] Utkarsh, [4] Varun Kumar

[1] Associate Professor, Department of Information Technology, Delhi Technological University, Delhi, India
[2] [3] [4] Department of Information Technology, Delhi Technological University, Delhi, India
Corresponding Author Email: [1] virenderranga@dtu.ac.in, [2] raj22rishi@gmail.com, [3] utkarshrai3000@gmail.com, [4] varunkumar15103@gmail.com

Abstract*— The rapid increase in the number of vehicles in the urban areas has increased parking challenges leading to severe congestion, inefficient utilization of available parking spaces and unnecessary fuel consumption. Traditional parking systems largely depend on manual operations making them more prone to human errors, delays and mismanagement. To address these issues, this paper presents an Automated Car Parking System that uses computer vision and deep learning techniques for smart parking space management. OpenCV is used for real-time detection of parking spaces, YOLOv3 for accurate vehicle detection and tracking and EasyOCR is integrated for license plate detection and recognition. The system also explicitly displays the availability of parking slots reserved for disabled individuals, promoting accessibility and inclusivity. The proposed system aims to improve traditional parking systems, optimize space usage, improve operational efficiency, minimize delays and contribute to the development of smarter and more sustainable urban infrastructure.

Index Terms*— Automated Parking System, EasyOCR, License Plate Recognition, Object Detection, OpenCV, Parking Management, YOLOv3.

I. INTRODUCTION

The exponential rise of vehicles in urban spaces has created significant parking challenges that require efficient Parking Management systems. Traditional manual systems are not good at managing the increased number of vehicles, leading to inefficiencies such as excessive waiting times, long queues, congestion and difficulty in locating vacant parking slots. These systems are also prone to errors in vehicle tracking and fee collection resulting in loss of time and money and adding to user inconvenience. Drivers spend more time searching for parking spaces, causing traffic congestion while poor space utilization leads to unnecessary wastage of available slots. Moreover, manual tracking of vehicle entry and exit increases the chances of errors and unauthorized parking and there is difficulty in ensuring reserved parking for disabled individuals due to the lack of proper allocation and monitoring. To address these issues, this research aims to develop an Automated Car Parking System that with the help of computer vision and deep learning techniques will try to optimize large parking space management. The system will detect and track available parking slots in real-time using OpenCV, identify vehicles entering and exiting using YOLOv3, recognize and log vehicle license plates using EasyOCR for automated entry/exit management and explicitly count and display the number of slots reserved for disabled individuals. Implementing such a system can play a crucial role in modern urban development and the advancement of smart city infrastructure by enhancing traffic management, operational efficiency and security. It will reduce congestion caused by vehicles searching for parking spaces by providing real-time availability of slots, automating fee collection and ensuring proper allocation of reserved slots thereby contributing significantly to urban parking management and improving the overall quality of life for urban residents.

II. RELATED WORK

A. Overview of Automated Parking

The idea of automated parking systems has gone over several technological iterations over the past years, leveraging technologies such as sensor-based systems, IoT and computer vision to solve this problem. Earlier systems relied on RFID and ultrasonic sensors for detecting vacant slots. However, these methods had limitations in scalability and cost-effectiveness [1]. More recent approaches use deep learning and image processing techniques to enhance detection accuracy and real-time monitoring of the vehicles and parking slots [2].

B. Computer Vision in Parking Slot Detection

Recent advancements in computer vision have enabled automated parking systems to process real-time video feeds and detect vacant and occupied slots with high accuracy. Studies have demonstrated the effectiveness of OpenCV for edge detection and contour-based slot recognition while deep learning-based models like YOLO and Faster R-CNN provide more robust object detection capabilities in dynamic environments [3, 4].

C. License Plate Recognition for Automated Parking Management

Optical Character Recognition (OCR) has become an important part of smart parking systems, allowing automated vehicle tracking and fee management. Traditional OCR techniques struggled with varied lighting conditions and complex fonts, but modern frameworks like EasyOCR and Tesseract have improved recognition accuracy, making them viable for large-scale parking management systems [5, 6].

D. Comparative studies on Detection Models

Table I: Comparison of Different Models for Parking Space Detection

Model	Accuracy	Cost	Complexity	Speed	Real-time Use	Why ours?
Sensor-Based	High	High	Extra hardware	Fast	Scalable but costly	No hardware needed
Background Subtraction	Medium	Low	Simple	Fast	Poor in changing light	More robust with tuning
Deep Learning	Very High	High	Heavy Training	Slow	Accurate, slow	Fast, no big data needed
Hybrid	Very High	Very High	Complex Setup	Fast	Accurate, hardware needed	Hardware-free, still accurate
ML Classifiers	High	Medium	Needs Features	Moderate	Manual tuning needed	No feature extraction
Edge AI	High	Medium-High	Needs Hardware	Fast	Device-bound	Runs on standard systems
Our Model	High	Medium	Balanced	Fast	Reliable in all lighting	Efficient image processing

Table II: Comparison of Different Models for Parking Space Detection

Model	Accuracy	Cost	Complexity	Speed	Real-time Use	Why ours?
Traditional OCR	Medium	Low	Simple	Fast	Struggles with noise	Handles distortions & fonts better
DL-based LPR	Very High	High	Heavy	Slow	Accurate, but slow	Good balance of speed & accuracy
Hybrid (CV + OCR + ML)	Very High	High	Complex Setup	Moderate	Accurate, but complex	Avoids extra ML while staying accurate
Cloud LPR APIs	Very High	High	API-Based	Fast	Needs internet	Runs offline locally
Edge AI LPR	High	Medium-High	Needs Hardware	Fast	Needs special hardware	Works on normal machines
Our Model	High	Medium	Balanced	Fast	Reliable real-time	Detects plates + reads text efficiently

III. RESEARCH METHODOLOGY

A. Overview of Automated Parking

For the purpose of this research we have used a public dataset of Parking Lot Detection Counter from Kaggle[9]. This dataset is designed for the task of detecting and counting empty and occupied spots in a parking area. It contains 6090 images and 4 videos taken in various lighting and weather conditions.

B. Data Preprocessing

To improve the accuracy and efficiency of the models several preprocessing steps are applied such as image resizing where all the images are resized to a standard dimension to ensure consistency across the dataset. This image is then converted to grayscale to reduce complexity. Data augmentation along with Noise reduction with the help of Gaussian blur and median filtering are applied to remove noise and improve detection accuracy.

C. Parking Slot Detection Using OpenCV

For Parking slot Detection, OpenCV is used to process video frames and detect vacant and occupied parking spaces in real time. To implement parking slot detection, we use OpenCV, Numpy, Pickle and Cvzone along with Matplotlib.

The Parking Slot Detection algorithm processes video feeds frame-by-frame to identify parking availability in real time. It begins by loading the input video and extracting individual frames for analysis. Using manual mouse-click events, parking slots are initially marked and labeled as either standard slots or reserved slots for disabled individuals, with their coordinates stored for future reference. Each frame undergoes preprocessing: the region of interest (ROI) is cropped, converted to grayscale to reduce computational load, and smoothed using Gaussian blur to minimize noise. A global thresholding technique with binary inversion is applied to enhance contrast, followed by dilation to improve feature visibility. Occupancy detection is performed by counting non-zero pixels within each slot's ROI—slots with pixel counts below a 400-pixel threshold are classified as vacant (marked green), while those exceeding it are tagged as occupied (red). Reserved slots for disabled individuals are highlighted in cyan/blue, and standard reserved slots in green. Results are dynamically overlaid on the video feed displaying real-time counts of slots for both regular and disabled users. The system continually updates frame to ensure proper monitoring, enabling drivers to identity available slots efficiently while optimizing space utilization

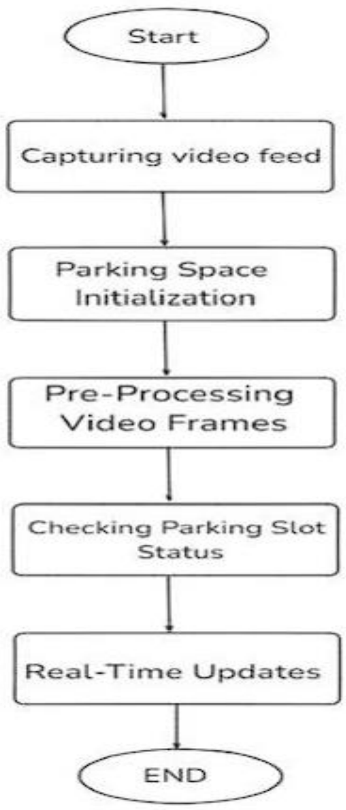

Fig. 1. Flowchart for Parking Slot Detection

D. Parking Slot Detection Using OpenCV

We conducted custom training of the YOLOv3 model to refine it for our specific object detection task, the results of this training are depicted in the following graphs. The process of training consisted of fine-tuning the model on a dataset that was labeled using transfer learning, which gave us the chance to leverage pre-trained weights while shaping the model to our custom classes. The graph shows key metrics such as training and validation losses (box loss, classification loss, and distribution focal loss), which show a consistent decline over epochs, showing increased model performance. Also, metrics like precision, recall, and mean Average Precision (mAP) show significant improvements as training progresses, with mAP values steadily increasing for both IoU thresholds of 0.5 and 0.5:0.95. These results highlight the effectiveness of the training process in achieving accurate object detection while reducing errors, making the model well-suited for deployment in real-world applications.

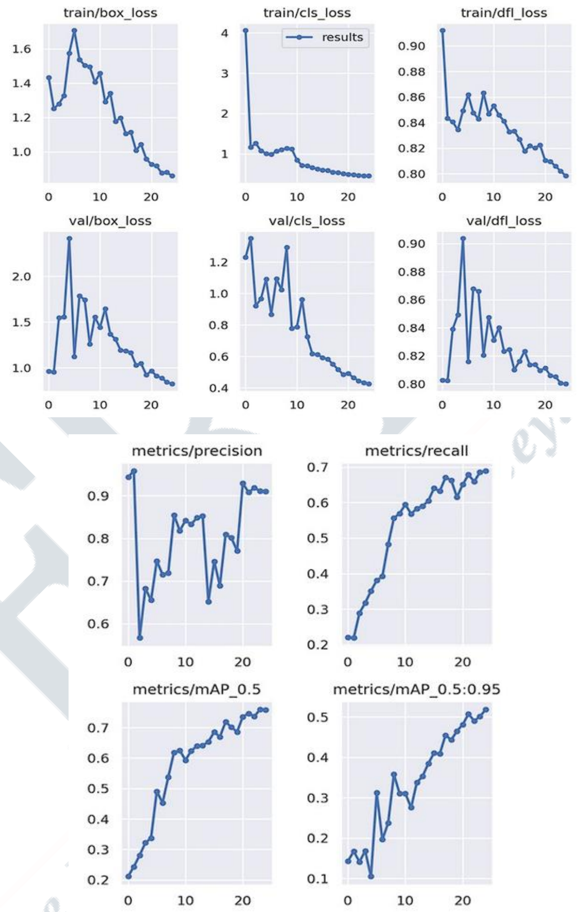

Fig. 2. Result of training of the model

E. Number Plate Detection and Recognition Using YOLOv3 and EasyOCR

Number plate detection and recognition is an important part of the automated parking system. The system used YOLOv3 for license plate detection and EasyOCR for recognizing the characters on the plate. The approach gave correct and efficient vehicle identification. To implement number plate detection and recognition we use OpenCV, Numpy, YOLOv3 and EasyOCR along with Matplotlib. This method gives a correct and real-time solution for vehicle identification, ensuring a smooth parking management experience. The next section will discuss system performance and evaluation metrics.

The Number Plate Detection and Recognition algorithm combines YOLOv3 and EasyOCR to accurately identify and extract text from vehicle license plates. The process begins with loading the trained YOLOv3 model along with the configuration files and class names to prepare for object detection. Input images are read and converted into Binary

Large Objects (Blobs) for efficient processing which are then passed through the YOLOv3 model for inference. This model detects objects in each image and generates a bounding box from which the box corresponding to the licence plate is extracted. Non-Maximum Suppression is applied on it to eliminate overlapping detections, retaining only the most probable result. The detected region of interest containing the license plate is cropped and preprocessed by converting it to grayscale first, then applying global thresholding for binary conversion, performing binary inversion to highlight the text and using dilation to enhance text visibility. The preprocessed license plate image is then passed into EasyOCR for character recognition where the text is extracted only if the confidence score exceeds 04. Finally, the recognized number plate text is displayed or stored for further use ensuring a robust and efficient pipeline for license plate detection and recognition.

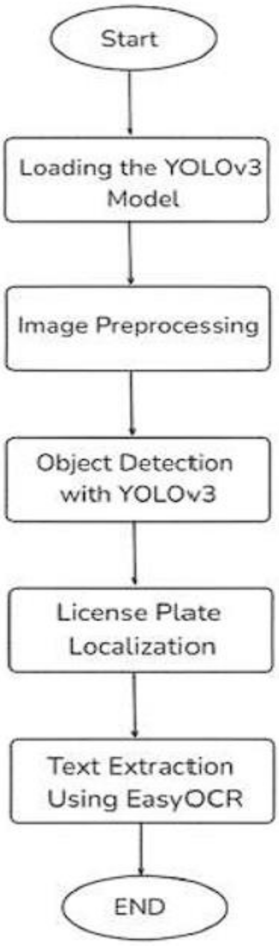

Fig. 3. Flowchart for Number Plate Detection and Recognition

IV. RESULT AND ANALYSIS

The performance of this automated parking system was evaluated based on the ability to correctly detect vacant and occupied parking slots and also recognize vehicle license plates in real-time.

The OpenCV based slot detection method provided reliable outcomes under natural lighting conditions. While extreme lighting scenarios such as bright sunlight or very dark lighting pose challenges and the use of adaptive thresholding helped in improving detection stability. The system accurately identified and differentiated between normal and disabled parking slots ensuring compliance with accessibility requirements. For plate recognition the combination of YOLOv3 and EasyOCR provided efficient detection and text extraction. The system performed well with clearly visible number plates but faced some difficulties when handling damaged, highly reflective or distorted plates. The implementation of non-maximum suppression helped improve the detection process by limiting redundant bounding boxes, ensuring correct identification by setting a confidence threshold with an accuracy of 90.96%.

Table III: Accuracy of the proposed system

Work	Component	Number of samples	Accuracy
This Work	Number plate Detection	100	93.02%
This Work	Character Recognition	-	90.96%
[10]	Number plate Detection	101	98.02%
[10]	Character Recognition	768	96.22%
[11]	Number plate Detection	250	85%
[11]	Character Recognition	214	80%
[12]	Number plate Detection	95	97.89%
[12]	Character Recognition	-	96.84%

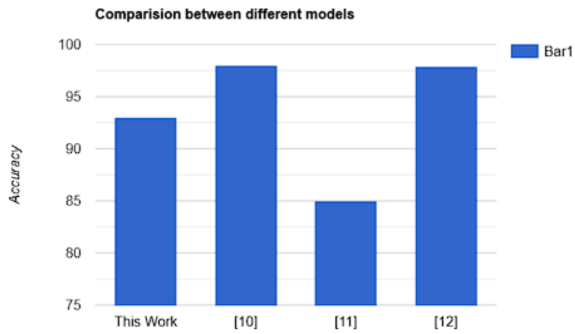

Fig. 4. Comparison of Accuracies

Overall, the model exemplified adaptability when tested with live video feeds and pre-recorded datasets. Both the libraries OpenCV and YOLOv3 allowed a good balance between speed, accuracy and efficiency. The ability to process frames in real time ensures that the system can be deployed in large scale parking facilities.

Fig. 5. Live Parking Feed

Fig. 6. Marking Parking Slots purple for normal and cyan for specially abled persons

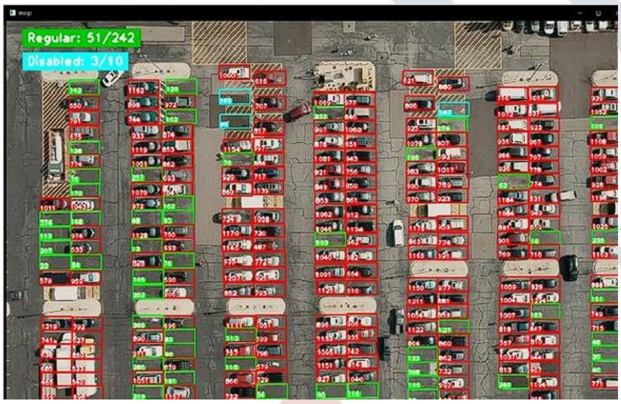

Fig. 7: Occupied slots are displayed in red boxes, free slots are displayed in green boxes and free slots for specially abled are displayed in cyan boxes

Fig. 8. Image of car for number plate recognition

Fig. 9. Dectecting Region of Interest

Fig. 10. Processing the croped Number Plates

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 06, June 2025

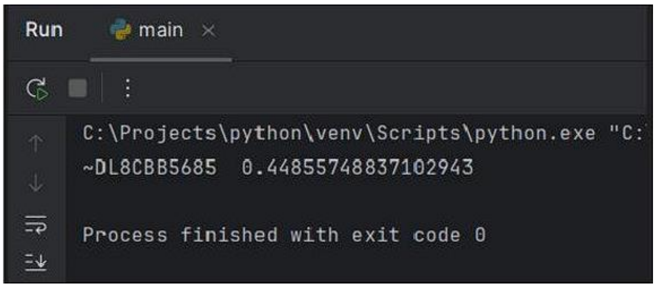

Fig. 11. Recognized Number Plate

V. CONCLUSION

This proposed automated parking system solves the challenges of the current traditional parking management by using CV techniques by real time slot detection and license plate recognition. Here, OpenCV used for detecting vacant and occupied slots and YOLOV3 with EASYOCR for vehicle recognition promises a robust, accurate and scalable solution. The system successfully improves parking efficiency, minimizes manual human involvement and increases accessibility by distinguishing the normal and reserved parking locations. However further improvements can be made to enhance functionality and usability in real time scenarios. Future work will involve integrating a database and a user-friendly interactive interface that allows the users to access parking reservations via the internet. Additionally, a slot booking could be implemented, allowing users to reserve their favorable parking spaces in advance. The system will also store vehicle details detected through number plate recognition in a database to enhance security and allow tracking of parked vehicles. These enhancements will create a more comprehensive and intelligent parking management system, making it even more suitable for real-world deployment in smart cities and commercial parking facilities.

REFERENCES

[1] Khan, A., & Khan, Z. (2020). Smart Parking System: Challenges and Opportunities. Int. J. Adv. Comput. Sci. Appl.

[2] Zhang, G., Li, X., & Liu, H. (2018). IoT-Based Smart Parking System with Real-Time Data Collection. IEEE Trans. Ind. Informatics.

[3] Zhou, Y., Zhang, X., & Li, Q. (2017). Intelligent parking management system based on image processing. J. Comput. Appl.

[4] Kim, D., Lee, S., & Kim, J. (2019). Deep learning-based parking space detection using object detection models. IEEE Access.

[5] Zhao, X., Liu, J., & Wang, L. (2019). License plate recognition for automated parking fee calculation. J. Intell. Transp. Syst.

[6] Bhat, S., Sharma, R., & Kumar, A. (2021). A review on the application of license plate recognition in parking management. Int. J. Comput. Sci. Eng.

[7] Sundararajan, V., Li, C., & Gopalakrishnan, G. (2018). Real-time vehicle detection and tracking using machine learning. Int. J. Comput. Vis. Image Process.

[8] Hussain, S., Rashid, H., & Siddiqui, S. (2020). Vehicle detection and tracking in parking lots using deep learning. Int. J. Electr. Comput. Eng.

[9] Parking Lot Detection counter dataset from Kaggle: https://www.kaggle.com/datasets/iasadpanwhar/parking-lot-detection- counter

[10] Sharma, D., & Singh, R. (2020). Automated parking management system using YOLO and EasyOCR. Procedia Computer Science, 167, 1234–1242. https://doi.org/10.1016/j.procs.2020.03.262

[11] M M Shidore, and S P Nante- (2011) "Number Plate Recognition for Indian Vehicles" International Journal of Computer Science and Network Security 11 (2): 143-146

[12] K Tejas, K Ashok Reddy: D Pradeep Reddy, K P Bharath, R Karthik and M. R. Kumar: (20 IS) 'Efficient License Plate Recognition System with Smarter Interpretation Through IoT," Bansal J, Das K, Nagar A, Deep K, Ojha A (eds) Sofi Computing for Problem Solvi Advances in Intelli ent stems and Com ut• 817: 207-220.