ESP IJAST
ESP International Journal of Advancements in Science & Technology
ISSN: 2583-9233/ Volume 4 Issue 4 October-December 2026
Paper Id: IJAST-V4I4P102/Page No: 09-15

Original Article

Smart Parking Using Computer Vision Techniques

Anirudh Dev[1], Prof. Er. Rahul Sharma[2]
[1]*B.Tech. Scholar, Computer Science & Engineering, Arya College of Engineering & I.T., Jaipur, India*
[2]*Professor, Computer Science & Engineering, Arya College of Engineering & I.T., Jaipur, India*

Received Date: 05 October 2026 Revised Date: 10 October 2026 Accepted Date: 15 October 2026

Abstract: *Finding a free parking space in busy cities, shopping malls, offices, and campuses wastes time and fuel, adds to traffic inside parking areas, and frustrates drivers. Traditional parking management relies on human attendants, tickets, or a sensor fitted at every slot, which is slow, costly, and hard to maintain. Computer vision offers a practical alternative: the CCTV cameras that already watch a parking area can be combined with artificial intelligence to find vehicles and decide, in real time, which slots are occupied and which are vacant.*

This paper presents a smart parking system based on computer vision. It reviews classical feature-based methods and deep learning detectors such as Faster R-CNN, SSD, YOLO, and RetinaNet, together with earlier smart parking studies [1]–[5]. It then describes a five-stage pipeline consisting of image capture, image processing, vehicle detection, data analysis, and information display, supported by authentication, cloud and edge deployment, and performance optimisation techniques such as lightweight models and temporal smoothing. The system counts the available spaces and shows live availability on LED boards, mobile applications, and websites.

The paper also discusses evaluation measures, real-time applications, advantages, and challenges such as installation cost, poor light and bad weather, privacy, and maintenance, followed by future directions including integration with autonomous vehicles, online reservations, smart city infrastructure, and AI-driven fee collection. The study shows that single-stage detectors make real-time parking monitoring possible on practical hardware, and that an existing camera view can replace many individual slot sensors.

Keywords: *Smart Parking, Computer Vision, Vehicle Detection, YOLO, Deep Learning, Real-Time Occupancy Detection, Smart City*

I. INTRODUCTION

Parking is a daily problem of modern urban life. As city populations and the number of vehicles grow, parking spaces become scarce, and drivers often circle a parking area again and again before finding a free slot. This wasted driving increases congestion inside the parking area, burns extra fuel, adds to pollution, and costs people valuable time.

Computer vision is the branch of artificial intelligence that enables machines to interpret images and video in a way similar to human vision. It can recognise objects and patterns, make decisions from visual data, and process live CCTV feeds for automation. In smart parking, cameras watch the parking area, a trained model finds each vehicle, and software works out which slots are free. The result is shown to drivers before or as they enter, so that they can go straight to an available space.

Many parking areas already have CCTV cameras installed for security. Reusing these cameras for parking management is attractive because one camera view can cover many slots, while a sensor-based system needs a separate device at every slot. Advances in deep learning, especially fast single-stage detectors such as YOLO [13], and large public datasets such as COCO [16] have made accurate real-time vehicle detection possible on ordinary hardware.

Building a practical smart parking system involves more than choosing an algorithm. A system must cope with several real-world problems:

- Manual parking management is slow and depends on human staff
- Occupancy information is delayed, so drivers are not told where free space is
- Vehicles vary in size, colour, and position, and slots may be partly blocked by other vehicles
- Light, shadows, rain, and night conditions change the appearance of the scene
- Continuous camera monitoring raises privacy and security concerns

This research focuses on understanding the main computer vision techniques for vehicle detection and on designing an efficient, reliable, and scalable smart parking system that provides live availability information.

II. LITERATURE REVIEW

This section summarises approaches to parking management, classical and deep learning methods for detecting vehicles, earlier smart parking studies, and the gaps that motivate the present work.

This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/2.0/)

A. Parking Management Approaches

Conventional parking areas are managed by attendants, tickets, and barriers, and the staff must judge by eye how many slots remain. Sensor-based systems place an ultrasonic, magnetic, or infrared sensor at each slot, which gives accurate results but needs many devices, wiring, and regular maintenance. Vision-based systems use cameras instead, so a single camera can watch many slots and the same video can also serve security purposes.

B. Classical Vision Methods

Classical methods extract hand-designed features and classify them with a separate classifier. SIFT finds distinctive local points that stay similar when an image is scaled or rotated [6]. HOG describes the shape of an object by the directions of its edges and, together with a Support Vector Machine and a sliding window, became a standard way to detect objects such as pedestrians [7]. Simple techniques such as comparing each frame with an image of the empty lot can also flag changes in a slot. These methods are light and easy to explain, but their accuracy drops under changing light, shadows, rain, and camera angle, and they must be tuned for each parking area.

C. Deep Learning Detectors

Deep learning allows the computer to learn features from data. AlexNet showed in 2012 that a deep CNN trained on a GPU could greatly improve image recognition [8], and ResNet added shortcut connections so that very deep networks can be trained [9]. Such networks serve as backbones that extract features for detectors. Two-stage detectors first propose candidate regions and then classify them: R-CNN [10] introduced the idea and Faster R-CNN [11] made it faster by sharing features with the region proposal network. They are accurate but slower.

Single-stage detectors predict boxes and classes in one pass over the image. SSD uses feature maps of different sizes to find objects of different scales [12]. YOLO divides the image into a grid and predicts boxes directly, which is fast enough for live video [13]. RetinaNet introduced the focal loss, which reduces the effect of the large number of easy background examples and closes much of the accuracy gap to two-stage detectors [14]. Lightweight backbones such as MobileNet are designed for phones and small devices [15], and transformer-based detectors such as DETR treat detection as a direct set-prediction problem [19]. The COCO dataset contains everyday scenes with many objects and is the standard benchmark for pre-training and testing detectors [16].

D. Related Smart Parking Studies

Several recent studies apply these techniques to parking. Nithya et al. describe an IoT-based computer vision approach for free parking spot detection that combines Faster R-CNN with YOLOv3 [1]. A parking information system based on an artificial neural network has been reported in Transportation Research Procedia [2]. Zhao et al. propose CMCA-YOLO, a real-time object detection model designed for parking lot surveillance imagery [3]. Safran et al. present an efficient multistage license plate detection and recognition method that uses YOLOv8 and a CNN for smart parking systems [4], and Nguyen and Sartipi describe a smart camera parking system with automatic parking spot detection [5]. Taken together, these studies show that YOLO-family and CNN-based models are the most common choice for parking detection, and that cameras can support both occupancy counting and vehicle identification.

E. Limitations and Research Gap

Existing methods and systems still have limitations:

- Classical methods are weak under changes in light, shadow, and weather
- Accurate deep models can be heavy for low-cost edge hardware
- Partly hidden vehicles and vehicles parked across slot lines are easily misjudged
- Single-frame decisions can flicker when a vehicle is only briefly detected or missed
- Continuous surveillance raises privacy concerns, and models can be fooled by deliberate image changes [18]

This work addresses these points by describing a complete smart parking pipeline with slot-level decisions, temporal smoothing, authentication and reliability measures, and a scalable cloud and edge deployment.

III. PROPOSED SYSTEM

The proposed smart parking system consists of five main stages that work together from image capture to the information shown to the driver, supported by a security and deployment layer.

A. Image Capture Module

CCTV cameras continuously record the parking area. Each camera is placed so that a group of slots is clearly visible, and video frames are taken from the live stream at a fixed rate.

Functions

- Capturing video from existing or newly installed CCTV cameras

- Extracting frames from the live stream at a suitable rate
- Sending frames to the processing unit on an edge device or a cloud server

B. Image Processing Module

Frames are prepared for the model by computer vision algorithms so that detection is stable under different conditions.

Functions

- Resizing and normalising frames to the input size of the model
- Correcting brightness and contrast for shadows, glare, and night scenes
- Data augmentation such as flips, crops, and colour changes during training to make the model more robust

C. Vehicle Detection Module

This is the core of the system, where AI models such as YOLO detect vehicles and slot occupancy.

Technologies Used:

- Python as the primary development language
- OpenCV for image processing and computer vision primitives [17]
- YOLO and other deep learning detectors for real-time vehicle detection [12], [13]
- Transfer learning, in which a model pre-trained on COCO [16] is fine-tuned on images of the parking area

D. Data Analysis Module

The system compares the detected vehicles with the marked parking slots, counts the available spaces, and updates the status of each slot. The total number of free spaces is calculated and sent to the display module.

E. Information Display Module

Availability is shown on LED boards at the entrance, in mobile applications, and on websites in real time, so that the driver sees the number of free spaces before entering and can plan the route.

F. Security and Deployment Layer

Features:

- Authentication verifies identity and access permissions, for example an administrator login, and protects against unauthorised access
- Reliability measures help the system operate consistently and give correct parking information
- Cloud and edge deployment give flexibility in latency and scale

Technologies Used:

- Edge devices that run light models close to the camera
- Cloud servers with GPUs for training and for processing many cameras
- Mobile applications and LED displays for the user-facing status

IV. METHODOLOGY

The working of the smart parking system follows the steps below.

A. Data Collection and Labelling

Images and video are collected from the parking area at different times of day and in different weather. Each vehicle is labelled with a box and a class such as car, bus, truck, or motorcycle. Public datasets such as COCO [16] can be used for pre-training, and the data are split into training, validation, and test sets.

B. Preprocessing and Augmentation

Frames are resized and normalised. Augmentation such as flipping, cropping, scaling, brightness changes, and colour changes creates more varied training examples and reduces overfitting, which is important because a model trained in daylight may fail at night.

C. Model Selection and Training

A single-stage detector such as YOLO is chosen because it is fast enough for live video [13]. Starting from a model pre-trained on a large dataset and fine-tuning it on parking images is usually faster and needs less data than training from scratch. The network is trained with a loss that combines a classification term and a box-position term.

D. Slot Marking and Occupancy Decision

Each parking slot is marked once on the camera view as a polygon. For every detected vehicle box, the part of the slot covered by the box is measured:

$$Overlap = Area(vehicle\ box \cap slot) / Area(slot)$$

If the overlap is above a chosen threshold, the slot is marked occupied; otherwise it is marked vacant. The threshold is set by testing on the actual camera view. The number of available spaces is then:

Available spaces = Total slots − Occupied slots

E. Post-Processing and Temporal Smoothing

Many overlapping boxes may be predicted for one vehicle, so Non-Maximum Suppression keeps the best box and removes the rest, and detections below a confidence threshold are discarded. To prevent the count from flickering when a vehicle is missed in one frame, the status of a slot changes only when the new state is seen in several consecutive frames.

F. Display and Update

The updated counts are sent to the display systems, for example the LED board, the mobile application, and the website. When a vehicle leaves, the count is updated immediately without any human intervention.

G. Evaluation

The detector is evaluated with Intersection over Union (IoU), precision, recall, and mean Average Precision (mAP) [16]. At system level, the proportion of slots whose status is reported correctly, the delay between a change in the lot and the change on the display, and the number of false alarms are measured.

H. Performance Optimisation

Parking monitoring must run continuously and in real time, so several optimisation techniques are used:

- Lightweight models such as MobileNet [15] and single-stage detectors [12], [13]
- Shrinking (compressing) models by pruning and quantisation for edge devices
- GPUs to speed up the work, with several camera frames processed together in batches
- Analysing only some frames per second, since parked vehicles move rarely
- Processing near the camera on edge devices to reduce delay and network use

V. SYSTEM ARCHITECTURE

The system follows a pipeline architecture, shown in Figure 1, in which each stage feeds into the next:

Image Capture → Image Processing → Vehicle Detection → Data Analysis → Information Display

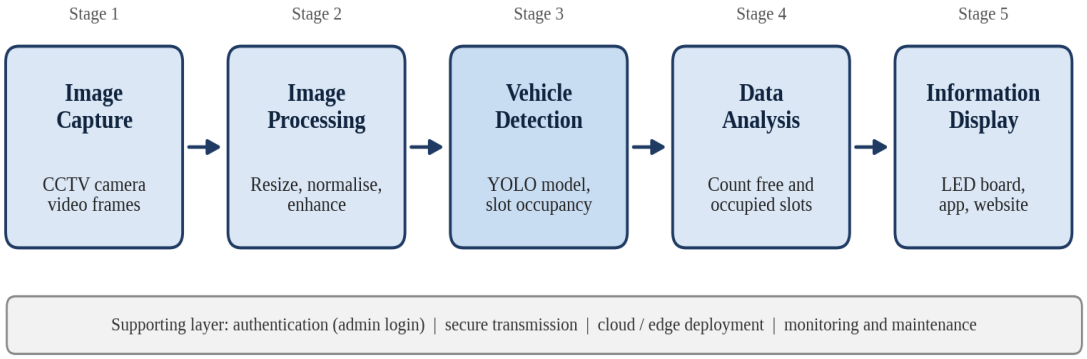


Figure 1: Architecture of the Proposed Smart Parking System

Each component plays a critical role:

- Image capture records the parking area through CCTV cameras
- Image processing resizes, normalises, and enhances the frames
- Vehicle detection finds each vehicle and decides slot occupancy
- Data analysis counts the available spaces and updates the status
- Information display shows the availability on LED boards, apps, or websites

For deployment, lightweight detection can run on edge devices near the cameras, while the cloud handles model training, storage, administrator access, and the data sent to the apps. This flexible split lets the system serve a single small lot or many parking areas of a city.

VI. RESULTS AND DISCUSSION

A. Comparison of Approaches

A comparison of the main approaches shows a clear trade-off between accuracy, speed, and resource use, as summarised in Table 1.

Table 1: Qualitative Comparison of Detection Approaches for Parking Occupancy

Approach	Accuracy	Speed	Suitability for smart parking
Classical (SIFT, HOG + SVM)	Limited under changing light, shadow, and weather	Fast on simple hardware	Only for small, controlled lots with stable lighting
Two-stage (Faster R-CNN)	High	Slower	Good when accuracy matters more than speed and a GPU server is available
Single-stage (SSD, YOLO, RetinaNet)	Good to high	Fast, suitable for live video	Best overall balance for real-time occupancy detection
Transformer-based (DETR)	High	Needs more data and training time	Promising, but heavier to train and deploy
Lightweight (MobileNet backbones)	Small loss in accuracy	Very fast	Edge devices placed close to the camera

The table is qualitative and based on the published literature [6], [15], [19]. Classical methods are light but sensitive to changing conditions [6], [7]; two-stage detectors are accurate but slower [10], [11]; single-stage detectors such as SSD and YOLO are fast enough for live video with good accuracy [12], [13]; and lightweight models run on phones and embedded devices with a small loss in accuracy [15]. Single-stage detectors therefore give the best balance for smart parking, which is consistent with their frequent use in the related studies [1], [3], [4].

B. Example Scenario

Consider a mall with 200 parking spaces:

- CCTV cameras continuously record the parking area
- Frames are processed and a YOLO-based model detects the vehicles
- The system observes 150 occupied and 50 vacant slots
- The display shows: Available Parking Spaces: 50
- When a car leaves, the count updates instantly from 50 to 51 without human intervention

This shows how computer vision can automate a task that is slow and tiring for people and give drivers accurate, live information.

C. Real-Time Applications

The same design can be used in shopping malls, airports, hospitals, offices, university campuses, residential complexes, and city street parking. In each case the benefit is the same: drivers are guided to free space quickly and managers receive real-time data on how the area is used.

D. Authentication and Reliability

Two properties are essential for a system that the public depends on. Authentication verifies identity and access permissions and protects against unauthorised access; for example, only a verified administrator can change slot settings, view stored video, or update the model. Reliability ensures consistent and correct operation and helps the system provide accurate parking information, for example by correctly detecting vacant spaces. Reliability is supported by temporal smoothing, regular camera checks, and a clear “status unknown” message when a camera fails, rather than showing a wrong count.

Note: The comparison above is qualitative and based on the published literature and the design of the system. A measured evaluation of detection accuracy, slot-level accuracy, and delay on a real parking area is planned as future work.

VII. ADVANTAGES

The proposed system offers several significant advantages.

A. Faster Parking

Drivers know in advance whether and where free space is available, which reduces the time spent searching.

B. Less Congestion

Fewer vehicles circle the parking area, which minimises traffic inside it and at its entrance.

C. Fuel Saving and Lower Emissions

Less searching means less fuel burned and lower carbon emissions, which supports cleaner urban mobility.

D. Operational Efficiency

Manual monitoring is no longer needed, staff can concentrate on tasks that need judgement, and managers get real-time data for planning.

E. Use of Existing Infrastructure and Scalability

Existing CCTV cameras can be reused, and one camera can cover many slots, so no sensor is needed at each slot. Cloud and edge deployment allow the system to grow from one lot to many.

VIII. CHALLENGES AND LIMITATIONS

Despite its benefits, a computer vision parking system faces several challenges.

A. High Installation Cost

Cameras, servers, edge devices, and displays need an initial investment, and some areas need extra cameras to remove blind spots.

B. Low Light and Bad Weather

Accuracy can fall at night, in heavy rain, fog, or strong glare. Night-time training data, image enhancement, and suitable lighting or infrared cameras can reduce this problem.

C. Occlusion and Camera Placement

Large vehicles can hide smaller ones, and vehicles parked across slot lines are hard to judge. Careful camera height and angle, and overlapping camera views, help.

D. Privacy Concerns

Continuous surveillance raises concerns about the recording of people and number plates. Processing on the device, storing only counts instead of video, limiting access, and clear notices are good practice.

E. Maintenance and Model Robustness

Cameras need cleaning and regular checks, and the AI model needs updating as vehicles and conditions change. Deep models can also be fooled by small deliberate changes to an image [18], so security of the system is important.

IX. FUTURE SCOPE

The proposed system has significant potential for improvement. Future developments include:

- Autonomous integration: connecting with autonomous vehicles so that they can find and enter free slots by themselves
- Online reservations: reserving parking spots through mobile apps before arrival
- IoT and smart city: tight integration with the wider smart city infrastructure, such as traffic and navigation systems
- AI fee collection: automated, AI-driven parking fee systems, for example with licence plate recognition [4]
- Better night and bad-weather performance through improved models and sensors
- Privacy-preserving processing that runs on the device and stores no video
- Measured evaluation on real parking areas and standard benchmarks

X. Conclusion

Smart parking using computer vision makes cities more efficient by automatically detecting available spaces and delivering real-time information to drivers. Classical methods based on hand-designed features are fast and simple, while deep learning detectors learn their features from data and reach much higher accuracy. Single-stage detectors such as YOLO and lightweight models make real-time detection possible, and with good optimisation the same system can run on cloud servers and on edge devices close to the camera.

However, installation cost, poor light and weather, privacy, and maintenance remain challenges, and the technology must be used responsibly with proper authentication and reliability measures. As AI and computer vision continue to evolve, smart parking will be a core element of future smart cities, supporting faster parking, less congestion, and lower emissions.

XI. REFERENCES

[1] R. Nithya, V. Priya, C. Sathiya Kumar, J. Dheeba, and C. Chandraprabha, "A Smart Parking System: An IoT Based Computer Vision Approach for Free Parking Spot Detection Using Faster R-CNN with YOLOv3 Method," *Wireless Personal Communications*, vol. 125, no. 4, 2022.

[2] "Parking Information System with Artificial Neural Network," *Transportation Research Procedia*, vol. 74, pp. 624–631, 2023.

[3] N. Zhao, K. Wang, J. Yang, F. Luan, L. Yuan, and H. Zhang, "CMCA-YOLO: A Study on a Real-Time Object Detection Model for Parking Lot Surveillance Imagery," *Electronics*, vol. 13, no. 8, Art. no. 1557, 2024.

[4] M. Safran, A. Alajmi, and S. Alfarhood, "Efficient Multistage License Plate Detection and Recognition Using YOLOv8 and CNN for Smart Parking Systems," *Journal of Sensors*, vol. 2024, Art. no. 4917097, 2024.

[5] T. T. Nguyen and M. Sartipi, "Smart Camera Parking System With Auto Parking Spot Detection," in *Proc. Asian Conference on Computer Vision (ACCV) Workshops*, pp. 232–241, 2024.

[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. https://doi.org/10.1109/CVPR.2005.177

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://arxiv.org/abs/1512.03385

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. https://arxiv.org/abs/1311.2524

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. https://arxiv.org/abs/1506.01497

[12] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *Proc. European Conference on Computer Vision (ECCV)*, 2016. https://arxiv.org/abs/1512.02325

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://arxiv.org/abs/1506.02640

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. https://arxiv.org/abs/1708.02002

[15] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861*, 2017. https://arxiv.org/abs/1704.04861

[16] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Proc. European Conference on Computer Vision (ECCV)*, 2014. https://arxiv.org/abs/1405.0312

[17] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proc. International Conference on Learning Representations (ICLR)*, 2015. https://arxiv.org/abs/1412.6572

[19] N. Carion et al., "End-to-End Object Detection with Transformers," in *Proc. European Conference on Computer Vision (ECCV)*, 2020. https://arxiv.org/abs/2005.12872