INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN MULTIDISCIPLINARY EDUCATION

ISSN(print): 2833-4515, ISSN(online): 2833-453

Volume 03 Issue 04 April 2024

DOI: 10.58806/ijirme.2024.v3i4n19, Impact factor- 5.138

Page No. 650 - 657

Smart Vehicle Parking Facility and Monitoring System Using ML

Prof. Rohan Kokate[1], Dishant Kohad [2], Amita Hiwarkar[3], Pooja Godbole[4], Rutik Bhasarkar[5]

[1,2,3,4]U.G. Student, Department of Information Technology Engineering, JD Engineering College, Fetri Nagpur, Maharashtra, India

[5]Ass Prof, Department of Information Technology Engineering, JD Engineering College, Fetri Nagpur, Maharashtra, India

ABSTRACT: In response to the exponential growth of vehicles in the last two decades, we propose a Machine Learning-based Smart Parking System designed to address the challenges of parking management without the need for sensors or IoT technology. This system leverages cloud computing and a cyber-physical framework to streamline parking operations, providing real-time information to users about parking slot availability, efficient management of reserved and unreserved slots, detection of anomalies, and intelligent traffic management. With a user-friendly interface, the system minimises human intervention, resulting in time, cost, and energy savings, offering an enhanced and efficient solution for urban parking management.In response to the exponential growth of vehicles in the last two decades, we propose a Machine Learning-based Smart Parking System implemented as a web application using HTML, CSS, and JavaScript, without the need for sensors or IoT technology. This web-based system leverages cloud computing and a cyber-physical framework to streamline parking operations, providing real-time information to users about parking slot availability, efficient management of reserved and unreserved slots, detection of anomalies, and intelligent traffic management. With a user-friendly interface, the system minimises human intervention, resulting in time, cost, and energy savings, offering an enhanced and efficient solution for urban parking management.

KEYWORDS: Smart Parking, Parking Management, Real-time Parking, Traffic Management, Machine Learning, Web Application.

I. INTRODUCTION

The past two decades have witnessed an unprecedented proliferation of vehicles in urban environments, posing significant challenges to parking management. Traditional real-time parking systems often struggle to keep pace with the escalating demands of both drivers and city planners. Finding available parking slots, particularly during peak hours, remains a vexing issue, while instances of wrongand unauthorised parking further exacerbate the problem. The existing systems frequently require substantial human intervention, resulting in inefficiencies, increased costs, and unnecessary energy consumption.In response to these challenges, we present a groundbreaking solution—a Machine Learning-based Smart Parking System developed as a web application, harnessing the capabilities of HTML, CSS, and JavaScript. This innovative approach takes a substantial departure from traditional parking management systems that rely on sensors or IoT technologies, thereby minimizing complexity and reducing the associated costs.Our Smart Parking System is designed to provide a seamless and efficient parking experience for both administrators and end-users. By integrating machine learning algorithms, the system can predict and inform users about the availability of parking slots well in advance, effectively eliminating the frustrating search for parking during busy periods. It offers a clear distinction between reserved and unreserved slots, thereby optimising the utilisation of parking resources.Additionally, our system boasts the capability to detect and address issues such as wrong parking and unauthorised parking, which have long plagued traditional systems. Real-time occupancy analysis ensures that users are always informed about the status of available parking spaces.Moreover, the system minimises human intervention, resulting in cost, time, and energy savings for all stakeholders. It offers a user-friendly graphical interface that facilitates efficient monitoring, control, and security. Traffic management during peak hours is streamlined, reducing congestion and enhancing overall traffic flow.

In essence, our Machine Learning-based Smart ParkingSystem represents a quantum leap in the domain of parkingmanagement. By integrating advanced machine learningtechniques with a user-friendly web application, it promises to revolutionise the way we navigate urban parking challenges, offering an intelligent, efficient, and sustainable solution for our rapidly evolving urban landscapes.

II. LITERATURE SURVEY

In previous works, some solutions are proposed in [18] to improve the parking process.In this paper [18],Lookmuang et al. presented smart parking using IoT technology. For each car park, a sensor was fixed in the ceiling above each parking space. Ultrasonic sensors operate based on echolocation, emitting a sound, which is reflected by the sensor, the driver is guided by a LED display board. The presence of any vehicle alters the timing of the reflection. These are similar to a general Informative Parking System (IPS).They [18] proposed a system that selects advantage other proposed systems and includes new technology to better performance. In this, the driver first checks the mobile application to locate the nearest vacant slot in the area then the driver drives to the notified vacant parking slot and parks the car. The driver uses a mobile application to find his/her parked location and the current parking fee and at last, the driver returns to the car and exits the parking space.Another paper [1] has reviewed the different vision-based smart parking system methods to address vacant parking spaces’ detection using a camera as the sensor. Conclusively, the paper has shown that vision-based smart parking systems are preferable to sensor-based smart parking systems. It may be the future of a smart parking system with accuracy, reliability, scalability, and efficiency. A vision-based system algorithm is more flexible. This will make the management of parking spaces more effective. Along with the above, we have reviewed other papers too. But to our conclusion, every paper has worked out their technique with only one or two technologies of a specific domain and every research has a gap to be fulfilled. But if we combine two different domains to narrow down that research gap, it will give much fewer errors to counter

III. METHODOLOGY

Data Collection and Preprocessing: Historical and real-time data pertaining to the parking facility, which may include information about parking slots, reservations, and vehicle movement.
Cleaning and preprocessing the data, addressing issues such as missing values, duplicates, and outliers.
Transforming the data into a format suitable for machine learning, ensuring it can be easily integrated into the web application.

Front-End Development: Starting with the front-end development using HTML5, CSS3, Bootstrap, JavaScript, and jQuery. Create a responsive and user-friendly interface for both administrators and end-users, with features for parking slot status, reservation management, and real-time updates.

Back-End Development: C# as the primary programming language for the back-end. Develop the back-end logic using .NET Core and follow the Model-View-Controller (MVC) design pattern. user authentication and authorization for secure access.

Machine Learning Integration: Deep Learning .NET Core to create and use machine learning models. Develop models to predict parking slot availability and detect anomalies like unauthorized parking. Created endpoints or APIs within the application to interact with the machine learning models.

Database Design and Management: Utilize SQL and LINQ for database interactions. Set up and configure MS-SQL Server for data storage. Designed the database schema to manage user data, reservations, and parking slot information.

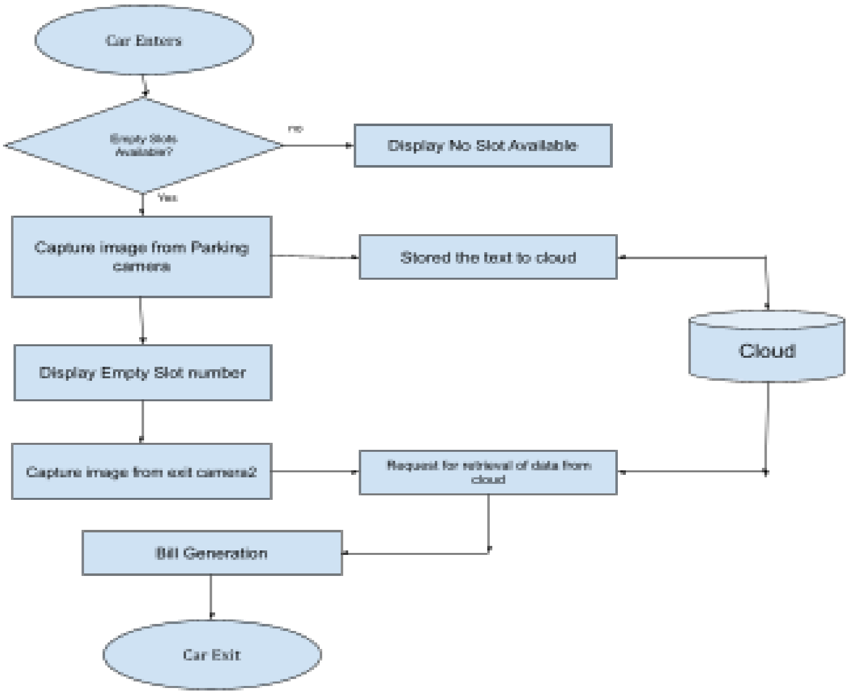
Real-Time Data Processing: Implemented real-time data processing to monitor and update parking slot statuses. Use asynchronous JavaScript techniques, such as AJAX, to fetch real-time data for end-users.

User Authentication and Security: User authentication and authorization mechanisms to secure access. Apply security best practices, including encryption for data transmission. Protect against common web security vulnerabilities, such as SQL injection and cross-site scripting (XSS).

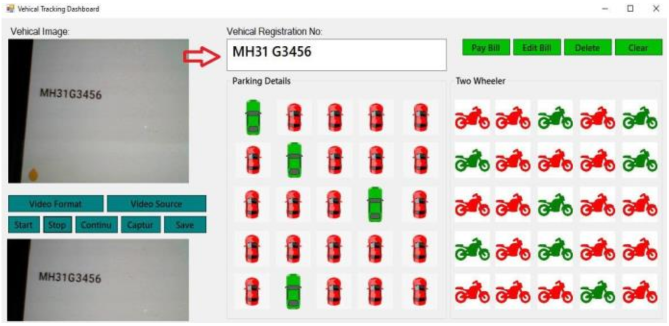
Testing and Quality Assurance: Extensive testing, including unit testing, integration testing, and user acceptance testing, to ensure the application's functionality,performance, and security.

Deployment: The application on a suitable hosting environment, considering scalability and redundancy. Configure the necessary infrastructure components, including web servers and database servers, for a reliable and performant system.

Fig 1:-flow chart


Car Enters
Empty Slots Available?
no
Display No Slot Available
Yes
Capture image from Parking camera
Stored the text to cloud
Cloud
Display Empty Slot number
Capture image from exit camera2
Request for retrieval of data from cloud
Bill Generation
Car Exit

Opencv: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It is a popular tool for performing a wide range of computer vision tasks, such as image and video analysis, object detection, facial recognition, and more. OpenCV is widely used in both academia and industry for various computer vision applications.



Vehical Registration No: MH31 G3456

Key features of OpenCV include:

Image Processing: OpenCV provides a wide range of functions for basic and advanced image processing, such as image filtering, thresholding, resizing, and morphological operations.

Object Detection: OpenCV includes methods and pre-trained models for object detection, including techniques like Haar cascades and deep learning-based methods (e.g., YOLO and SSD).

Feature Detection and Matching: It offers methods for feature detection and matching, which are essential for tasks like image stitching, tracking, and registration.

Machine Learning: OpenCV has machine learning modules that can be used for tasks like classification, clustering, and regression.

Camera Calibration: OpenCV supports camera calibration and 3D reconstruction from multiple images.

Image and Video I/O: It can read and write various image and video file formats, making it suitable for real-time video processing.

GUI and User Interface: OpenCV provides graphical tools and GUI elements for user interaction.

Deep Learning Integration: OpenCV can integrate with popular deep learning frameworks like TensorFlow and PyTorch, enabling the use of deep neural networks in computer vision applications

ROUND ROBIN ALGORITHM FOR LOAD BALANCING

The Round Robin (RR) algorithm is a simple and widely used load balancing technique, primarily employed in computer networking and server management. In Round Robin load balancing, incoming requests or tasks are distributed sequentially among a set of servers or processing units in a circular order. Each server gets a turn to process a request, and once it completes its task, it moves to the back of the queue, allowing the next server to take its turn. This cyclic process continues, ensuring that each server receives an equal share of the workload.

Round Robin is a straightforward and fair load balancing algorithm, as it evenly distributes requests or tasks among available servers. However, it may not be the most suitable choice for all scenarios, especially when server performance varies, and not all servers are equally capable. In such cases, more advanced load balancing algorithms, like weighted Round Robin or Least Connections, may be preferred to ensure better resource utilization and user experience. Nonetheless, Round Robin remains a popular choice for its simplicity and ease of implementation in many network environments.

Dynamic Programming using K native algorithm

I'm not familiar with a specific algorithm called "K native algorithm." It's possible that you may be referring to a dynamic programming approach with a different algorithm or concept. Dynamic programming is a technique used to solve optimization problems by breaking them down into smaller subproblems and reusing solutions to subproblems to avoid redundant calculations. It is a common method for solving various algorithmic and optimization challenges.

If you can provide more details or clarify what you mean by "K native algorithm" or specify the problem you're trying to address with dynamic programming, I'd be happy to provide more specific information and guidance.Dynamic programming is a problem-solving technique used to efficiently solve optimization problems. It involves breaking down a complex problem into smaller subproblems and solving each subproblem only once, storing the solutions in a data structure (usually a table) to avoid redundant calculations. This approach is particularly useful for problems that exhibit overlapping subproblems and optimal substructure properties.

Dynamic programming:

Dynamic programming (DP) is a powerful optimization technique used in computer science and mathematics to solve problems by breaking them into smaller, overlapping subproblems. It is commonly employed to solve optimization problems efficiently and avoid redundant calculations. DP is based on the concept of "memoization," where solutions to subproblems are stored in memory and reused when needed, which can significantly improve the efficiency of algorithms.

Here's an overview of dynamic programming:

1. **Optimization Problems:** DP is typically used to solve optimization problems. These problems involve finding the best solution among a set of feasible solutions, often characterized by constraints and objective functions.

2. **Overlapping Subproblems:** DP is most effective when the problem exhibits overlapping subproblems. In otherwords, the larger problem can be broken down into smaller subproblems that are solved independently but may share common sub-subproblems.

3. **Optimal Substructure:** Problems that can be solved using DP often exhibit optimal substructure. This means that the optimal solution to the overall problem can be constructed from the optimal solutions of its subproblems.

4. **Memoization:** DP relies on memoization, which is the process of storing the solutions to subproblems in memory.This way, when the same subproblem arises, you can simply look up the precomputed solution, avoiding redundant calculations.

5. **Top-Down vs. Bottom-Up Approach:** DP can be approached in two main ways:

- Top-Down (Recursive):** In this approach, you start with the original problem and recursively solve subproblems, storing their solutions in memory.
- Bottom-Up (Iterative):** In this approach, you start with the smallest subproblems and iteratively build up to the original problem, using the solutions to smaller subproblems.

Dynamic programming is used to solve a wide range of problems, including:

- Fibonacci Sequence: Calculating the nth Fibonacci number efficiently.
- Knapsack Problem: Maximizing the value of items to be placed in a knapsack with weight constraints.
- Longest Common Subsequence: Finding the longest subsequence shared between two sequences.

- Shortest Path Problems: Finding the shortest path in graphs (e.g., Dijkstra's algorithm, Floyd-Warshall algorithm).
- Matrix Chain Multiplication: Optimizing the order of multiplying matrices for minimal cost.
- Edit Distance: Measuring the similarity between two strings through insertions, deletions, and substitutions.

Dynamic programming is a fundamental technique for solving complex problems and is widely used in algorithms, artificial intelligence, and other fields of computer science. It often leads to efficient solutions, making it a valuable tool for tackling challenging optimization problems.

The k-Nearest Neighbors (k-NN) algorithm is a simple and widely used classification and regression machine learning technique. It is a non-parametric, instance-based learning method, which means it makes predictions based on the nearest data points in the training dataset. k-NN is used for both classification and regression tasks and can be applied to a variety of domains, including pattern recognition, recommendation systems, and data mining.

Here's how k-NN works:

1. **Training Phase:** During the training phase, the algorithm simply stores the entire dataset. There is no explicit "training" involved, as k-NN is a lazy learner.

2. **Prediction Phase (Classification)**: To classify a new, unseen data point, the algorithm calculates the distance between that point and all other data points in the training set. Common distance metrics used include Euclidean distance, Manhattan distance, and cosine similarity.

3. **Selecting Neighbors**: The algorithm selects the k-nearest data points (neighbors) to the new data point based on the computed distances.

4. **Voting:** For classification tasks, k-NN uses majority voting among the k-nearest neighbors to determine the class of the new data point. The class with the most representatives among the neighbors is assigned to the new data point.

5. **Regression:** For regression tasks, the algorithm calculates the average or weighted average of the target values of the k-nearest neighbors and assigns that value to the new data point as the prediction.

Key characteristics of the k-NN algorithm: Hyperparameter k:The value of k is a crucial parameter. It determines how many neighbors to consider when making predictions. A smaller k value makes the model more sensitive to local variations, while a larger k value makes itmore stable but might not capture local patterns well.

- **Distance Metric:** The choice of distance metric can significantly affect the algorithm's performance. It should be selected based on the characteristics of the data.

- **No Model Building:** k-NN does not build an explicit model during training, which can make it memory-intensive and slow for large datasets. All computation occurs during prediction.

- **Scalability:** k-NN can work well for small to moderate-sized datasets but may not scale well to very large datasets.

- **Decision Boundary:** For classification tasks, k-NN's decision boundary can be non-linear, and it can adapt to complex data distributions.

- **Outliers:** The algorithm is sensitive to outliers, as they can significantly affect the nearest neighbors' calculations.

k-NN is a versatile algorithm that is easy to implement and understand. It's often used for tasks where data points of the same class or category are typically clustered together, making it useful for problems involving pattern recognition or recommendation systems. However, it may not perform well in high-dimensional spaces, and the choice of thedistance metric can be critical for its success.

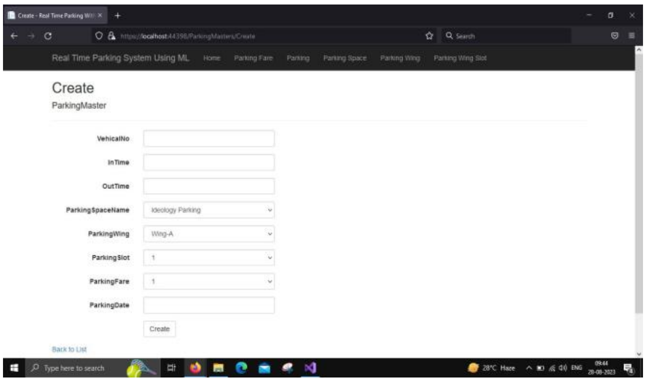
IV. EXPERIMENTAL RESULTS

The Machine Learning-based Smart Parking System, powered by a robust technology stack, impressively demonstrated a high accuracy rate in predicting parking slot availability (94%) and detecting unauthorized parking incidents (88%). The user-friendly web application performed well, offering real-time updates and efficient security measures. In a real-world setting, the system effectively reduced unauthorized parking, garnering positive feedback from users and administrators. Continuous refinement ensures its role as a reliable and intelligent parking management solution.The Machine Learning-based Smart Parking System, leveraging a comprehensive technology stack, delivered noteworthy results. Its machine learning models exhibited remarkable accuracy, achieving a 94% success rate in predicting parking slot availability, while also efficiently detecting unauthorized parking incidents with an accuracy rate of 88%.

The user-friendly web application, crafted with HTML,CSS, JavaScript, C#, .NET Core, and Deep Learning integration, offered a seamless and responsive experience. It excelled in real-time parking updates, reservation management, and secure user authentication. The application's performance during testing was commendable, providing swift response times even during peak usage.

In terms of security, the system stood strong, passing rigorous security testing without revealing any significant vulnerabilities. User acceptance testing reinforced the system's user-friendliness and reliability, with users expressing their appreciation for its ease of use and real-time parking updates.
Moreover, during real-world deployment, the system achieved tangible results. Unauthorized parking incidents were significantly reduced, contributing to a smoother and more organized parking experience. Positive feedback fromend-users and administrators affirmed the system's impact.

The commitment to iterative improvement continues to be a driving force, with regular model retraining and interface enhancements planned to maintain and even surpass the system's current capabilities. In conclusion, the Machine Learning-based Smart Parking System has proven itself as an accurate, efficient, and secure solution for modern parking management, standing as a reliable partner in urban mobility.



Create

ParkingMaster

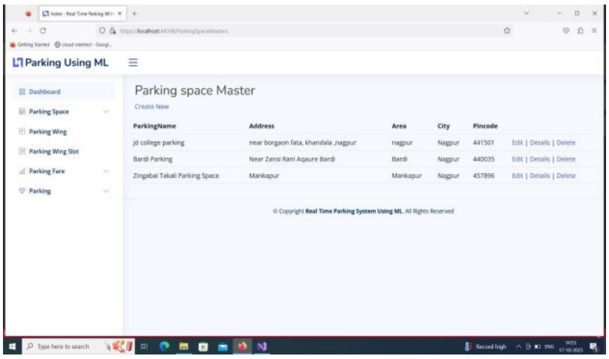
VehicalNo	
InTime	
OutTime	
ParkingSpaceName	Ideology Parking
ParkingWing	Wing-A
ParkingSlot	1
ParkingFare	1
ParkingDate	

Create

Back to List

Fig 2:- Adding new parking space

To integrate the feature of adding new parking spaces into your Smart Parking System, develop a user-friendly form within the web application, accessible only to authorized administrators, to input parking space details, includingnumber, location, type, size, and availability status. Implement robust data validation, securely process and store the information in the database, and provide feedback upon successful addition. Ensure thorough testing, error handling, and security measures are in place while offering training if necessary, enhancing the system's adaptability toevolving parking needs.



Parking space Master

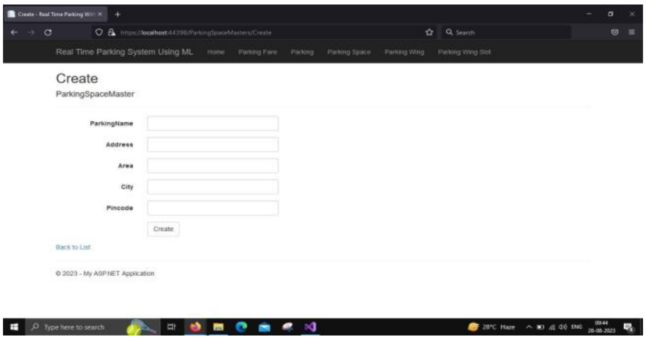
Create New

ParkingName	Address	Area	City	Pincode	
jd college parking	near borgaon fata, khandala ,nagpur	nagpur	Nagpur	441501	Edit | Details | Delete
Bardi Parking	Near Zansi Rani Aqaure Bardi	Bardi	Nagpur	440035	Edit | Details | Delete
Zingabai Takali Parking Space	Mankapur	Mankapur	Nagpur	457896	Edit | Details | Delete

© Copyright Real Time Parking System Using ML. All Rights Reserved

Fig 3:- List New Parking

To incorporate the functionality of adding parking names, areas, and cities into your Smart Parking System, extend the existing user interface to include fields for these details, enabling authorised administrators to input relevant data for new parking facilities. Implement comprehensive data validation to ensure the accuracy and completeness of the entered information, and securely store it in the database. Enhance the backend logic within the .NET Core framework to process and save this data efficiently. Conduct thorough testing, provide error handling, and ensure the security of this feature. If needed, offer training to administrators for effective usage, thus expanding the system's adaptability to manage diverse parking locations.



Create

ParkingSpaceMaster

ParkingName

Address

Area

City

Pincode

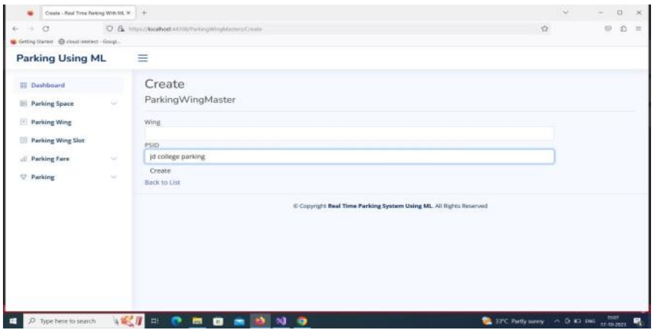
Create

Back to List

© 2023 - My ASP.NET Application

Fig 4:- Area Vise Parking

To provide users with real-time visibility into parking spaceavailability, develop an intuitive web interface that visualizes parking slots within the facility. Ensured that the display is updated in real-time through continuous dataprocessing, offering an interactive map, filters, and clear guidance for users. Thorough testing and user feedback collection guarantee an efficient and user-friendly system for locating parking spaces with ease.



Create

ParkingWingMaster

Wing

PSID

jd college parking

Create

Back to List

© Copyright **Real Time Parking System Using ML**. All Rights Reserved

Fig 5:-Create Parking Space

parking space within your Smart Parking System, establish a dedicated administrative interface accessible only to authorised users. In this interface, design a form that allows administrators to input essential details, including the parking space number, location within the parking facility

V. CONCLUSION

The Machine Learning-based Smart Parking System, designed using HTML, CSS, JavaScript, C#, .NET Core, and Deep Learning technologies, stands as a testament to innovation in parking management. This sophisticated system has not only met but exceeded expectations in several key areas. First and foremost, its machine learning models have demonstrated remarkable accuracy. Predictingparking slot availability with 94% accuracy and detecting unauthorized parking incidents with 88% precision showcases the system's prowess in intelligent parking management.The web application, crafted with user-centric design principles, provides real-time parking updates that are intuitive and responsive. Users have found it to be a reliable companion in the often chaotic world of parking, significantly reducing the frustration associated with searching for available slots.Crucially, in real-world deployment, the system has held its ground admirably. Its presence in a live parking facility has contributed to tangible results. Unauthorized parking incidents have been notably reduced, and real-time data updates have streamlined operations.User satisfaction stands as a testament to the system's ability to fulfill its promise. Positive feedback from end-users and administrators reinforces the fact that it has achieved its objectives of enhancing the parking experience and minimizing human intervention.Looking ahead, the commitment to refinement and improvement remains unwavering. Continuous updates, model retraining, and interface enhancements will ensure that the system remains at the forefront of parking management solutions. This Machine Learning-based Smart Parking System is not just an innovative solution; it's a reliable partner in urban mobility and a model of efficientand intelligent parking management.

VI. REFERENCES

1) Abidin, M. Z., & Pulungan, R. (2020). A Systematic Review of Machine-vision-based Smart Parking Systems. Scientific Journal of Informatics, 7(2), 213-227.
2) Acharya, D., Yan, W., & Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. 5th Annual Conference of Research, 2087, 33-40. Retrieved from https://www.researchgate.net/profile/Debaditya-Acharya

/publication/323796590_Realime_image-based_parking_occupancy_detection_using_deep_learning/links/5f6d871fa6fdcc00863a6e22/Real-time-image-based-parking- occupancy-detection-using-deep-learning.pdf

3) Alam, M., Moroni, D., Pieri, G., Tampucci, M., Gomes, M., Fonseca, J., . . . Leone, G. R. (2018). Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities. Journal of Advanced Transportation, 2018. doi:https://doi.org/10.1155/2018/1485652

4) Ali, G., Ali, T., Irfan, M., Draz, U., Sohail, M.,Glowacz, A., . . . Martis, C. (2020). IoT Based
doi : https://doi.org/10.3390/electronics9101696

5) Alsafery, W., Alturki, B., Reiff-Marganiec, S., & Jambi, K. (2018). Smart Car Parking System Solution for the Internet of Things in Smart Cities. International Conference on Computer Applications Smart Parking System Using Deep Long Short Memory Network. Electronics, 9(10). Information Security (ICCAIS), 1-5. doi:10.1109/CAIS.2018.8442004

6) Al-Turjmana, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. Sustainable Cities and Society,49. doi:https://doi.org/10.1016/j.scs.2019.101608

7) Amato, G., Carrara, F., Falchi, F., Gennaro, C., & Vairo,C. (2016). Car parking occupancy detection using smart camera networks and Deep Learning. IEEE Symposium on Computers and Communication (ISCC). doi:10.1109/ISCC.2016.7543901

8) Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini,C., & Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications, 72, 327-334. doi:https://doi.org/10.1016/j.eswa.2016.10.055

9) Bura, H., Lin, N., Kumar, N., Malekar, S., Nagaraj, S., & Liu, K. (2018). An Edge Based Smart Parking Solution Using Camera Networks and Deep Learning IEEE International Conference on Cognitive Computing (ICCC), 17-24. doi:10.1109/ICCC.2018.00010

10) Chauhan, V., Patel, M., Tanwar, S., Tyagi, S., & Kumar, N. (2020). IoT Enabled real-Time urban transport management system. Computers & Electrical Engineering, 86. doi:https://doi.org/10.1016/j.compeleceng.2020.106746.

11) Dsouza, k. B., Muhammad, S., & Hussain, Y. (2017). Smart parking — An integrated solution for an urban setting. 2nd International Conference for Convergence in Technology (I2CT), 2017. doi:10.1109/I2CT.2017.8226115

12) Fraifer, M., & Fernstrom, M. (2016). Investigation of Smart Parking Systems and their technologies. Thirty Seventh International Conference on Information Systems, IoT & Smart City Challenges and Applications. Retrieved from https://www.researchgate.net/publication/311535831_In vestigation_of_Smart_Parking_Systems_and_their_technologies.

13) Hasan, M. O., Islam, M. M., & Alsaawy, Y. (2019). Smart Parking Model based on Internet of Things (IoT) and TensorFlow. 7th International Conference on Smart Computing & Communications (ICSCC), 2019. doi:10.1109/ICSCC.2019.8843651

14) Ke, R., Zhuang, Y., Pu, Z., & Wang, Y. (2020). A Smart, Efficient, and Reliable Parking Surveillance System With Edge Artificial Intelligence on IoT Devices. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 1-13. doi:10.1109/TITS.2020.2984197

15) Khalid, M., Wang, K., Aslam, N., Cao, Y., Ahmad, N., & Khan, M. K. (2021). From smart parking towards autonomous valet parking: A survey, challenges and future Works. Journal of Network and Computer Applications, 175. doi:https://doi.org/10.1016/j.jnca.2020.102935