Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Smart Parking System Over Distributed Node Network using Computer Vision

Aswin Sreekumar.[1], H Kailash.[2], Greeshwar R S.[3], Satish Kumar L.[4]

[1]Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli, India

[2]Computer Science Engineering, National Institute of Technology, Tiruchirappalli, India

[3]Production Engineering, National Institute of Technology, Tiruchirappalli, India

[4]Electrical and Electronics Engineering, National Institute of Technology, Tiruchirappalli, India

E-mail: [1]aswinsreekumar.main@gmail.com, [2]h.kailash1501@gmail.com, [3]greeshwar02@gmail.com, [4]satishkumarVV2002@gmail.com

Abstract

The growing demand for urban parking has led to significant challenges, particularly with conventional centralized smart parking systems that often face single points of failure, limited scalability, and complex deployment processes. Existing solutions are inadequate in addressing these issues, prompting the need for a more resilient approach. This research presents a distributed, cost-effective, innovative parking system to overcome these limitations. The proposed system features a network of nodes strategically deployed across parking lots and interconnected through wireless LAN (Local Area Network), enabling real-time data access through a cloud-based website and back-end server. By utilizing low-power cameras for object detection, the system enhances operational efficiency and accuracy while minimizing costs. Additionally, the Google Maps API (Application Programming Interface) integration provides a user-friendly interface, facilitating easy navigation to available parking spaces. This solution explicitly targets the unique parking challenges in India, demonstrating promising applications for parallel parking scenarios.

DOI: https://doi.org/10.36548/jucct.2024.4.001
Received: 08.10.2024, received in revised form: 01.11.2024, accepted: 14.11.2024, published: 26.11.2024

Keywords: Smart Parking, Distributed, Wireless LAN, Object Detection, Camera, Cloud Deployment, Local Server, Google Maps, Indian Roads.

1. Introduction

Parking has always been an issue in metropolitan cities. Searching for parking spots wastes time and fuel, and drivers tend to circle the place searching for parking spots. This increases traffic congestion, pollution, and fuel waste [11-13]. This is seen on a risky scale during holidays and social gatherings. Unable to find proper parking spots, some vehicles are parked in narrow spaces, causing traffic blocks and congestion and leading to further wastage of resources [14,15]. In major Indian cities, studies estimate that parking demand can exceed available space by as much as 30-40 %. Bangalore, the silicon capital of India, tops the traffic congestion charts in the world at the second spot after London, with many other Indian metropolitan cities following up on the list [16-18]. This calls for a streamlined approach through which people can easily find and park their vehicles close to their destinations, and the solution must be directly accessible to the public [19-20].

The proposed system consists of installing low-cost camera modules in multiple parking lots across the city, which stream live images to the corresponding remote server. The remote server processes the data from the camera module and decides the number of vacant parking spaces available in the parking lot. The remote server updates the number of vacant parking slots and a number of filled parking slots to a database in the cloud.

The number of vacant parking slots and their location are displayed in a web application accessible to the general public and free to use. The database is updated continuously, ensuring a pristine user experience. The system aims to adequately inform drivers of the available parking spaces around the destination, enabling lesser traffic congestion across the city and saving time and fuel through convenience.

2. Related Work

R. A. Hadi et al. [1] and Yuan Liu et al. [2] presented a comprehensive study of existing vision-based intelligent parking systems, detailing the hardware and software components utilized in prominent approaches. This review provided insights into

advancements within the field and identifies specific limitations in various approaches that the system proposed in this research aims to solve.

Sayani Banerjee et al. [3] used CCTV (closed circuit TV) as visual input and a CNN (Convolutional Neural Network) model for classifying objects within the frame and marking vehicles for occupancy computation, which the proposed system aims to achieve at lower compute resources using pre-trained SSD (Single Shot Detection) models. M. Dixit et al. [4] proposed an intelligent parking system for indoor controlled spaces using vision and image processing. The system proposed in the current paper scales this up towards several parking spaces to be handled and routed based on users' location, enabling a higher impact toward adequate roadside parking.

Polprasert et al. [5] presented an smart parking system using a single camera placed at enough height to cover the entire parking area. However, this approach cannot efficiently scale up the system based on parking area size, which the system proposed in this study would solve using a distributed node network. R. Nithya et al. [6] detailed using a YOLOv3 (You Only Look Once) model for vehicle detection compared to other classification models. R. Lookmuang et al. [7] used cameras and ultrasonic sensors to obtain parking occupancy data for updating the application using data fusion. J. Dasari et al. [8] outlined the smart parking system application using Python and OpenCV through pre-trained object detection models. M. M. Bachtiar et al. [9] have implemented the parking system using the HAAR Cascade Classifier method for vehicle detection and highlighted the improvement in the detection speeds at comparable computational power. L. Hu and Z. Liu [10] extended the application of the system to path planning inside an indoor parking space using the A* algorithm from the entry to the parking spot and exit.

3. Proposed Solution

The proposed system design can be broken down into four segments, with coordination and data exchange between them to constitute the overall system [21].

- Distributed server (Raspberry Pi) for image processing, computation, and network management.

- ESP32-CAM hardware setup for wireless image transmission and reception by server.
- Object detection, computation of available parking slots, and updation of the database.
- Cloud Deployed and completely scalable website and cloud database management.

The major hardware components of the prototype are the camera nodes and the local processing server. As a vision-based solution, the system utilizes the camera nodes to capture images and transmit them to the local processing compute server over the intranet. The primary software components of the prototype are the object detection algorithms running on the local server, the database in the cloud, and a fully scalable website deployed on the internet. A generic bash script running in the computer links up all these software components and ensures system operations under different circumstances.

The images received from the different camera nodes, stored in the server, are processed using object detection and additional algorithms to determine the final occupancy numbers within the space, and the culminating numbers are updated within the cloud database. The website is updated in real-time from this database and displayed to the user using the website above. The site is intuitive and directs the user to the chosen parking location with the help of Google Maps API. The API also helps determine the user's location and calculate the closest available parking spots. The entire working of the system is illustrated in Figure 1.

4. The Prototype

4.1 The Camera Setup

ESP32-CAM is a low-cost ESP32-based development board with an onboard OV2640 camera module, small in size. It is an ideal solution for IoT applications as the board integrates WiFi, traditional Bluetooth, and low-power BLE with two high-performance 32-bit LX6 CPUs.

In this research, ESP32 CAMs take pictures at regular intervals and transmit them to a remote server in the same LAN. The ESP32 CAM connects to the LAN and establishes a

connection with a remote server. A frame is captured from the camera and stored in a camera DBT pointer, which holds the pixel data, height, and width of the image. The pixel data is transmitted to the server through TCP protocol. However, since the size of pixel data exceeds the size limit of a single TCP socket, pixel data is broken up into chunks and sent individually. This process repeats for approximately two seconds.

The camera nodes were fabricated using perf-boards, and header pins were soldered onto the ESP32-CAM board. The fabricated ESP32 header boards are shown in Figure 2. The mobile power supply was provided using two 3.7V 18650 Li-ion batteries. The same can be scaled up by connecting the same to the AC (Alternating Current) main power through an adapter, and the battery can serve as a power backup in case of AC power fluctuations or failure. The battery voltage was regulated down to 5V for the ESP32-CAM using an LM7805 voltage regulator. The circuit diagram of the camera node is illustrated in Figure 3.

4.2 The Server

Raspberry Pi 4B is used as a local server for image reception, processing, slot computation, and updation to the cloud. PuTTy is used to connect wirelessly to the Pi over SSH and FileZilla was used to transfer the file over SFTP (Simple File Transfer Protocol).

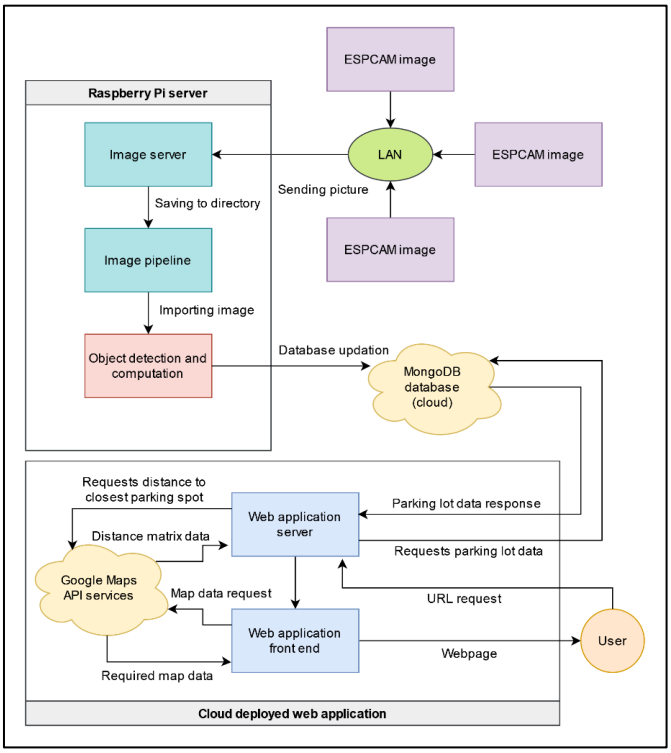

Figure 1. System Flowchart

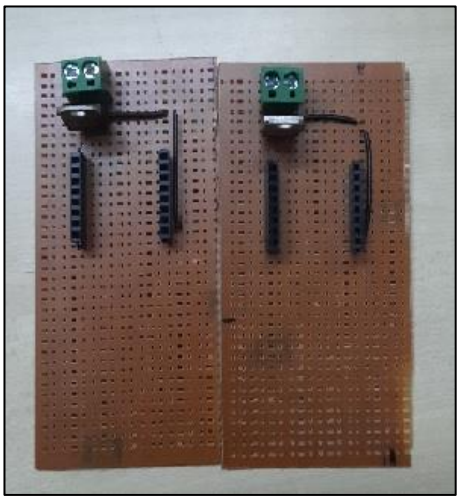

Figure 2. Fabricated ESP32 Header Board

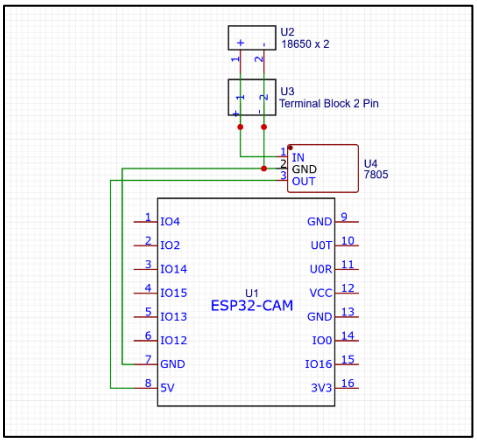

Figure 3. Fabricated ESP32 Header Board.

The ESP32 image reception and object detection Python scripts run on boot-up. This was implemented by modifying the .bashrc script to execute the Python scripts when booted-up or when a terminal is launched. The ESP32 script receives pixel data of the image as a chunk of bytes. This is stored in a byte array and is converted into a JPEG image using the Pillow library. The script then stores each image under the proper naming convention (ESP-XX-CHN-X-time) and stores it in the respective directory. The Raspberry Pi is connected to the same local network as the ESP32 CAMs through WiFi.

Internet is also enabled to perform updation to the cloud. A power bank was used to power the Raspberry Pi for testing purposes, as shown in Figure 4, and it can be switched to AC mains through an SMPS for fixed deployment.

Figure 4. Raspberry Pi Server

4.3 Object Detection and Computation

The object detection (Figure 5) script runs indefinitely, and it processes the images obtained from parking lots one after the other through a pipeline, identifying the bounding boxes of the objects (cars and bikes) utilizing a pre-trained model (MobileNet SSD model) in the OpenCV DNN (Deep Neural Network) module.

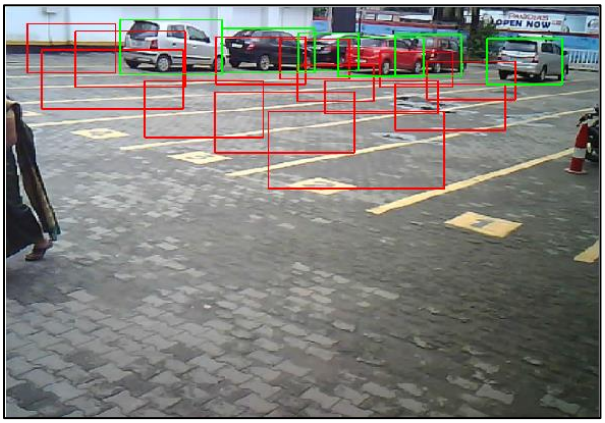

Figure 5. Object Occupancy/Vacancy Detection

The name of the image contains city and parking lot details, along with the capture timestamp. This information is used to process data from the image.

4.4 The Website

The website is developed using Node.js as the backend technology and uses multiple APIs offered by Google and in-built browser APIs. The homepage of the website is shown in

Figure 6. It is a scalable, cloud-deployed, responsive web application accessible to the general public and free to use.

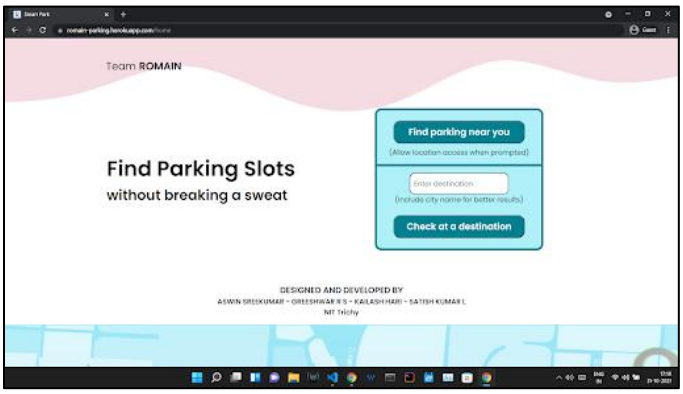

Figure 6. The Website

Upon reaching the website, a user can either choose to find parking slots around their current destination, upon which the browser will prompt the user to give access to the location. When the permission is granted, the location of the user's device is used to find the closest city in the database. Then, the coordinates are used to find the closest ten parking lots in the city. Then, these found parking locations are sorted again based on the distance to reach them using the Geomatrix API and then displayed on the map as well as on the list. This navigation feature in the website is depicted in Figure 7.

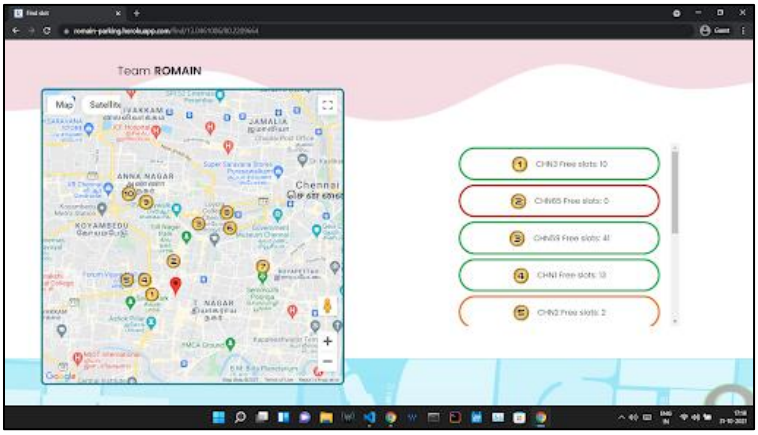

Figure 7. Website in Action

The user can click on the map marker for the parking lot or the card to open the location in Google Maps for directions. The list of cards also displays the number of free

parking slots in that particular parking location so that the user can know how many free slots to expect at that parking lot.

The database consists of the city's information and the parking lots in each location in tables. MongoDB database service has been used for this.

The application uses geolocation technology to find the device's location and prints out an interactive map with the ten closest parking lots and their availability. The user can click on the parking slot in the map, which links to Google Maps for directions to that parking lot from the user's location (in Mode 1) or from the destination (in Mode 2). It uses Express as the server technology. This web app was hosted for free on the Heroku platform for testing purposes. The website is fully responsive, and the mobile version is illustrated using Figure 8.

Figure 8. Responsive Website

5. Testing and Results

The setup was tested at a moving 30+ car parking space. Two ESP32 CAMs were mounted on a pole, and a wall diagonally to each other, as shown in Figure 9 and Figure 10, and the corresponding scripts were run at the ESP and Raspberry Pi's side. The entire setup

was interconnected over LAN through a mobile router. The images were captured at intervals of 2 seconds and transmitted to the server. Naming conventions, as described in the earlier section, were followed, and sources of images were tracked down using the local IP of ESP32-CAM.

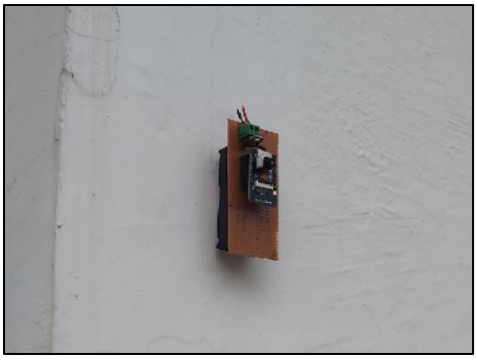

Figure 9. On-site ESP32 CAM

The server processed each image from the directory pipeline, and cars were detected using the model. Occupied parking slots were computed, and the same was updated for the MongoDB cloud after the computations. Upon resetting the ESP32-CAM, it connected to the required IP automatically and resumed the process of image capture and transmission. Rebooting the Raspberry Pi caused it to instantly connect to the LAN and run the required scripts without hassle, and it continued to update the MongoDB cloud database without errors. The final images with bounding boxes of vehicle occupancy and vacancy detection are illustrated using Figure 11 and Figure 12.

Figure 10. On-site ESP32 CAM

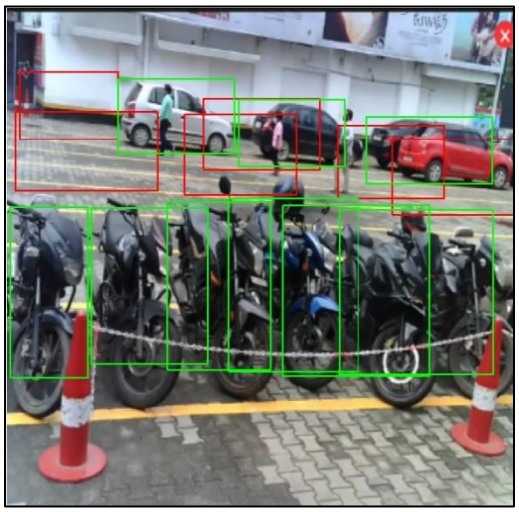

Figure 11. On-site Testing

6. Comparison to Contemporary Solutions

Many existing solutions use one or a limited number of powerful cameras placed over a large parking spot and are connected to a single system for processing. However, this approach is not suitable for roadside parking, which is the most utilized space in India. The proposed approach reduces the load on cameras and increases its number at lower costs. The decentralized server topology implemented in this solution avoids single-point failure as a set of cameras is connected to one dedicated server.

Expanding this system is simpler compared to existing systems, as the node simply consists of a Raspberry Pi and a set of ESPCAMs. This node can be easily connected to an existing framework. Web-based solution to aid cross-compatibility and easier updation/changes. This system is a power and cost-efficient solution compared to other technologies.

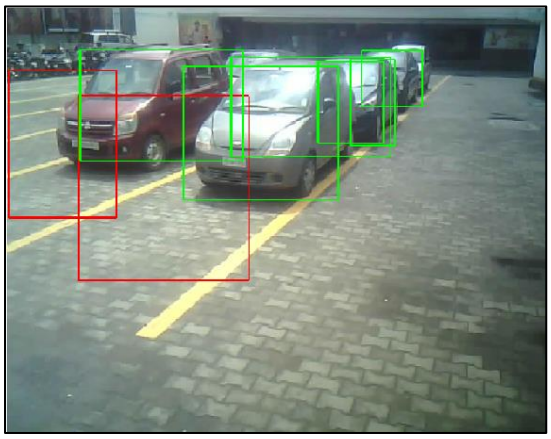

Figure 12. On-site Testing

7. Conclusion

In conclusion, this research presents a low-cost solution to a significant contributor to traffic congestion in urban areas. By deploying affordable, low-power camera modules in parking lots, the system transmits live images to a remote server that utilizes computer vision techniques to determine available parking spots and update a cloud database in real-time. A public online application provides users with convenient access to this information.

The benefits of this system are substantial. It streamlines parking management, reduces the time spent searching for available spaces, and helps alleviate traffic congestion. Additionally, it enhances the user experience by offering real-time updates, optimizing resource utilization, and contributing to more efficient urban mobility. This innovative approach has the potential to transform parking dynamics in cities, ultimately leading to improved urban transportation systems. Future research could explore the integration of this system with other smart city initiatives, further enhancing its impact on urban infrastructure and sustainability.

References

[1] Hadi, Raad Ahmed, Loay Edwar George, and Zainab Jawad Ahmed. "Computer Vision-based Approaches, Datasets, and Applications of Smart Parking Systems: A Review." Iraqi Journal of Science (2024): 979-1000.

[2] Liu, Yuan, Xiangju Liu, Shuaimao Wang, and Rongqiang Tian. "Research on parking guidance system based on computer vision." In Journal of Physics: Conference Series, vol. 2425, no. 1, IOP Publishing, 2023. 012054.

[3] Banerjee, Sayani, T. S. Ashwin, and Ram Mohana Reddy Guddeti. "Automated parking system in smart campus using computer vision technique." In TENCON 2019-2019 IEEE Region 10 Conference (TENCON)Kochi, India, IEEE, 2019. 931-935.

[4] Dixit, Madhur, C. Srimathi, Robin Doss, Seng Loke, and M. A. Saleemdurai. "Smart parking with computer vision and IoT technology." In 2020 43rd International Conference on Telecommunications and Signal Processing (TSP)Milan, Italy. IEEE, 2020. 170-174.

[5] Polprasert, Chantri, Chaiyaboon Sruayiam, Prathan Pisawongprakan, and Sirapob Teravetchakarn. "A camera-based smart parking system employing low-complexity deep learning for outdoor environments." In 2019 17th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand pp. 1-5. IEEE, 2019.

[6] Nithya, R., V. Priya, C. Sathiya Kumar, J. Dheeba, and K. 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 125, no. 4 (2022): 3205-3225.

[7] Lookmuang, Rachapol, Krit Nambut, and Sasiporn Usanavasin. "Smart parking using IoT technology." In 2018 5th International Conference on Business and Industrial research (ICBIR), Bangkok, Thailand. IEEE, 2018.1-6.

[8] Jashwanth Dasari , Shivani Vodnala , Swapna Enugala“Smart Parking System using Python and OpenCV,” International Journal for Research in Applied Science and Engineering Technology , 2023. 1976-1985

[9] Bachtiar, Mochamad Mobed, Adnan Rachmat Anom Besari, and Atikah Putri Lestari. "Parking management by means of computer vision." In 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE),Surabaya, Indonesia IEEE, 2020. 1-6.

[10] Hu, Lin, and Zhiping Liu. "Intelligent Parking lot Assistance System Based on Machine Vision and A∗ Algorithm." In 2021 3rd International Symposium on Smart and Healthy Cities (ISHC),Toronto, ON, Canada. IEEE, 2021. 28-35.

[11] Bura, Harshitha, Nathan Lin, Naveen Kumar, Sangram Malekar, Sushma Nagaraj, and Kaikai Liu. "An edge based smart parking solution using camera networks and deep learning." In 2018 IEEE international conference on cognitive computing (ICCC),San Francisco, CA, USA. IEEE, 2018. 17-24.

[12] Prabagar, Ajanthwin, N. Sri Madhavaraja, S. Arunmozhi, and K. Suresh Manic. "Artificial Vision Based Smart Urban Parking System." In 2021 International Conference on System, Computation, Automation and Networking (ICSCAN),Puducherry, India. IEEE, 2021. 1-4.

[13] Al-Absi, Hamada RH, Justin Dinesh Daniel Devaraj, Patrick Sebastian, and Yap Vooi Voon. "Vision-based automated parking system." In 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), Kuala Lumpur, Malaysia. IEEE, 2010. 757-760.

[14] Budihală, Bogdan, Todor Ivașcu, and Sebastian Ștefănigă. "Motorage-Computer vision-based self-sufficient smart parking system." In 2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC),Linz, Austria. IEEE, 2022. 250-257

[15] Sieck, Noah, Cameron Calpin, and Mohammad Almalag. "Machine vision smart parking using internet of things (IoTs) in a smart university." In 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops),Austin, TX, USA. IEEE, 2020. 1-6.

[16] Zhang, Wenjin, Jason Yan, and Cui Yu. "Smart parking system based on convolutional neural network models." In 2019 6th International Conference on Information Science and Control Engineering (ICISCE),Shanghai, China. IEEE, 2019. 561-566.

[17] Kiruthika, S., P. Sakthi, K. Raam, G. Sanjay, and M. Mohamed Rafi. "Vision Based Smart Parking System." Turkish Journal of Physiotherapy And Rehabilitation 32 (2021).

[18] Elomiya, Akram, Jiří Křupka, and Stefan Jovčić. "A Smart Parking System Using Surveillance Cameras and Fuzzy Logic: A Case Study at Pardubice University's Campus." Procedia Computer Science 225 (2023): 4881-4890.

[19] Archana, M., S. Shanmuga Raju, D. Preethi, S. K. Mydhili, and U. Vinothkumar. "Smart Automated Parking System using IoT." In 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India. IEEE, 2023. 1580-1585.

[20] Izudheen, Sminu, P. V. Gokul, Jesbin Joseph, Karthik S. Nair, and Manu Krishnan NS. "IoT based smart parking system." In 2023 International Conference on Control, Communication and Computing (ICCC), Thiruvananthapuram, India. IEEE, 2023.1-6.

[21] https://rmi.nitt.edu/project/26