SMART PARKING SYSTEM BASED ON COMPUTER VISION TECHNIQUES

[1]G. Kalyani , [2]K. Krishna Jyothi, [3]M. Likhitha

Department of CSE, Geethanjali College of Engineering and Technology, Hyderabad, India

***Abstract*— Smart parking systems are generally IoT based and use sensors to detect available slots in that area. This system can cost a lot of money. These systems are usually automated, but they require regular maintenance to ensure everything is working smoothly. Instead of using sensors for every slot of the parking lot, we just need to install a camera that can capture several slots at a time. Cameras can be installed in any urban area without creating traffic problems. The installation can be done at any time since the device becomes operational in a few minutes without affecting normal parking activities. Once the cameras are functional, we collect the live parking footage to perform cv2 methods and calculate the pixel count, where the change in the pixel count can define if the slot is occupied or not. The results are then uploaded to a cloud database and can be sent to drivers via a web or mobile application in real-time.**

Keywords— Computer Vision, OpenCV(cv2), Pixels, Cloud database, Real-Time System

I. INTRODUCTION

Smart parking system refers to innovative technologies providing solutions in the parking industry. The main idea behind the parking system is self-explanatory, it is a system that helps people and organizations to manage their parking spaces.

Managing car parks isn't an easy task for companies and organizations because there are moving parts, including traffic and the availability of spaces. We might need to pay for high usage or access fees, software maintenance fees, etc. It is a time-consuming task, requires human labor, and is inefficient.

The Systems are usually automated, but they require regular maintenance to ensure everything is working smoothly. This means ensuring that everything works properly, such as updating portions of code or optimizing tasks for quickness and efficiency, etc. Regular maintenance of parking systems requires money and time. If the system malfunctions, it could lead vehicles to park in the wrong places, which could contradict the whole idea of setting up a parking system for easy parking.

They also need security to monitor all the sensors and other devices, which add up to additional cost. An efficient Smart parking system is a system that can help in reducing businesses' overhead on parking and also can reduce the impact of their parking space on their local community.

The rest of the paper is organized as: Section II portrays various methods proposed for smart parking systems. Section III depicts a short description of the proposed accident detection framework. Further, Section IV addresses the methods used in this model from open cv. The proposed architecture of the model is briefed in Section V. The results are discussed in Section VI, and the conclusion is elucidated in Section VII.

II. RELATED WORKS

A. *Intelligent smart parking algorithm.*

In 2017, Dharmini Kanteti et al. [10] presented an algorithm for smart parking and proposed the use of IoT devices for processing vehicle detection, software working OCR, Arduino as microcontroller and Raspberry Pi to interface all the components. The paper focuses on better placement of vehicles in the parking spots using the algorithm but in case all the parking spots are full then the user won't be able to park the car after arriving at the parking lot. Also, the amount of hardware used in the proposed idea considerably increases the installation expense.

B. *An IoT Based Smart Parking System Using LoRa.*

In 2018, Ravi Kishore Kodali et al. [13] proposed the use of the LoRa Esp32 module for transmitting the status of parking spots to the Wi-Fi. The proposed idea in the paper only provides information about the parking spots but does not involve security features in case the vehicle gets stolen. Moreover, IBM Watson is not so cost-effective when it comes to scaling the applications to the industrial level.

C. *Implementation of Smart Parking Solution by Image Analysis.*

In 2018, Aleksejs Zacepins et al. [15] presented a paper on A smart parking system based on video processing and analysis on a python program with real-time parking lot monitoring using a machine learning technique. The method used by the system was focused on video analysis and public video streaming.

D. *Detecting Efficient Parking Space Using Smart Parking*

In 2018, Ajay Zajam et al. [16] proposed an algorithm, which will identify best efficient route between user and parking place based on the real time traffic. The proposed algorithm helps user to identify parking place with least cost based on the distance and time needs to reach to parking place. The experiment result shows that proposed algorithm improves the user waiting time for identification of parking place with minimum cost.

E. *Estimating the Occupancy Status of Parking Areas by Counting Cars and Non-empty Stalls*

In 2019, Di Mauro et al. [20] proposed and compared different vision-based approaches to estimate the occupancy status of parking areas by counting cars and non-empty parking stalls. He considered both the scenario in which parking stalls are marked on the ground and the more challenging one in which no assumption on the presence or position of stalls is assumed. He developed a solution based on image classification, vehicle detection and semantic segmentation.

III. SHORT DESCRIPTION ON ACCIDENT DETECTION FRAMEWORK

We proposed a system where we use cameras for parking management and security of the parking lot. We get the data from the camera in real-time to detect the availability of parking slots. To get results with the most accuracy, the coordinates for where the occupancy detection must be made need to be given before analyzing the real-time camera feed. Once we have the coordinates, we can use cv2 methods like, COLOR_BGR2GRAY, GaussianBlur, adaptiveThreshold, medianBlur, dilate, np. Ones, etc., to calculate the pixel count of the given coordinates, with and without the vehicle in the parking slot. After analyzing the pixel count, we can now identify the availability in the parking lot and upload the results to the cloud in real time. The data from the cloud can be used to build customized web-based or mobile-based applications according to the organization which uses the parking system.

A. *Video/ Image Pre-processing:*

This phase is responsible for processing video data within the system. Its main task is to read the video data and extract individual image frames from the video. In the context of slot detection, this module plays a crucial role as it allows the subsequent modules to analyze each frame for the occurrence of an accident. Video data can be encoded in different formats or configurations, and for the system to function properly, it requires homogeneous data in a consistent format and configuration. The colour conversion module addresses this issue by converting the video data to the RGB format. RGB (Red, Green, and Blue) is a commonly used colour model in digital imaging where each pixel is represented by the intensities of these three primary colors.

B. *Pixel counting using cv2*

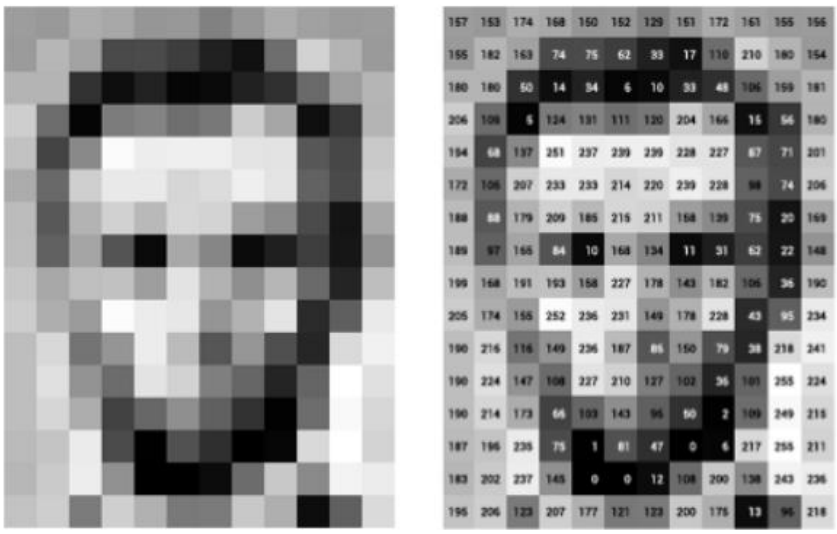


Fig 1: Pixel counting with cv2

we use pixel counting method with open cv which can used to differentiate a plane surface that is the empty slot with the 3d surface that is the occupied slot with vehicles. We use open CV as it is the most useful library for image processing. Open cv is used in python and we just need to use the built in function or keywords. OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

In this model we first collect the real time footage from camera and for each frame of the video we perform cv2 methods. First, we convert the area of the given coordinates to gray scale using BG2GRAY method. Then we apply GaussianBlur to soften the high frequency edges of the images. The image frame is further softened using medianBlur to get the count of zeros and one in the image. Then we calculate the total number of ones according to which we set a threshold value.

C. *Real time cloud database*

We use a real time cloud database to get details of real time availability of parking slots. The data from this cloud data base can be used for building customized websites or applications.

IV. METHODS IN OPEN CV(CV2)

The different cv2 methods and how they are used in the model are described below

1. cv2.resize(): To resize an image
2. cv2.cvtColor(): Used to convert an image from one color space to another.
3. cv2.GaussianBlur(): Sharp edges in images are smoothed while minimizing too much blurring.
4. cv2.adaptiveThreshold(): The threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions.
5. cv2.medianBlur(): Used to remove noise from an image or signal, under certain conditions, it preserves edges while removing noise.
6. cv2.dilate(): Used to accentuate features and increase the object area.
7. cv2.countNonZero(): Returns the number of nonzero pixels in the array matrix, zero pixels denote black color.

V. PROPOSED ARCHITECTURE

A. *System Overview*

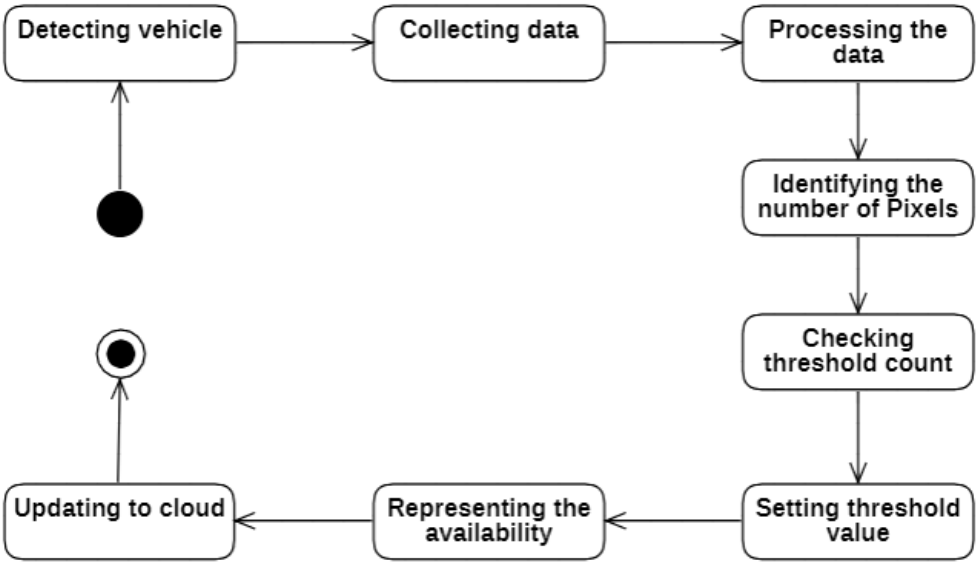


Fig 2: System Architecture

In Fig. 2 a total of 8 processes are mentioned on how exactly the proposed system will work.
Process 1: The arrival of vehicle is checked using sensor and is allowed into the parking lot.
Process 2: Collecting the yt file in which we have stored the data. We have coordinates for all the slots in the given parking lot.
Process 3: Taking the real time footage of parking lot we draw boundaries in shape of square for all the coordinates present in the data file.
Process 4: For the given boundaries we convert them into gray scale and then into zeros and ones pixels to get the pixel count for total number of white pixels.
Process 5: Running the model and find the standard pixel count for empty slots of parking lot.
Process 6: Checking the threshold value we give condition that if the pixel count is less than threshold value then the slot is occupied or it is not occupied.
Process 7: *We represent the occupied slots in red boxes and unoccupied slots in green boxes which informs us that the green boxes are the slots free for parking our vehicle.*

Process 8: *The identified results are uploaded to cloud data base in real time. The data from this cloud data base can be used for building customized websites or applications.*

B. Required Technologies and Techniques

- Industrial grade Ultrasonic Sensors: These proximity sensors should be able to withstand the harsh climate and should be durable and long lasting.
- CCTV cameras: The CCTV cameras will be used to check if there is actually a car at the parking spot or not.

C. System Hardware Architecture

There are two sets of hardware which are used for the proposed idea..

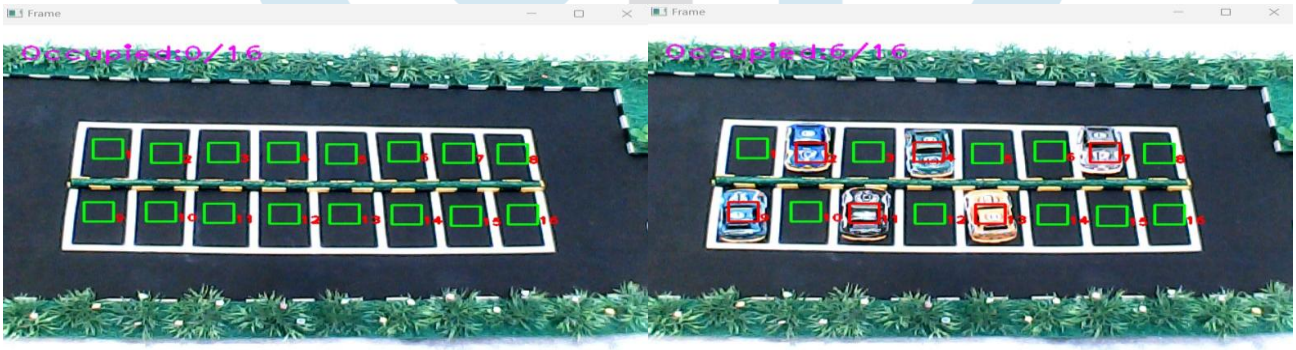
- IoT Device: The IoT device which is planted at the entrance the parking spot has two steps of working. First step is to use the ultrasonic sensor to determine if there is a car in front of the IoT system or not. The second step is to open the entrance to allow one vehicle at a time.
- CCTV Cameras: The CCTV cameras which are deployed at the parking lots for security purposes can be used to pass the video feed to a server which converts the video stream into bytes which is then processed and the cloud database is updated accordingly after processing the video taken by the CCTV.

VI. RESULTS AND DISCUSSION

The model is executed in real time and is able to give results accurately. This model doesn't use any kind of data set or complex algorithm so it runs faster than the traditional parking system. It is simple to build and execute this model in basic processors which can support python.

This model is able to take the coordinates from user and store it in a .yt file which stores volumetric data. The model can then perform cv2 methods on each coordinate of a single frame. After calculating the pixel count, we are able to set a threshold value and run the model.

The model successfully identifies the occupied and un occupied slots based on difference of pixel count value from threshold value. The standard pixel count represents empty slot and pixel count less than threshold value represents the occupied slot.



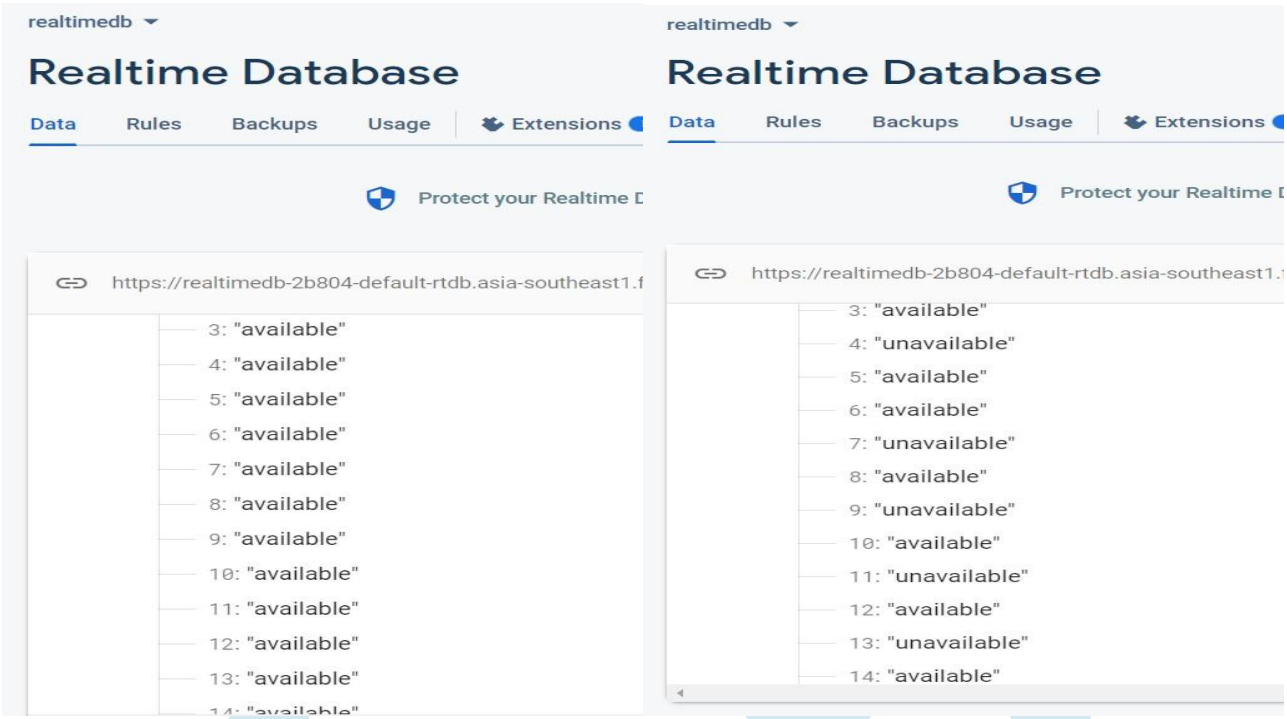


Fig 3: Smart parking prototype tested using the model

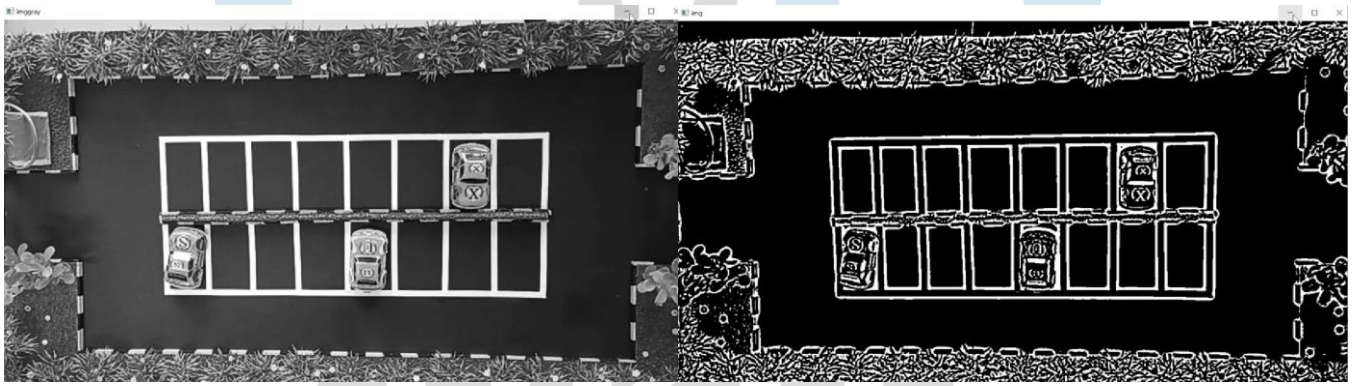


Fig 4: Conversion of frames to calculate pixel count

VII. CONCLUSION

The proposed system was designed and tested. The proposed system can detect the availability of free parking spot through a real time camera feed. The proposed system collects real time data using camera for processing the parking slot information. The coordinates of parking slot are given first, then the availability is detected. Using the coordinates, we detect the change in the pixels and know if the parking slots are available or not. Although there are a number of existing systems, the proposed system provides a unique feature by using camera instead of regular sensors. This system helps in reducing the cost, manpower, and implementation that which are used in a sensor-based parking system.

References

[1] Yusnita, R., Norbaya, F. & Basharuddin, N. (2012). Intelligent Parking Space Detection System Based on Image Processing. International Journal of Innovation, Management and Technology, 3(3), 232-235.

[2] Almeida, P., Oliveira, L. S., Silva Jr., E., Britto Jr., A. & Koerich, A. (2013). Parking Space Detection using Textual Descriptors. In 2013 IEEE International Conference on Systems, Man, and Cybernetics, (pp. 3603- 3608). Manchester, UK: IEEE.

[3] Choeychuen, K. (2013). Automatic Parking Lot Mapping for Available Parking Space Detection. In 5th International Conference on Knowledge and Smart Technology (KST), (pp. 117-121). Chonburi, Thailand: IEEE.

[4] Polycarpou, E., Lambrinos, L., & Protopapadakis, E. (2013). Smart Parking Solutions for Urban Areas. In 2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), (pp. 1-6). Madrid, Spain: IEEE.
[5] Al-Kharusi, H., & Al-Bahadly, I. (2014). Intelligent Parking Management System Based on Image Processing. World Journal of Engineering and Technology, 2(2), 55-67
[6] Moses, N., & Chincholkar, Y. D. (2016). Smart Parking System for Monitoring Vacant Parking. International Journal of Advanced Research in Computer and Communication Engineering, 5(6), 717-720.
[7] Amato, G., Carrara, F., Falchi, F., Gennaro, C., & Vairo, C. (2016). Car Parking Occupancy Detection Using Smart Camera Networks and Deep Learning. In 2016 IEEE Symposium on Computers and Communication (ISCC), (pp. 1212-1217). Mersina, Italy: IEEE.
[8] Lin, T., Rivano, H., & Le Mouël, F. (2017). A Survey of Smart Parking Solutions. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229-3253
[9] Fraifer, M., & Fernström, M. (2017). Designing a Smart Car Parking System (PoC) Prototype Utilizing CCTV Nodes: A Vision of an IoT Parking System via UCD Process. Advances in Science, Technology and Engineering Systems Journal, 2(3), 755-764.
[10] Kanteti, D., Srikar, D. V. S., & Ramesh, T. K. (2017). Intelligent smart parking algorithm. 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon).
[11] Enrìquez, F., Soria, L. M., & Álvarez-Garcìa, J. A. (2017). Existing Approaches to Smart Parking: An Overview. In International Conference on Smart Cities (Smart-CT 2017), LNCS 10268, (pp. 63–74). Malaga, Spain: Springer
[12] Kommey, B., Addo, E. O., & Agbemenu, A. S. (2018). A Smart Image Processing-based System for Parking Space Vacancy Management. International Journal of Computer Applications, 182(5), 1-6.
[13] Kodali, R. K., Borra, K. Y., G. N., S. S., & Domma, H. J. (2018). An IoT Based Smart Parking System Using LoRa. 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC).
[14] Paidi, V., Fleyeh, H., Håkansson, J., & Nyberg, R. G. (2018). Smart Parking Tools Suitability for Open Parking Lots: A Review. In 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018) (pp. 600-609). Madeira, Portugal: SciTePress.
[15] Zacepins, A., Komasilovs, V. & Kviesis, A. (2018). Implementation of Smart Parking Solution by Image Analysis. In 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018) (pp. 666-669). Madeira, Portugal: SciTePress.
[16] Zajam, A. emergency, & Dholay, S. (2018). Detecting Efficient Parking Space Using Smart Parking. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (pp.1-7). Bangalore, India: IEEE.
[17] Hilmani, A., Maizate, A., & Hassouni, L. (2018). Designing and Managing Smart Parking System Using Wireless Sensor Networks. Journal of Sensor and Actuator Networks, 7(2), 1-20.
[18] Zajam, A., & Dholay, S. (2018). Detecting Efficient Parking Space Using Smart Parking. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (pp.1-7). Bangalore, India: IEEE.
[19] Loong, D. N. C., Isaak, S., & Yusof, Y. (2019). Machine Vision-based Smart Parking System Using Internet of Things. TELKOMNIKA, 17(4), 2098-2106.
[20] Di Mauro, D., Furnari, A., Patanè, G., Battiato, S., & Farinella, G. M. (2019). Estimating the Occupancy Status of Parking Areas by Counting Cars and Non-empty Stalls. Journal of Visual Communication and Image Representation, 62, 234–244.
[21] Hakim, I. M., Christover, D., & Marinda, A. M. J. (2019). Implementation of an Image Processing Based Smart Parking System Using Haar-Cascade Method. In 2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE), (pp. 222-227). Malaysia: IEEE.
[22] Trivedi, J. D., Devi, M. S., & Dave. D. H. (2020). Different Modules for Car Parking System Demonstrated Using Hough Transform for Smart City Development. In the International Conference on Intelligent Manufacturing and Energy Sustainability (ICIMES 2019), (pp. 109-121). Hyderabad, India: Springer.