Automated Parking Detection System

Onkar Kanse[1], Atharva Mestry[2], Riddhi Wakde[3], Prof. Jaychand Upadhyay[4]
Dept. of Information Technology, Xavier Institute of Engineering, Mumbai

Abstract: Automated parking analysis is an emerging research area in computer vision that aims to develop efficient parking solutions for modern urban environments. This paper proposes a methodology for automated parking analysis using OpenCV. The proposed methodology involves segmenting the parking lot into rectangular blocks to specify parking spots, acquiring live footage from a top-down view camera of the parking lot, and employing image processing techniques to count the number of foreground and background pixels to determine the occupancy status of each parking spot. To mitigate the influence of environmental factors, a threshold value of 900 pixels is used to differentiate between foreground and background pixels. If the count of foreground pixels in a block exceeds 900, the spot is considered occupied; otherwise, if the count of foreground pixels is less than 900, the spot is considered vacant. This system displays the number of available parking spots and total parking spots on a screen.

Index Terms/ Keywords:- Image Processing, OpenCV, GreyScale, Segmentation, Dilation.

I.INTRODUCTION

Automated parking systems have become increasingly popular in recent years, as they offer a convenient and efficient solution to the growing problem of parking space scarcity in urban areas. One of the critical components of an automated parking system is the detection of parking spots and the determination of their occupancy status. This technical paper presents an automated parking detection system using OpenCV, a popular computer vision library that provides tools for image processing, object detection, and machine learning.

The proposed system uses a bird-eye view camera to capture real-time video footage of the parking area, and the OpenCV library is used to perform various image processing techniques to identify and track vehicles in the parking lot. This paper discusses the various steps involved in developing the automated parking detection system, including image preprocessing, segmentation, grey-scaling, and counting the number of pixels. The proposed automated parking detection system using OpenCV offers several benefits, including increased accuracy, reduced false positives, and improved efficiency in detecting and displaying parked vehicles.

The paper concludes by discussing potential future research directions in this area and highlighting the system's practical applications in real-world scenarios.

II.RESEARCH GAP AND MOTIVATION

Traditional methods for automated parking systems often rely on ultrasonic sensors or magnetic sensors to detect the presence of a vehicle. However, these methods are susceptible to interference from other nearby sensors or environmental factors such as rain, snow, or dirt. As a result, they may not be as accurate or reliable as computer vision-based methods.

On the other hand, OpenCV-based automated parking systems use cameras to capture images of the parking lot, and advanced algorithms are used to analyze these images and detect the presence of parked vehicles. OpenCV's robust image processing capabilities, such as feature detection, object recognition, and edge detection, allow for accurate and efficient analysis of images captured by cameras. Additionally, OpenCV's real-time performance optimizations, such as parallel processing and hardware acceleration, enable automated parking systems to process large amounts of image data in real-time.

The literature survey also suggests image processing algorithms such as Haar Cascade, YoloV3, SVM, etc. However, Compared to other image processing algorithms, OpenCV offers several unique advantages, such as its vast range of features and algorithms, high processing speed, and ease of use. Its vast range of features and algorithms, including object detection, tracking, segmentation, and recognition, enable developers to implement complex image processing tasks with ease. Moreover, it is optimized to perform image processing operations in real time, enabling it to process large amounts of image data at high speeds. It is a highly effective and reliable solution for image processing applications, offering unparalleled accuracy, speed, and flexibility that set it apart from other image processing algorithms.

III.RANGE OF APPLICATIONS

An automated parking detection system using OpenCV has several potential applications beyond parking systems. Its ability to analyze video footage in real-time and detect and track objects, such as people and vehicles, makes it suitable for use in surveillance and security applications. For example, an automated parking detection system can be used to monitor the entrance and exit of vehicles from high-security areas or sensitive locations such as airports, government buildings, or military bases.

In addition, it is suitable for use in traffic management applications. By analyzing the flow of vehicles and pedestrians, these systems can provide real-time data on traffic patterns, congestion, and road safety issues, which can help to optimize traffic management and reduce the risk of accidents.

Furthermore, the integration of OpenCV-based automated parking detection systems into robotics can provide visual perception and object recognition capabilities to robots, enabling them to navigate and interact with their surroundings autonomously.

In healthcare applications, these systems can be used to monitor patient movements and track patients entering and exiting hospital rooms, contributing to infection control efforts.

OpenCV-based automated parking detection systems provide a cost-effective solution for various computer vision and image processing tasks in a wide range of industries, from security and traffic management to retail and healthcare.

IV.RELATED WORK

Engineers have made use of various methods to make an automated parking system with image processing, sensor network, and CNN has been made. The main work includes the following steps: image acquisition, image enhancement, morphological processing, and representation and detection. The algorithm returns a count of the closed boxes deduced from the line marking on the ground using a sequence of image processing steps emphasizing morphological transformations.[1]

In another methodology, the program was implemented in MATLAB. The data used in the study were taken using a CCTV camera with a height of 2.5 meters, the method used was Haar Cascade Classifier and YOLOv3 with their respective accuracy. The test was carried out using ten different scenarios. The highest accuracy obtained in this study was 96.88% using YOLOv3 with a probability of 90%. In contrast, the accuracy obtained by using the Haar Cascade Classifier is 63.34%. [2]

In a paper by D. Delibaltov, W. Wu, R. P. Loce, and E. A. Bernal, “Parking lot occupancy determination from lamp-post camera images”. The 3-D geometry of the parking spaces is inferred, and the occupancy is determined by using a vehicle detector. They have shown that the system is able to correctly identify occupied parking spaces on three different datasets captured at widely varying conditions. The result showed more accurate results during fewer traffic hours[3].

Further another attempt of making a Parking Detection where the authors tested the effect of our parking space detection system through the following three standards False rate, Undetected rate, and False alarm rate. The output is that night-time images can reach an ideal result because of fewer environmental changes than that of the day due to fewer disturbances[12].

A paper from S. Kumar, E. Thomas, and A. Horo, “Identifying Parking Lots from Satellite Images using Transfer Learning” presents a conventional preprocessing of images followed by each slot identification using the Convolutional Neural network model. The highest accuracy the model could achieve was 97.21%. With this accuracy, the model performed considerably well on the random test satellite images[4].

E. Karbab, D. Djenouri, S. Boulkaboul, and A. Bagula, "Car park management with networked wireless sensors and active RFID proposed about CPF(Car Parking Framework), a novel smart car park framework that uses the integration of both wireless sensor network (WSN) and Radio Frequency identification (RFID) technology has been proposed.

The framework is scalable and provides advanced services. The results show that the clustering of a bunch of spots with a single mote considerably reduces the cost, as well as the energy consumption[5].

Yet another approach proposed by Amato G., Carrara, F., Falchi, F., Gennaro, C., and Vairo, C., in "Car parking occupancy detection using smart camera networks and deep learning". It proposed and evaluated an approach for parking status detection, which uses Convolutional Neural Networks to classify the parking space occupancy directly on board of a Raspberry Pi camera where each camera is able to cover a large portion of a parking lot by monitoring simultaneously about 50 parking spaces[6].

V.METHODOLOGY

The main flow of the framework is shown in Fig 1. Input data(live videos) were acquired from the top view of the parking arena, from ten feet heightened camera. Parking spaces in the video are segmented into frames. Then from each segment a keyframe is extracted and further processing is applied to this keyframe, to reduce computational complexity. The system automatically generates virtual parking lines keeping in view the size of the vehicle. The maximum capacity of parking slots in our training model is sixty-nine. A unique numeric label is assigned to each parking.

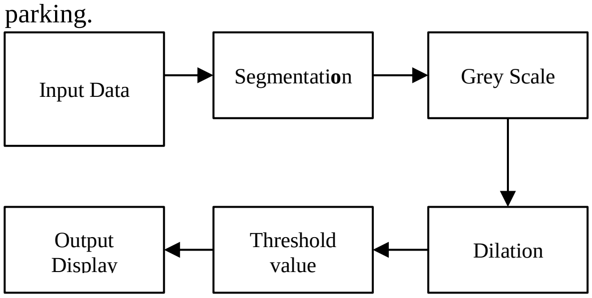

Fig 1: Implementation Process

The Livestream video of the parking lot from the camera when a car enters or leaves the parking lot is processed. RGB input is converted to monotones through grayscaling. Each block is converted to binary and then inverse binary to get the car in white colour and the parking area into black colour. Dilation is performed to classify pixels. Further, the number of pixels is counted in each block. a threshold value is set for the number of white pixels. It is calculated in every block to detect whether that block contains a car or not. If the value is less than the threshold value then that block is free and available for parking cars and if the value is greater then the block is occupied.

With open CV and several other libraries live CV2, pickle, CV zone, NumPy and a camera it is possible to track the parking spaces. A bird-eye view camera is capturing the parking space. Parking spaces are first demarcated into rectangular slots, each rectangular slot is provided a unique id. The captured images are in RGB format which are grey scaled to black and white images. In each image black and white pixel points are counted. Pixel points range from 000 – black to 255 – white.

A higher number of white pixel points indicates the presence of a car in a particular rectangular parking spot. Thus, indicating that the space is occupied and vice versa. This system also displays logos to indicate the status of parking slots, it displays FREE in a particular slot if the number of black pixel points is more and displays PARKED if the detection of white pixel points is greater.

A. IMPLEMENTATION PROCESS

The methodology proposed in this paper is --- firstly, live footage is acquired from a top-down view camera of the parking lot. The camera is positioned such that it provides a clear view of all the parking spots in the lot. The footage is pre-processed to remove any noise or artifacts that may affect the accuracy of the detection algorithm.

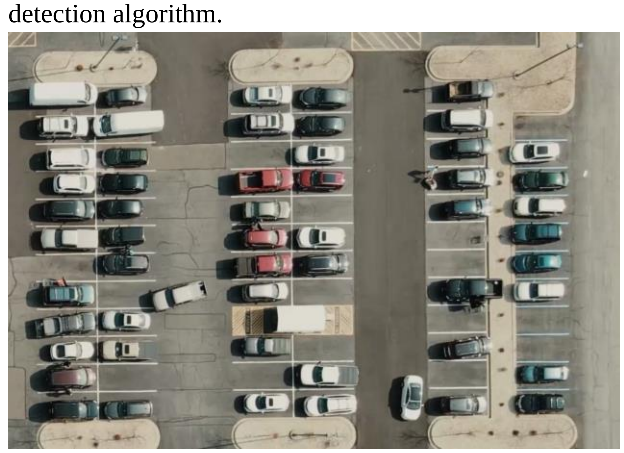

Fig 2: Parking spaces

Second, the parking lot is segmented into rectangular blocks to specify parking spots. The size of the blocks can be customized based on the size of the parking lot and the size of the vehicles that will be parked in the lot. The blocks are arranged in rows and columns to cover the entire parking lot.

Fig 3: Segmentation of parking spaces

Third, the pre-processed footage is transformed into grayscale to convert into monotones. Thus a binary product is obtained. One important reason for grey scaling is that by converting to grayscale, it separates the luminance plane from the chrominance planes. Luminance is also more critical for distinguishing visual features in an image. This prevents any ambiguity in detection due to any presence of shadows. thresholded to segment the foreground and background pixels. A threshold value of 900 is used to differentiate between foreground and background pixels. If the count of foreground pixels in a block exceeds 900, the spot is considered occupied; otherwise, if the count of foreground pixels is less than 900, the site is considered vacant.

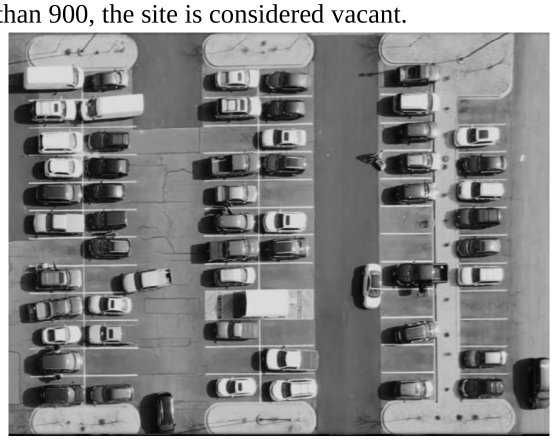

Fig 4: GreyScale image

Further, dilation is performed on the live footage. Dilation of an image is the process by which the object area in the image is increased. This process is used to accentuate features in the image. It increases the white region in the image or the size of the foreground object increases. This process helps in the classification of pixels black and white pixels.

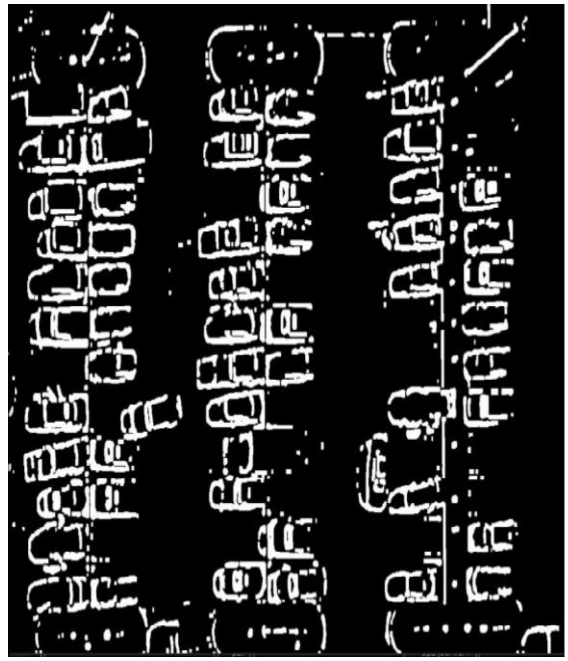

Fig 5: Dilation of the input data

Lastly, after the classification of pixels, the number of white pixels is counted. A threshold is set to segment the foreground and background pixels. A threshold value of 900 pixels is efficiently set to differentiate between foreground and background pixels. If the count of foreground pixels in a block exceeds 900 pixels, the spot is considered to be occupied; otherwise, if the count of foreground pixels is less than 900, the site is considered vacant and is available for parking. After detecting the occupancy status of the parking lot. It displays the number of available parking spots to the total parking spots on a screen. Available spaces are marked in green and the occupied ones in red.

Fig 6. Representation of parking lot after detection

B. GRAPHICAL USER INTERFACE

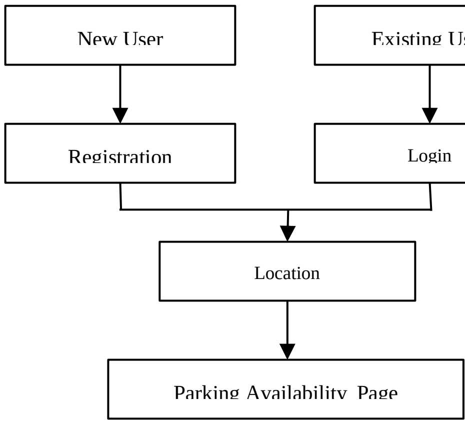

Fig 7. Flow chart of the GUI

An app is developed using React Native to allow users to access this information on their mobile devices. This app features a registration and login system for the users. Once a user logs in, he/she is redirected to a parking lot of his/her chosen location. Multiple locations are available in the data set for this proposed system. The database for this app is created using MongoDB. There are two data collections, one for the user and one for the vehicles. It is created using the Key-value pair. The app provides a real-time update of the occupancy status of each parking spot, allowing users to find available parking spots easily and quickly.

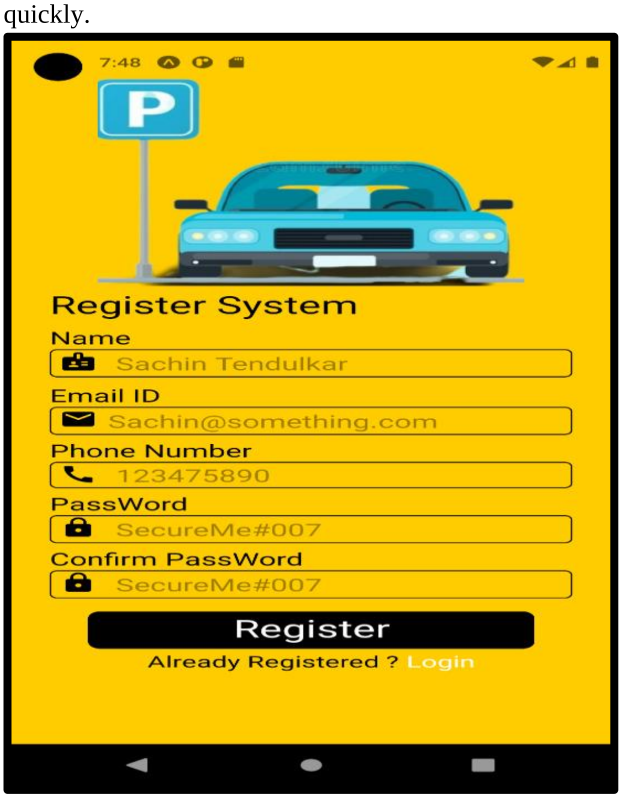

Fig 8. Registration page

Fig 9. Login page of the application

Fig 10. Parking Availability Display Page for a Location.

EVALUATION AND RESULTS

The proposed system demonstrated robustness in handling challenging scenarios such as occlusions, shadows, and partial obstructions, which can often occur in real-world parking lots. The system was also capable of detecting vehicles of different sizes, shapes, and colors, making it versatile for use in various parking lot environments.

The system was evaluated in various scenarios to determine its performance. The results showed that the system was able to detect and track vehicles in real time with a high degree of accuracy. The system was able to allocate parking spots to vehicles in a timely manner and to ensure that no parking spots were occupied by more than one vehicle.

The system's performance was also evaluated concerning processing speed and memory usage. The results showed that the proposed system had a fast processing speed, with an average processing time of 0.15 seconds, and low memory usage, which is important for real-time applications.

CONCLUSION

Automated parking analysis using OpenCV is an efficient and accurate method for detecting the occupancy status of parking spots in a parking lot. The proposed methodology involves segmenting the parking lot into rectangular blocks, acquiring live footage from a top-down view camera, and applying image processing techniques to count the number of foreground and background pixels to determine the occupancy status of each parking spot. The use of a threshold value to differentiate between foreground and background pixels mitigates the influence of environmental factors, resulting in an accurate detection algorithm. The developed app provides real-time updates of the occupancy status of each parking spot, enabling users to find available parking spots quickly and easily.

REFERENCES

[1] M. I. Ashqer and M. Bikdash, "Parking Lot Space Detection Based On Image Processing," 2019 SoutheastCon, 2019, pp. 1-6, doi: 10.1109/SoutheastCon42311.2019.9020584.

[2] Z. Bin, J. Dalin, W. Fang and W. Tingting, "A design of parking space detector based on video image," 2009 9th International Conference on Electronic Measurement & Instruments, 2009, pp. 2-253-2-256, doi: 10.1109/ICEMI.2009.5274579.

[3] D. Delibaltov, W. Wu, R. P. Loce and E. A. Bernal, "Parking lot occupancy determination from lamp-post camera images," 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, pp. 2387-2392, doi: 10.1109/ITSC.2013.6728584.

[4] S. Kumar, E. Thomas and A. Horo, "Identifying Parking Lots from Satellite Images using Transfer Learning," 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2019, pp. 1-8, doi: 10.1109/ICICT46931.2019.8977634

[5] E. Karbab, D. Djenouri, S. Boulkaboul and A. Bagula, "Car park management with networked wireless sensors and active RFID," 2015 IEEE International Conference on Electro/Information Technology (EIT), 2015, pp. 373-378, doi: 10.1109/EIT.2015.7293372

[6] Amato G., Carrara, F., Falchi, F., Gennaro, C. and Vairo, C., 2016, June. Car parking occupancy detection using smart camera networks and deep learning. In 2016 IEEE Symposium on Computers and Communication (ISCC) (pp. 1212-1217). IEEE

[7] J. K. Suhr and H. G. Jung, "Sensor Fusion-Based Vacant Parking Slot Detection and Tracking," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, pp. 21-36, Feb. 2014, doi: 10.1109/TITS.2013.2272100

[8] J. Yang, J. Portilla and T. Riesgo, "Smart parking service based on Wireless Sensor Networks," IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, 2012, pp. 6029-6034, doi: 10.1109/IECON.2012.6389096

[9] R. Salpietro, L. Bedogni, M. Di Felice and L. Bononi, "Park Here! a smart parking system based on smartphones' embedded sensors and short range Communication Technologies," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015, pp. 18-23, doi: 10.1109/WF-IoT.2015.7389020

[10] M. Veres and M. Moussa, "Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 8, pp. 3152-3168, Aug. 2020, doi: 10.1109/TITS.2019.2929020

[11] B. Sairam, A. Agrawal, G. Krishna and S. P. Sahu, "Automated Vehicle Parking Slot Detection System Using Deep Learning," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 750-755, doi: 10.1109/ICCMC48092.2020.ICCMC-000140

[12] H. Bura, N. Lin, N. Kumar, S. Malekar, S. Nagaraj and K. Liu, "An Edge Based Smart Parking Solution Using Camera Networks and Deep

Learning," 2018 IEEE International Conference on Cognitive Computing (ICCC), 2018, pp. 17-24, doi: 10.1109/ICCC.2018.00010

[13] D. J. Bonde, R. S. Shende, A. S. Kedari, K. S. Gaikwad and A. U. Bhokre, "Automated car parking system commanded by Android application," 2014 International Conference on Computer Communication and Informatics, 2014, pp. 1-4, doi: 10.1109/ICCCI.2014.6921729

[14] H. R. H. Al-Absi, J. D. D. Devaraj, P. Sebastian and Yap Vooi Voon, "Vision-based automated parking system," 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), 2010, pp. 757-760, doi: 10.1109/ISSPA.2010.5605408