CS528: Smart Parking Finder

Mehrnoush Alizade
Worcester Polytechnic Institute
Worcester, USA
malizade@wpi.edu

Maryam Ataei Kachooei
Worcester Polytechnic Institute
Worcester, USA
mataeikachooei@wpi.edu

Anushka Bangal
Worcester Polytechnic Institute
Worcester, USA
asbangal@wpi.edu

Sudipta Biswas
Worcester Polytechnic Institute
Worcester, USA
sbiswas1@wpi.edu

Reza Saadati Fard
Worcester Polytechnic Institute
Worcester, USA
rsaadatifard@wpi.edu

ABSTRACT

Our Smart Parking Finder App revolutionizes the parking experience at Worcester Polytechnic Institute (WPI) by seamlessly integrating real-time parking space availability, navigation, and geofencing technologies. Utilizing Firebase for robust data storage, the app empowers users with an intuitive interface to select parking lots, view live occupancy status, and navigate effortlessly using Google Maps. The innovative geofencing feature enhances user awareness, providing notifications when users are in proximity to WPI, accompanied by real-time updates on available parking spaces. This paper delves into the app's architecture, design, and implementation, emphasizing a user-centric approach to optimize the parking experience at WPI.

Keywords: Smart Parking, Mobile Application, User-Centric Design, Google Maps API, Firebase, Navigation, Geofencing.

1 Introduction

Finding an available parking space is more than a mere inconvenience; it represents a substantial economic and environmental burden. As it has been shown in a study by Dan et al [4] the cumulative effect of individuals searching for parking culminates in an estimated 3.6 billion hours of lost time and 1.7 billion gallons of fuel annually, needing of optimizing this aspect of urban infrastructure. The inefficiency of current parking systems and their implications are profoundly felt within the confines of academic institutions such as Worcester Polytechnic Institute (WPI), where students frequently confront the stress of securing parking under time-sensitive and weather-compromised conditions or the urgency to attend classes on time. This scenario is further substantiated in by McCowan et al [5], which illuminated the problem of optimizing parking management at WPI. They noted that a considerable fraction of parking resources remains underused, largely due to the absence of real-time parking information. This gap in the system leads to unnecessary delays and heightened anxiety among students, particularly during rainy days or just before critical academic sessions.

Addressing this problem, our team has developed a specialized mobile application tailored to meet the specific needs of the WPI student community. This application as a user-friendly interface, offering real-time updates on available parking spots, thereby significantly reducing the time and stress involved in finding parking. Its design is rooted in the principles of efficiency and user-centricity, ensuring that students can promptly locate and navigate to a parking space.

This paper delves into the operational mechanics of our application, detailing its technological framework and the resultant impact on the WPI campus environment. We aim to demonstrate how our application not only enhances the parking experience for students but also contributes to a more organized and efficient utilization of campus parking resources.

2 Related Works

SpotHero stands as a widely esteemed smart parking application that facilitates users in locating and reserving parking spaces across diverse locales. Offering real-time availability insights, pricing particulars, and the convenience of advanced reservations, SpotHero has garnered acclaim for its intuitive interface and expansive coverage of parking facilities in urban settings.

ParkMobile emerges as another influential contender within the smart parking app landscape. It presents a comprehensive array of features, encompassing the identification of available parking spaces, mobile payment execution, and the remote extension of parking sessions. ParkMobile streamlines the parking experience by obviating the necessity for physical payment methods, presenting users with a seamless, cashless transaction process.

In conjunction with these established applications, our Smart Parking Finder App distinguishes itself by providing users with a robust solution to effortlessly locate and secure parking spaces. Equipped with real-time availability information and user-friendly features, our application enriches the parking experience by offering a convenient and efficient means to discover and reserve parking spaces across varied locations.

3 Methodology

The architecture of our parking management application is constructed around a dual-component model, which includes the server-side infrastructure and the client-side interface. The server-side is anchored by two pivotal elements: sophisticated image processing techniques and a comprehensive Database Management System (DBMS). Cameras have been installed across the parking areas, optimally positioned to ensure expansive coverage. These cameras stream live

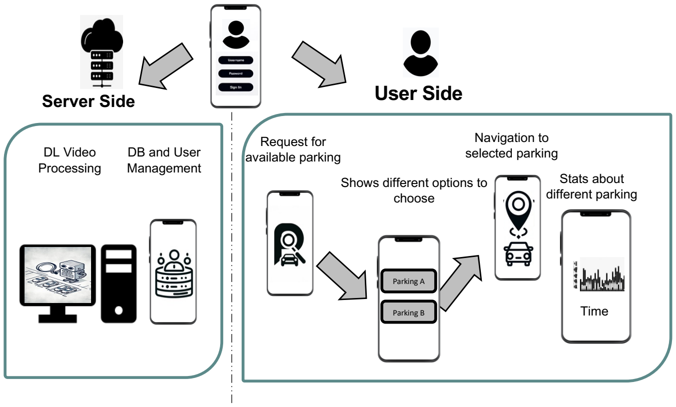

Figure 1: Schematic overview of the system architecture highlighting the dual-component methodology with server-side image processing and data management, alongside the user-side interface for real-time parking navigation.

video to the central server—operated via a MacBook—where custom-developed Python scripts analyze the footage to identify and distinguish between occupied and available parking spaces.

Once the analysis is complete, the information regarding available spaces is transmitted to the Firebase Realtime Database and Update the data for each space. This enables users to view, select, and be guided to an available parking slot in real time. The application thus bridges the gap between the detection of vacant spaces and the user's action to occupy them, providing a streamlined experience. The structural blueprint of the system, which delineates the interaction between the server's analytical capabilities and the user's navigational activities, is depicted in the subsequent Figure **??**.

A relevant demonstration can be accessed via a YouTube video, which provides insights into the practical application of the system (see Link to youtube video for a detailed visual representation).

3.1 Database

In the development of our parking management system, the selection of a robust Database Management System (DBMS) was crucial [3]. We opted for Firebase Cloud due to its proficiency in several key areas. Firstly, Firebase excels at efficient real-time data handling, a necessary feature for an application where the timely reflection of parking space availability is essential. Its real-time database ensures that changes are immediately synchronized across all connected clients, providing users with instantaneous updates. Secondly, Firebase offers a scalable infrastructure. This is pivotal to our application's design, as it must accommodate a potentially large and fluctuating number of users. The cloud-based environment of Firebase allows for seamless scaling, ensuring consistent performance even as user numbers increase. Thirdly, the integrated analytics provided by Firebase is instrumental for ongoing optimization. The analytics tools offer valuable insights into user demographics and behavior, enabling data-driven decisions that enhance user experience. The reports generated by Firebase not only aid in immediate operational adjustments but also contribute to strategic planning by revealing usage patterns and trends.

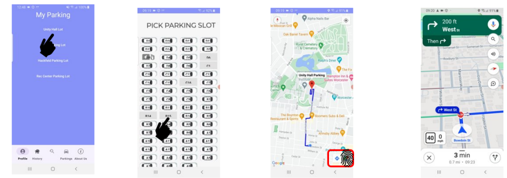

Figure 2:
Initiate route visualization from the user's location to the selected parking area, by clicking on a parking lot. By tapping the marker representing the parking lot on the map, users seamlessly transition to the Google Maps application for detailed navigation

Furthermore, Firebase's architecture is adept at managing a significant user load in real time. This capability ensures that the system remains responsive and efficient, regardless of the number of concurrent users. The utility of Firebase in our system extends beyond mere data storage; it serves as a backbone for the interactive elements of the application, providing administrators with actionable intelligence to continually refine the parking management process.

3.2 Navigation

The navigation functionality of the app is designed to enhance the user experience through two key components: Map Navigation and Geofencing. These components contribute to an effortless and efficient parking process for Worcester Polytechnic Institute (WPI) users.

3.2.1 Map Navigation

The Map Navigation feature of the Smart Parking Finder App provides users with a straightforward process to find and navigate to their desired parking spaces. The user journey is as follows:

Select the Desired Parking Lot: Users are presented with multiple parking lot choices.The app allows users to select their desired parking lot from the available options.

Parking Lot Route: Upon selecting a parking lot, the app initiates route visualization from the user's current location to the chosen parking area. Users can seamlessly view the suggested route, including directions and estimated travel time.

Detailed Navigation with Google Maps: To further enhance the navigation experience, users can tap the marker representing the selected parking lot on the map.

This action seamlessly transitions users to the Google Maps application, providing detailed turn-by-turn navigation instructions. This effortless process ensures that users can efficiently locate and navigate to their chosen parking spots within the Worcester Polytechnic Institute (WPI) campus.

3.2.2 Geofence

Geofencing is a pivotal feature in modern mobile applications, particularly in the realm of location-based services. In our case, developing a Smart Parking Finder application for WPI University involved leveraging geofencing technology to enhance user experience. Geofencing is a technique that uses GPS, RFID, or other location-based technologies to establish a virtual boundary around a geographical area.

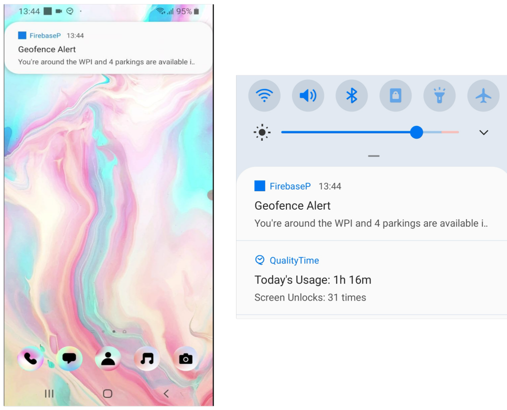

Figure 3: Instant Notification Displayed by the Smart Parking Finder App

When a mobile device enters or exits this predefined area, it triggers specific actions or notifications [6].

In our implementation, a geofence with a radius of 700 meters was set around the university's vicinity. When users' devices enter this zone, the app fetches real-time data from the parking database, retrieving the available slot numbers across all parking lots. The important functionality revolves around accessing the database to obtain the current availability of parking slots across all parking lots near WPI. This seamless integration allows the app to fetch precise details about the number of empty parking spaces available at any given moment.

The notification system then delivers a personalized message to the users. As 3 shows the notification message promptly informs them of the available parking slots in a specific parking lot: "You are around WPI, and 'x' parking slots are available in the 'y' parking lot."

By leveraging geofencing technology, the Smart Parking Finder application optimizes the user experience by providing immediate, location-specific information about parking availability, even when they are not in the application. This proactive approach empowers users to make informed decisions about parking, reducing the time spent searching for vacant spots.

The app's workflow commences with users needing to log in to our application, as illustrated in Fig **??** In the event that users do not possess an account, they have the option to select the "Sign Up" option, as depicted in Fig. 2. Upon clicking "Sign Up," users are prompted to input a new username and password. Subsequently, their account details are saved in our database, and they are then directed to log in again. Upon successful login, the main menu becomes accessible with a toast message confirming that the user has infact successfully logged in. The bottom navigation pane displays four icons, namely User Profile, History, My Parkings and About Us. The user can Browse through the History by searching available parking Spots in WPI by referencing the graphs through a drop-down list of last day, last week and last month. The user can search for parking spaces and then decide which available spot is desired and open the map navigation.

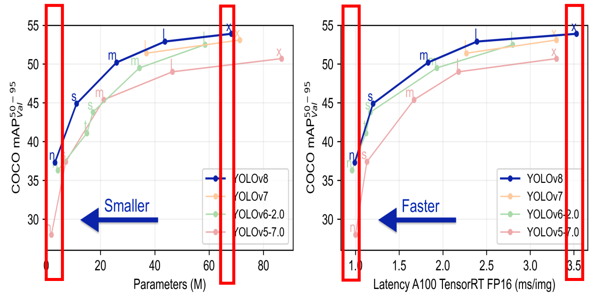

Figure 4: Comparison between different yolov8 models based on parameter size and speed

3.3 Video Processing

The video processing is the hart of our presented SmartParking application. This section presents an exploration of the integration of YOLOv8 [1], a highly proficient deep learning model, within a mobile application framework. The focus is on employing YOLOv8 to detect parking slot occupancy efficiently through the analysis of real-time video streams.

3.3.1 Video Streaming and Preprocessing

The operational sequence starts with the server-end transmission of video data captured by an installed camera (we use an iphone for capturing video), forwarding it to a desktop server. Initial preprocessing of these video frames takes place using the OpenCV library [2] to optimize their suitability for subsequent analytical procedures. This frame-by-frame processing occurs at 5-second intervals, balancing computational efficiency with real-time responsiveness.

3.3.2 Integration of YOLOv8 Model

Central to the process is the deployment of the YOLOv8 model, available in multiple variations with parameter counts ranging from 'n' (1 million parameters) to 'x' (3.5 million parameters). Figure 4 displays various versions of the yolov8 model, demonstrating the speed and accuracy differences among each variant. This model serves as the cornerstone for object detection within the video frames. Pretrained YOLOv8 exhibits remarkable proficiency in identifying diverse objects and provides precise coordinates, employing rectangular pixel-based delineations for entities detected within the frames. Its adaptability and precision render it instrumental in detecting vehicles such as cars, buses, and trucks within specified regions of interest.

3.3.3 Parking Slot Assessment

Before evaluating the outputs of the YOLOv8 model, it is important to specify the parking slots of interest. Designating specific areas within the frames as parking slots, the system relies on the model's detections within these demarcated zones to find out parking occupancy status. Detection of a vehicle (car, bus, or truck) within a designated slot leads to its categorization as 'occupied,' while the absence of detected vehicles denotes the slot as 'vacant.' These assessments are promptly updated within the database. This pipeline is described in Figure 5

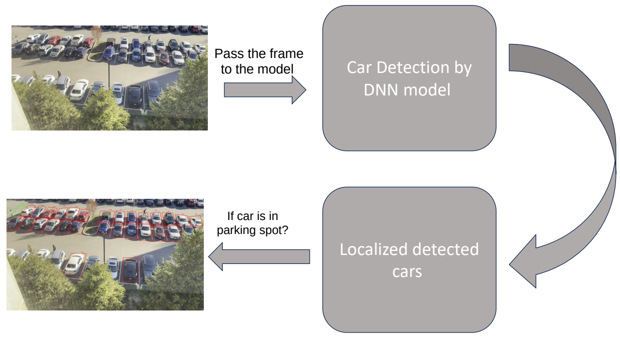

Figure 5: Pipeline for image processing

3.3.4 Database Integration and User Accessibility

Instant updates to the database enable users to easily check available parking spaces. Through the mobile app, people can quickly search for empty slots, making the whole parking search much easier. This combination of YOLOv8 in the video system is a smart way to instantly know which parking spots are free, making it simpler for users to find available spaces.

4 Application Layout

The app's workflow commences with users needing to log in to our application, as illustrated in Fig 6 In the event that users do not possess an account, they have the option to select the "Sign Up" option, as depicted in Fig.[6]. Upon clicking "Sign Up," users are prompted to input a new username and password. Subsequently, their account details are saved in our database, and they are then directed to log in again. Upon successful login, the main menu becomes accessible with a toast message confirming that the user has infact successfully logged in.

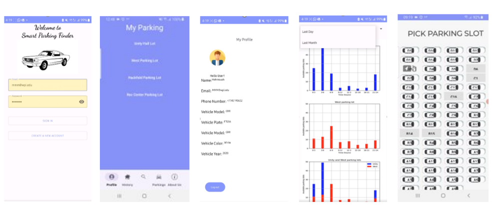

Figure 6: Screens in the App

The bottom navigation panel showcases four distinct icons: User Profile, History, My Parkings, and About Us. Users possess the capability to peruse historical data by searching for available parking spaces using graphs at WPI. This is facilitated through referencing graphical representations, accessible via a drop-down menu presenting options for the last day, last week, and last month. Users can initiate a search for available parking spaces and subsequently make informed decisions regarding their preferred parking spot, leading to the initiation of map navigation. Upon opening the map navigation, users are guided to their designated parking lot.

5 Evaluation

To validate the effectiveness of our proposed model, we conducted two sets of tests. Initially, we assessed various YOLOv8 models through server-side image processing, aiming to select the most suitable model. Subsequently, a real-world test involved our team using the application to locate parking spaces.

5.1 Image Processing Result

Our focus in this section was on comparing the performance of YOLOv8 models, particularly models 'n' and 'x', which represent smaller and more powerful versions within the YOLOv8 spectrum. The test involved passing 10 frames of streaming video through these models, targeting the identification of 32 parking spaces in the Kaven Parking at Worcester Polytechnic Institute. In the context of our parking slot detection system, we define the positive class as an 'empty parking slot' and the negative class as an 'occupied parking slot.' This categorization aids in understanding the model's predictions. A True Positive (TP) occurs when the model accurately identifies an empty parking slot as empty. Conversely, a True Negative (TN) is recorded when the model correctly identifies an occupied parking slot as occupied. False Positives (FP) arise when the model mistakenly predicts an occupied parking slot as empty, erroneously indicating an occupied spot as available. On the other hand, False Negatives (FN) occur when the model incorrectly predicts an empty parking slot as occupied, failing to detect an available spot. These definitions help in constructing the confusion matrix, a crucial tool for evaluating the model's classification performance. Based on defined metrics, the accuracy can be calculated by the below equation:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The evaluation, depicted in Figure 6, showcases a notably superior accuracy in YOLOv8x compared to YOLOv8n (this link shows all test frames). Remarkably, both models displayed impressive efficiency with frame processing times ranging only between 3-5 milliseconds. Given the substantial accuracy advantage of YOLOv8x over YOLOv8n without a significant disparity in processing time, we opted to employ YOLOv8x as our detection model for its superior performance. Figure 5 This assessment underscores the significance of choosing an accurate and efficient model, ensuring reliable parking space detection within the application.

5.2 Applicatoin Test

Our Smart Parking Finder App underwent extensive real-world testing to validate its functionality and usability. The app was deployed and evaluated in the parking lots adjacent to WPI. A comprehensive demonstration of the application's features, user interface, and real-time functionality is available in a demo video accessible on YouTube. The demo video showcases the high accuracy in parking detection by the server, app's seamless navigation, live parking availability updates, and user-centric design. Access the demo video at YouTube Demo Video to explore the application's capabilities in action.

This real-world testing and the accompanying demo video serve as a testament to the application's practicality and effectiveness in addressing parking challenges at WPI.

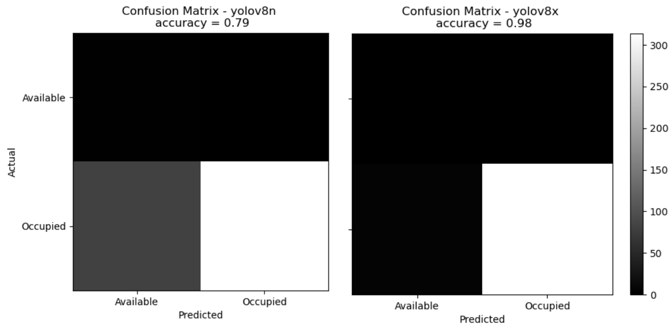

Figure 6: The comparison in accuracy between yolov8n (on the left) and yolov8x (on the right) reveals a significant difference. YOLOv8x boasts an accuracy of 98%, marking a 19% improvement over the n model.

6 Conclusions

The Smart Parking Finder App represents a transformative solution tailored for WPI, reshaping the parking experience through technological innovation. By seamlessly integrating geofencing, real-time database management, advanced image processing, and intuitive navigation, our app mitigates the challenges associated with parking on campus.

This user-centric application empowers users with real-time parking space updates and effortless navigation from their location to the selected parking lot. This feature significantly reduces stress and time spent searching for parking, particularly during critical academic periods or adverse weather conditions. Utilizing Firebase for instantaneous data synchronization, the app ensures prompt updates on parking availability and leverages geofencing technology to proactively notify users. The YOLOv8-powered video processing optimizes parking slot assessment, enhancing user accessibility to available spaces.

In summary, the Smart Parking Finder App not only addresses immediate parking concerns but also contributes to a more organized and resource-efficient campus environment. This application exemplifies the potential of technology to streamline everyday experiences and redefine parking management at WPI. As smart parking evolves, our app stands as a testament to the power of technology in addressing real-world challenges, reshaping the parking landscape for the better at WPI.

6.1 Future works

The following future work suggestions aim to advance the application's capabilities and address potential areas of development:

Top-View Camera Integration: Enhance accuracy by incorporating top-view cameras for detailed aerial parking lot views, improving occupancy detection.

Multiple Cameras per Lot: Expand camera networks tailored for each lot, increasing coverage for precise slot monitoring and status updates.

Cross-Platform Expansion: Extend app compatibility to iOS and desktop platforms, broadening accessibility for diverse users.

UI/UX Refinement: Continually improve the user interface and experience, considering accessibility features for inclusively.

Enhanced Security Measures: Strengthen data encryption and access controls for fortified user data protection.

Community Feedback Integration: Establish feedback channels to engage users, ensuring updates align with user needs.

7 Acknowledgments

The completion of this work was made possible through the insights gained during the course CS528: Mobile and Ubiquitous Computing. The knowledge and skills acquired in this course, which took place during Fall Semester 2023, significantly contributed to the development and implementation of the Smart Parking Finder App.

We extend our sincere gratitude to our professor, Emmanuel Agu, for his guidance, expertise, and support throughout the duration of the course. Professor Agu's commitment to fostering a dynamic learning environment played a pivotal role in shaping our understanding of mobile and ubiquitous computing, laying the foundation for the successful realization of this project.

8 References

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

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

[3] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath. Real-time communication application based on android using google firebase. *Int. J. Adv. Res. Comput. Sci. Manag. Stud*, 6(4), 2018.

[4] S. Das. A novel parking management system, for smart cities, to save fuel, time, and money. *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) Proceedings*, pages 0950–0954, 2019.

[5] M. McCowan, T. Luzardo, J. Dumas, and L.-E. Joubert. Optimizing parking management at worcester polytechnic institute. *Master thesis at WPI*, 2004.

[6] M. Zin, M. Sazali, A. Isa, and M. Isa. Development of auto-notification application for mobile device using geofencing technique. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 7(2):169–173, 2015.