# Machine Vision Smart Parking Using Internet of Things (IoTs) In A Smart University

Noah Sieck
*Department of Physics, Computer Science and Engineering*
*Christopher Newport University*
Newport News, VA, USA
noah.sieck.18@cnu.edu

Cameron Calpin
*Department of Physics, Computer Science and Engineering*
*Christopher Newport University*
Newport News, VA, USA
cameron.calpin.14@cnu.edu

Mohammad Almalag
*Department of Physics, Computer Science and Engineering*
*Christopher Newport University*
Newport News, VA, USA
mohammad.almalag@cnu.edu

***Abstract*—The concept of smart cities is becoming more widespread as the critical need for smart parking systems becomes evident in these cities. This paper suggests a method of smart parking in a heretofore relatively untouched area: midsize college campuses. With a high concentration of people and expansion with a relatively low increase in parking spots, parking problems are commonplace on college campuses. There is no designated terminology for universities utilizing smart systems, with suggestions ranging from "smart campuses" to "smart universities." This paper discusses a proof of concept structure utilized by the researchers to implement a smart parking system in a midsize university parking lot. The system looks promising as evidenced by the results gathered using the proposed system. The results are discussed later in the paper and the implications for the smart parking system.**

***Index Terms*—smart parking, machine learning, smart university, smart campus, parking system, parking tracker**

## I. Introduction

The benefits of implementing a smart parking system are not only parking and time related, but also environmentally related. Shoup [1] calculated that 47,000 gallons of gasoline are wasted per year while searching for parking in Westwood Village, a small business district in Los Angeles with a population of over 50,000 in 2008. This excess expenditure of fuel generates seven-hundred-and-thirty tons of carbon dioxide from that single district. It is estimated that smart parking systems will save 220,000 gallons of fuel by 2030, and 300,000 gallons by 2050 if smart parking systems are integrated successfully into the current parking infrastructure [2].

The authors were approached to create a short-term solution for on-campus parking while the location and construction of new parking lots are decided at Christopher Newport University (CNU); a midsize university. Other options, such as public transportation, were determined to be ineffective on the campus due to its small size. Additionally, the university already implements a paid parking system, so the researchers had to find a new way to optimize parking that did not interfere with already existing structures.

To further demonstrate the severity of the lack of parking not only on CNU's campus, sources were found supporting the problem that is evident nationwide [3], [4].

According to INRIX [5] it is estimated that the average US driver wastes seventeen hours a year searching for parking, costing an estimated $72.7 billion a year from fuel and emissions, or $345 per year per driver. This is a large sum of money that drivers lose every year due to the struggles of parking, which can be alleviated by a smart parking system.

In this paper, we propose a smart parking system for university campuses that struggle with having sufficient parking. While not a long-term solution to parking problems, the smart parking system and associated mobile application aim to improve efficiency in the short-term for parking.

The paper is organized as follows. We discuss related work in section II, as similar projects have been attempted before to varying degrees of success and implementation. In section III, the smart parking system is presented, along with the software developed that the smart parking system application is using. Our results are then presented in section IV, affirming our choice of classifier. Lastly, we discuss future work that we would like to perform to further improve the smart parking system in section V.

## II. RELATED WORK

Gupta et. Al., [6] presented a smart parking system that utilizes a server which collects data from ultrasonic sensors and transmits it to an application, called "ParkX," to help users expedite the parking process. The mobile application exists in both desktop and Android mobile forms and uses Wi-Fi to transmit the data between the server and application. Additionally, the mobile app has a map to view parking spots but lacks a close-up view of any of the parking lots. The ParkX application had an additional objective to deal with in the form of collecting payment for parking spots, which our own app did not have to concern itself with. Other smart parking systems that use ultrasonic sensors and mobile applications include [6]–[10].

Satre et. Al. [11] presented another mobile app that focuses on parking lots. A literature review of many diverse ways to accomplish the building a smart parking system was included in the paper. While the authors used RFIDs in their research, some different proposed methods included image processing, vision-based systems, and license plate recognition, of which the former two techniques our own system utilizes. One area of the system where the authors agreed heavily with the article was on the use of a MySQL database to store data on the parking spots [10]–[12]. Easily updated and manipulated, the MySQL database was an important choice of the many databases available which is affirmed by its use in other similar smart parking systems. Other researchers have used RFIDs in their parking applications [12]–[16]. In a survey published by Lin et. Al. [17], they claim that "…if drivers can have real-time parking availability information, they will be able to adjust their traveling schedule without spending time cruising the city in vain," and suggest the use of a smart parking system to accomplish this objective. This validates the use of a real-time application and smart parking system. The survey also covers software previously used in smart parking systems including SQL servers and transference of data through Wi-Fi and JSON files, which the Park Fast smart parking system uses. The survey also suggests an easy-to-use user interface for drivers so they might benefit from the smart parking system.

Alam et. Al. suggest using real time smart parking systems in their paper [18]–[20]. However, their smart parking system uses magnetic sensors instead of cameras to provide real time data. This requires a large number of sensors to be acquired for each parking spot and then installed. With a camera-based detection system, the detection devices (a camera versus many smaller sensors) are concentrated in a single piece of equipment, making it easier to update and manage that part of the system. The paper also mentions a web and mobile application developed to operate in real time which when tested had a noticeable effect in the parking lots it was installed in.

Kamble et. Al. created a conceptual smart parking system and coupled app, called "Spark," to manage paying parking spots [21]. While the similarities are slim between Spark and Park Fast, the former seeking to manage for-profit parking and the latter freely providing information on parking, both use an app to help streamline the parking process for their users. The paper did not provide the cost it would take to implement the system, so the researchers are unable to compare the cost effectiveness of their system. Spark provides a feature to allow its users to reserve parking spots in advance [20]. Such a feature would be useful to implement in ParkFast, at least in a restricted sense. To ensure fairness the user would have to have a GPS location near campus before reserving a spot.

Bura et. Al. describes a smart parking solution using cameras with a full view of a parking lot connected to an edge device that will gather the occupation status of parking spots and the license plates of parked cars [22]. This system is extremely close to the proposed ParkFast system, using cameras to evaluate the availability of parking spots in a parking lot. The paper even suggested using an Nvidia Jetson Tx2, which the researchers are using a newer, smaller, and cheaper version of called the Nvidia Jetson Nano. The web application uses Google Maps to display the live data and allows users to find their vehicle by inputting their license plate, simultaneously allowing them to pay their parking fee through the application. The paper also focuses on a solution for parking garages by using LIDAR sensors instead of cameras. One way our system improves on the paper is by building a real time mobile application to help disseminate information about open parking spots, instead of a website. The data gathered using this system is analyzed to show the positive impact the system has had on university parking.

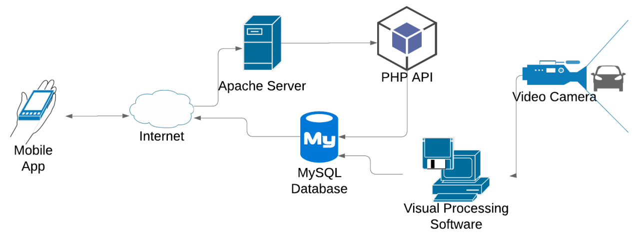

Fig. 1. Smart parking system architecture diagram.

## III. SMART PARKING SYSTEM

### A. System Architecture

The system that we have developed can be seen in Fig. 1. The system begins with cameras overlooking parking lots to identify open parking spaces lots on campus and mark them as available for drivers. This information is fed into a MySQL database, which the app we developed draws data from using a PHP API. The system operates in real time to provide drivers with the most accurate information possible. The app then displays the data in a user-friendly way, allowing drivers to pinpoint exactly where the open spot(s) is(are) available in the parking lot. We recognize that a variable in our system is camera height, as it must be adjusted for various sized lots. The smaller and medium-sized lots can be monitored by mounting several cameras onto the sides of buildings, while the larger lots will require cameras mounted on poles. We acknowledge the variables that come with this solution, such as winds and exposure to the elements. We plan to encase the parts of the system that are vulnerable to the elements in a weatherproof box.

### B. Application

The app, called ParkFast, was developed as an iOS application using Swift 5 on Xcode version 10.2.1. While apps have been developed for smart parking systems before, the accuracy

Fig. 2. Map of campus parking lots.

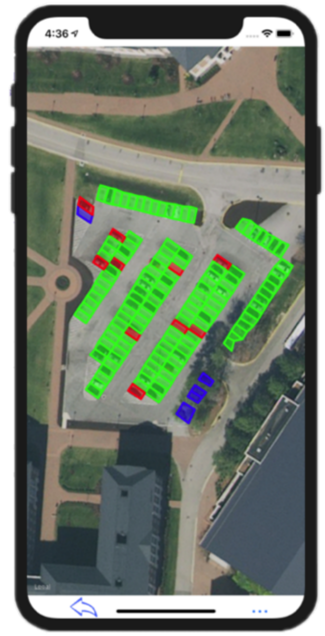

Fig. 3. Close up view of a parking lot.

and specificity to which the app helps the user identify open parking spaces has not, as we believe, been developed to this extent described hereafter.

*1) User Interface:* There are four “levels” of the app the user can navigate to whilst using it, however one is a landing screen and the other a login screen, and neither will not be seen again by the user after they have entered the necessary information when the app is opened for the first time.

The login screen asks if the user is faculty, staff, or student, because certain parking lots on the campus are closed to ensure professors are able to make it to class, or staff is able to arrive on time. The app stores this data for later, changing the availability of certain lots based on the user’s choice. For example, a “student” user would see the faculty parking lot “E4” colored red, signaling the student user that while the lot may have parking spots available, the lot is unavailable for them to park in. After viewing the subtext that notifies the user that by pressing the “Submit” button they agree to the terms and conditions of the app, the user is taken to the main screen of the app seen in Fig. 2. The user will no longer see the previous two screens again as their selection is saved in the app, and when loading the app again it will automatically send the user to the main screen to avoid having to re-input all their data again.

On the main screen of the app all the parking lots on campus are marked with large colored annotations for easy visual access. The annotation is slightly transparent to allow the user to see what lies behind the annotation, for example, a road. It would not be conducive to the functioning of the app if it were to obstruct the user’s vision of main roads that the user could utilize for navigation purposes. The number of available parking spots in each lot is displayed inside the annotation above it. By being displayed in the annotation, it does not clutter the map or detract from the name of the lot displayed underneath of the annotation, and it will scale automatically with the size of the annotation as the user scrolls in and out on the map. When tapped on, the annotation will take the user into a single view of the selected parking lot.

The researchers implemented a novel technique in their application in the form of a map overlay of the parking lots. As seen in Fig. 3, a close-up view of a single parking lot is shown with each parking space highlighted and outlined. The annotation for the parking lot disappears to allow the user an unobstructed view of all the parking spots. This will allow the user to easily identify and navigate directly to an open parking spot, further saving them time in the search for parking. By pressing the arrow button in the bottom left corner, the user can navigate back to the main map view of campus parking lots. The user can navigate to the last level of the app by clicking on the ellipses in the bottom right corner of the app.

The app has one last “Help” screen which provides detailed information on how to utilize the app to its fullest extent. This screen can be accessed from both the campus-wide screen and the individual parking lot screens. After exiting the help screen through the ‘X’ button in the top right corner, the user is brought back to the main screen and the campus-wide view of parking lots or the individual parking lot they have selected.

*2) Functionality:* The app was developed using Swift 5.0 to accomplish the objective that the app be deployable on an iOS platform. Swift’s MapKit Application Programming Interface (API) was used to display the Apple Maps map in the app. Apple Maps was chosen over the Google Maps Software Development Kit (SDK) because MapKit is native to the development environment and as such should allow the app to run faster than an SDK. Additionally, a nonnative SDK or API would have to be downloaded and installed, significantly increasing the download size of the app. The Google Maps SDK would also lower the speed of the app as it would have to contact its own servers to display the data.

The user's choices on the login screen are stored via core data, eliminating the need to store and later access the data from a separate database. The collected data is so few it was deemed acceptable to store on the user's device, as it would not interfere with the speed of the running app. It is also easier to retrieve core data in the context of the app, which is to check if the user is allowed to park in certain parking lots on campus.

On the main map screen, the annotations change color depending on how many parking spots are left in that parking lot. If there are more than fifty spots available to park in, the annotation will be green. If there are between fifty and twenty-six parking spots left, the annotation will turn yellow. If there are between twenty-five and one parking spaces open, the annotation will turn orange, and if there are no parking spaces left in the parking lot, the annotation will turn red. Another feature of the main map screen is if the user pans or zooms away from the campus too far, the app will re-center and adjust the zoom level of the map back to the campus. This feature was implemented in order to keep the app's focus on fulfilling its objective. It would be pointless to allow users access to the entirety of the Apple Maps map when only a small portion of it is utilized by the app. Thus, to ensure the app fulfills its specific niche, the map will fix itself if the user manipulates the map too far away from the campus.

When the user selects a parking lot for a close-up view, a back arrow appears in the bottom left corner of the app in the existing whitespace. This button only appears to the user when a transition back to the main map screen is needed from a close-up parking lot view. The button disappears when it is pressed, and it transitions the user back to the campus-wide map view. Each parking space displays a given color at any point in time and is liable to change if the requirements are met. A green highlighted parking spot means that the parking spot is open and available to be parked in. A blue parking spot denotes a handicapped parking spot and is used to indicate to users that they should not park in that parking spot unless allowed to do so by law. A red parking spot signifies the spot is occupied, and the user cannot park there.

The parking tracker software uses multiple programming languages, some more than others. Most of the program is written in Python 3.6 which will be necessary for the OpenCV software. Since the iOS app needs to fetch data from the parking tracker app, we built a MySQL database with a table containing two data objects: the parking ID and the parking status. The MySQL database will help us visually see the data being sent from the parking tracker program. The other programming language that is used is PHP. We needed a script to access the database and send a dictionary containing the parking ID and the parking status for the iOS program to receive.

TABLE I
THE CASCADE CLASSIFIER RESULTS

| | **Cascade Classifier Results** | | |
|---|---|---|---|
| | ***True Positives*** | ***False Positives*** | ***Detection Rate*** |
| Test Set 1 | 1,425 | 75 | 95% |
| Test Set 2 | 1,340 | 160 | 89.3% |

### C. MACHINE VISION

"The system uses the Haar Feature-based Cascade Classifier firstly introduced by Viola and Jones which is known for its fast processing and good detection rates. The process requires representative data sets to be used for training and validation including positive (presence of objects to detect) and negative (absence of objects to detect) image samples. The result of the application of each feature to a particular image region is given by the sum of the pixels that lie within the black rectangles of the feature subtracted by the sum of the ones overlapping the white rectangles." [23]. The first method of verification uses the complexity of horizontal edge features obtained by horizontal Haar-like feaures. The second method uses the symmetric properties of the vehicle. Most rear parts of vehicles have symmetric features. It is one of the features that appears continuously without any changes, so the symmetric feature of the vehicle is the most important feature for detecting vehicles. Each frame that is processed goes through a Gaussian Filter. This is a linear filter that will assist in blurring and reducing noise in the image (Gaussian Filter Equation). The Gaussian filter is also faster since it multiplies and adds instead of sorting. To detect shadows, we use background subtraction on top of the Gaussian and Laplacian filters. The background subtraction process includes the detection of the background image from a sequence of the frames generated from the captured video [24]. We use a machine learning based approach where a cascade function is trained from a multitude of positive and negative images. Table I shows the cascade classifier results from analyzing and predicting thousands of images It is then used to detect objects in other images. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from sum of pixels under the black rectangle. For each feature calculation, we need to find the sum of the pixels under white and black. The equipment that we used in this system are as follow:

*1) Computer:* The computer being used to process the parking tracker program is the NVIDIA Jetson Nano. The Nano is equipped with a Quad-core Arm Cortex-A57 cores each clocked at 1.43 GHz, a 128-core Maxwell GPU, 4GB 64-bit LPDDR4 RAM and 16GB eMMC flash storage. Besides the processing power, the size of the Jetson (70 x 35 mm) is quite small and portable. The Nano provides two options as a power source: USB Type B Micro and DC barrel jack.

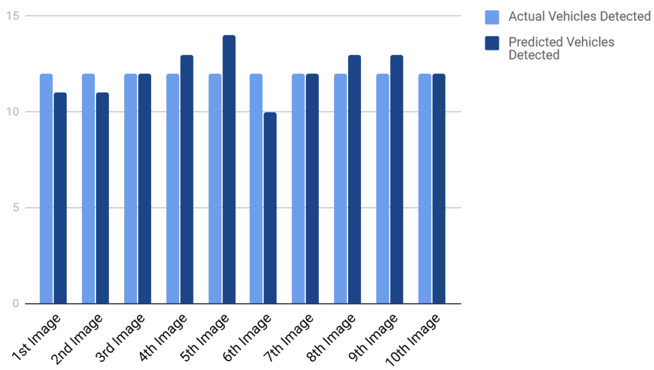


Fig. 4. Classifier one results showing more cars than exist.

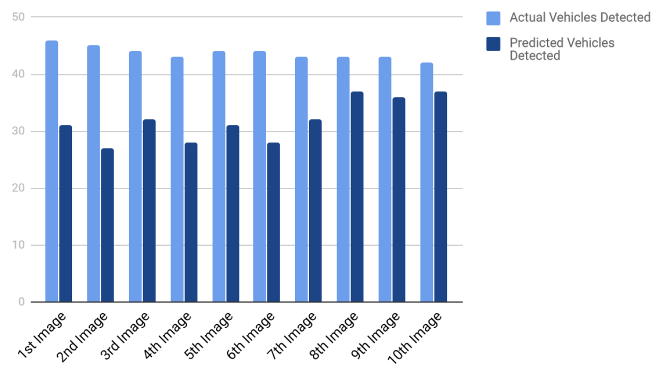


Fig. 5. Classifier two results showing accurate detection of vehicles.

The USB is rated at 5V/2A whereas the barrel jack is rated at 5V/4A. To utilize all the processing power we need, we use a Matek Systems 5V 12V/4A universal battery eliminator circuit.

*2) Cameras:* We used two cameras to obtain the data and simulate the parking tracker program. The first camera is the Logitech C920 Pro HD webcam, connected via USB, which is used to test and develop the machine vision classifier. This webcam is capable of recording 1080p at 30 frames per second. The camera was mounted on window sills and on top of parking garages to get a good angle and simulate where the second camera will be placed. The second camera is the Funxwe security camera. This will be used when our program is deployable and ready for real time use. This camera is solar powered and is powered by two 5800mAh LiPo batteries. The resolution for the security camera is the same as the webcam, 1080p at 30 frames per second. The security camera is also wireless, so we can set up a server and have our parking program communicate via TCP/UDP. To visualize the program running, we use RTSP protocol which will have the program overlapping the live streaming data from the camera. The Wi-Fi network operates on 2.4GHz.

*3) Solar System:* The power source that runs the Nvidia Jetson and the Ubiquiti antenna is an ExpertPower 12V 12AH Sealed Lead Acid (SLA) battery. Not only is this SLA battery reliable and long-lasting, the size of the SLA battery and the voltage rating are perfect for operation and box fitting requirement. To recharge the SLA battery, we selected a Renogy 80W 12V solar panel. The voltage rating matches the 12V SLA battery and provides enough power to recharge the SLA battery while powering the Nvidia Jetson and the Ubiquiti Antenna. To regulate the charge capacity of the SLA battery from the solar panel, we attached a Renogy Wanderer 10A 12V/24V PWM solar charge controller.

## IV. RESULTS

The algorithms being used for this program are best suited for static cameras. "A static camera observing a scene is a common case of a surveillance system. Detecting intruding objects is an essential step in analyzing the scene. A usually applicable assumption is that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that don't t the model" [20]. To see how well the Haar Cascade classifier performed, we mounted the Logitech webcam on top of one of our parking garages at CNU. Plans weren't pushed yet to mount a tall pole in one of the lawns near Luter Hall overlooking Luter parking lot. To counter the problem, we decided to have the webcam overlooking Luter lot from the third floor of Luter Hall in one of the classrooms. This was a problem because the sample set that the Haar Cascade classifier was trained on couldn't distinguish car features from a skewed angle. We executed a program that ran the classifier on some images of Luter lot and the classifier was struggling to detect cars in a skewed position. To obtain better results from the classifier, we chose another spot that overlooked cars that were perpendicular to us. Testing images from the third story of CNU parking garage proved to show better results. (Insert image with test classifier of CNU parking garage and results).

There are two classifiers used with two different image sets of cars. The first classifier predicted more cars that the actual amount specified, as seen in Fig. 4. The second classifier, seen in Fig. 5, showed better, if not near perfect, results for classifying the number of cars present in a lot. To obtain these results, we had to tune the algorithm's parameters to fit the model of the sample photos given.

After some tuning of the algorithm's parameters, the end result of the classifier proved to be working efficiently. The reaction time for a region of interest to detect and switch contour colors took only a couple of frames. The colors that indicate whether a parking space is open or closed is nearly perfect. Fig. 6 shows a screenshot of the classifier identifying cars and determining the parking space status. The only drawback with testing at the CNU parking garage is the height the camera is visualizing from. From Fig. 6 we can see that one of the cars cuts into one of the regions of interest and changes status from open to closed. If we increase the height of the webcam, then the chances of a car cutting into a region of interest decreases thus giving more accurate readings and

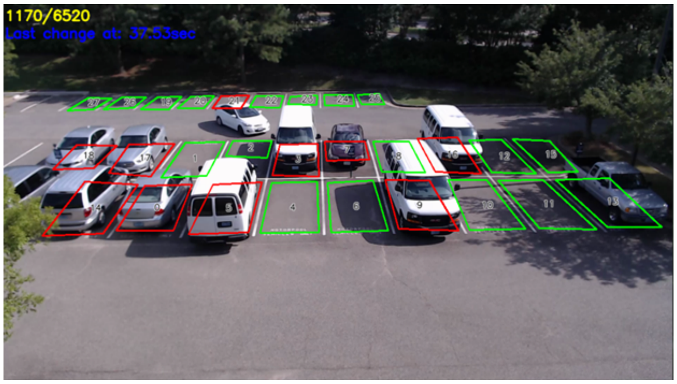


Fig. 6. Classifier two results identifying every vehicle in the parking lot.

less interference.

## V. FUTURE WORK

There are several features the researchers would like to implement in the system as they continue to develop it. They would like to implement a hands-free way to use the app as it is l that user will be accessing the app while driving. In order to make it as safe as possible, the hands-free aspect will be implemented in the form of a Siri Shortcut. This feature would allow the user to say a customized phrase and be directed towards the nearest parking lot with an open space. Another feature that would be beneficial to implement would be parking spot reservation. It would only function if the user is physically within a mile of the campus, and the app would allow the user to reserve a parking space in an open parking space. This feature would only be implemented as the app gains popularity among faculty, staff, and students, as those without the app cannot see reserved parking spaces and thus might park in one, not knowing it is reserved. The researchers would also like to expand the number of parking lots the system covers. Permissions must be obtained from the university before installing the permanent camera fixtures and as such not all parking lots on the university campus are equipped with the technology required to become compatible with the system. As data is continued to be collected with our smart parking system, we would like to begin to integrate AI features within the system. With the data, we would like to be able to predict the times at which the parking lots will reach their highest point of occupancy. This data would be invaluable for research purposes and would allow us to help manage parking on campus more efficiently.

## REFERENCES


[1] D. Shoup, “Cruising for parking,“Access, no. 30, p. 16, Spring 2007. [Online]. Available: http://shoup.luskin.ucla.edu/wp-content/uploads/sites/2/2015/02/CruisingForParkingAccess.pdf [Accessed January 15, 2020].

[2] A. Basu, “Smart Parking,“ Happiest Minds Technologies Pvt. Ltd., Bangalore, India, [Online]. Available: https://www.happiestminds.com/whitepapers/smart-parking.pdf [Accessed January 15, 2020].

[3] [3]L. dell’Olio, et al. “A methodology based on parking policy to promote sustainable mobility in college campuses,“ Transport Policy, Vol. 80, pp. 148-156, 2019.

[4] B. Glatz, “Who Benefits from Smart Parking¿‘ Fybr, Saint Louis, MO, USA, 2014 [Online]. Available: http://www.fybr-tech.com/who-benefits-from-smart-parking [Accessed October 22, 2019].

[5] G. Cookson and B. Pishue, “The impact of parking pain in the US, UK and Germany,“ INRIX, Kirkland, Washington, USA, [Online]. Available: http://www2.inrix.com/research-parking-2017 [Accessed: July 18, 2019]

[6] A. Gupta, S. Kulkarni, V. Jathar, V. Sharma, N. Jain, “Car parking management system using IoT,“ American Journal of Science, Vol. 2, No. 4, pp. 112-119, 2017.

[7] A. Sahfutri, et al. “Smart parking using wireless sensor network system,“ ICECOS, pp. 117-122, 2018.

[8] J. Ruili, W. Haocong, W. Han, E. O’Connell, S. McGrath, “Smart parking system using image processing and artificial intelligence,“ ICST, pp. 232-235, December 2018.

[9] J. Singh, N. Ravi, and S. Krishnan, “IoT based parking sensor network for smart campus,“ IJET, Vol. 7, No. 4.35, pp. 26-34, 2018.

[10] R. Grodi, D. B. Rawat and F. Rios-Gutierrez, “Smart parking: Parking occupancy monitoring and visualization system for smart cities,“ SoutheastCon 2016, Norfolk, VA, 2016, pp. 1-5.

[11] S. Satre, P. More, S. Shaikh, O. Mhatre, “Smart parking system based on dynamic resource sharing,“ IJESC, Vol. 8, No. 3, pp. 16236-16238, March 2018.

[12] 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, Montreal, QC, 2012, pp.

[13] E. Tsiropoulou, J. Baras, S. Papavassiliou, S. Sinha, “RFID-based smart parking management system,“ Cybre-Physical Systems, Vol. 3, pp. 22-41, Aug 2017.

[14] Z. Pala and N. Inanc, “Smart Parking Applications Using RFID Technology,“ 2007 1st Annual RFID Eurasia, Istanbul, 2007, pp. 1-3.

[15] L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi and R. Vergallo, “Integration of RFID and WSN technologies in a Smart Parking System,“ 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, 2014, pp. 104-110.

[16] M. Patil, V.N Bhonge, “Wireless sensor network and RFID for smart parking system“, International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 4, April 2013, pp. 188-192.

[17] T. Lin, H. Rivano and F. Le Mouël, “A Survey of Smart Parking Solutions,“ in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 12, pp. 3229-3253, Dec. 2017.

[18] M. Alam, et al. ,“Real-time smart parking systems integration in distributed ITS for smart cities,“ Journal of Advanced Transportation, 2018. [Online] Available: https://www.hindawi.com/journals/jat/2018/1485652/B19 [Accessed January 15, 2020].

[19] Z. Qadir, F. Al-Turjman, M. A. Khan and T. Nesimoglu, “ZIGBEE Based Time and Energy Efficient Smart Parking System Using IOT,“ 2018 18th Mediterranean Microwave Symposium (MMS), Istanbul, 2018, pp. 295-298.

[20] H. He, Z. Zhang, P. Yan, “A real-time reservation service for smart system,“ Proceedings of 2018 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, pp. 1–6, 2018.

[21] P. Kamble, S. Chandgude, K. Deshpande, C. Kumari, K. M. Gaikwad, “Smart parking system,“ International Journal of Advance Research and Development, Vol. 3, No. 4, pp. 183-186, 2018.

[22] 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), San Francisco, CA, 2018, pp. 17-24.

[23] M.L. Oliveira, V. Santos, “Automatic Detection of Cars in Real Roads using Haar-like Features,“ Proceedings of The 8th Portuguese Conference on Automatic Control, 2008.

[24] Z. Zivkovic, ”Improved adaptive Gaussian mixture model for background subtraction,” Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Cambridge, 2004, pp. 28-31 Vol.2.

[25] P. S. S, V. Vibha, C. Manisha, “Vehicle Detection and Tracking using the Optical Flow and Background Subtraction“, Elsevier Proc. of Int. Conf. on Advances in Computer Science and Application, pp. 741-747, 2013.