# Car Park Occupancy Monitoring System Based on Background Subtraction and Object Detection

Ash Gupta
Department of Computer Engineering
University of Canterbury
Christchurch, New Zealand
agu50@uclive.ac.nz

Richard Green
Department of Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract* – This paper proposes a method to automatically monitor the occupancy of a car park. The current parking process is inefficient as it requires users to manually search for a parking space, resulting in an unproductive use of time.**

**The proposed approach utilises a camera-based system with background subtraction, adaptive thresholding, and dilation to detect movement in a user-defined parking space. The detected movement is then validated using the 'You Only Look Once' (YOLO) object detector to determine whether a vehicle was detected. The occupancy status of the parking space is then updated accordingly.**

**The proposed approach was tested on video footage from the Psychology car park at the University of Canterbury and achieved a final accuracy of 89%, which was higher than the previous research with an accuracy of only 68%. The proposed solution could be deployed to make the parking process more efficient for both the users and the parking enforcement officers by allowing for easy monitoring of the car park.**

***Keywords – background subtraction, adaptive thresholding, dilation, object detection, car park monitoring, YOLO object detection***

## I. Introduction

There are currently over 1.2 billion cars on roads globally [1], and this number is set to double by the year 2040 [2]. With this ever-growing increase in cars, the problem of finding a car park becomes a time-consuming task for the users. A research conducted by Inrix concluded that an average driver in the U.S. spends around 17 hours per year searching for a car park, which results in $345 in wasted time, fuel and emissions [3]. This shows that the current method of manually searching for a car park is inefficient and an ineffective process.

Another problem with the current parking system is the difficulties with regulating and monitoring the parking spaces to detect users who have not paid and illegally parked their vehicles. Currently, the parking wardens oversee the car parks and ensure that the parking trespassers are issued a fine. This is achieved by parking wardens manually checking every car for their paid time, and issuing a ticket if the user has overstayed. This is a very labour-intensive process and often results in parking wardens missing some illegally parked cars, which is a loss of revenue for the parking companies.

This paper discusses the implementation of a camera-based system for autonomously monitoring the occupancy of a parking lot.

## II. Background

There have been prior attempts and research with the aim of resolving the issues faced with the current parking system, as discussed in the relevant sections below.

### *A. Electronic sensors for detecting the presence of a vehicle*

Wireless Sensor Networks (WSNs) have been previously utilised in the field of intelligent parking management systems [4]. Such systems operate by placing a wireless module in each parking space, which houses a vehicle detection sensor coupled with a wireless transceiver for transmitting the sensor data to a gateway node (Figure 1). The data from the gateway node is then uploaded to a cloud server and is used by the parking enforcement companies to monitor the occupancy of the carpark.

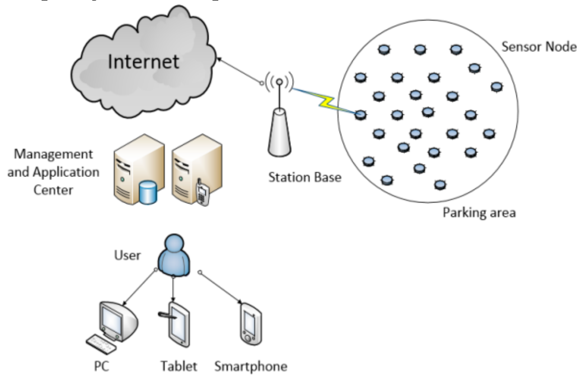


*Figure 1. Shows how the wireless sensor network is deployed in a smart parking application*

For detecting the presence of a parked vehicle, the parking modules houses either a singular or a combination of vehicle detection sensor such as ultrasonic, infrared, or magnetic.

Ultrasonic sensors are commonly used in measuring distance between objects and have been previously deployed in the intelligent parking-monitoring applications [5]. However, the high-power usage of ultrasonic sensors, makes such sensors inefficient and impractical in the smart parking industry, as they require regular charging of the battery, which increases the maintenance cost.

Infrared sensors have also been previously realised in such systems [6]. However, infrared sensors use heat pulses to detect the presence of an object, which limits their usage in certain geographical locations. As the body of the car is mainly comprised of metal, this can also interfere with the heat pattern, making the sensors ineffective [7].

Lastly, the magnetic sensors have been implemented with great success in such applications [8]. Such sensors observe the change in magnetic flux caused by a moving metallic object, which is ideal for detecting the presence of a car.

Wireless sensor systems such as the one proposed by Hilmani et al. [4] feature the following limitation which prevents them from being deployed at a large scale:

- The initial purchase cost of such systems is a major disadvantage, as it requires a parking module to be placed in every parking location. This is especially a problem for large car parks that house hundreds of parking spaces.
- The installation requires drilling a hole in the current parking spaces and fitting the wireless modules in place. This is a labour-intensive process which requires construction workers.
- The ongoing maintenance is another bottleneck for the system. Due to the modules being secured underground, in the event of failure or damage, the entire unit needs to be removed before it can be fixed. This increases the overall cost of the system and makes such systems impractical.

### *B. Camera-based systems for monitoring a car park*

Alongside the implementation of physical sensors, camera-based systems for monitoring the occupancy of a car park have also been previously researched.

In 2015, Abduo proposed a camera-based solution with the aim of providing a cost-effective parking monitoring system [9]. Abduo's proposed solution required users to manually define the parking spaces that were to be monitored. This was achieved by users drawing a polygon on a calibration image, which consisted of an empty car park. The coordinates from the polygons were saved in a file as region of interests (ROI). Absolute difference comparison and hue-saturation-value (HSV) space analysis were implemented to detect the presence of a car in the ROIs.

Absolute difference comparison functions by subtracting two input images and ensuring that the result of the output matrix is positive [10]. Abduo used the calibration image and the current frame as the input images for the absolute difference comparison method. The number of non-zero pixels from the resulting matrix was then measured. The status of the parking space was updated depending on the number of non-zero pixels measured in each ROI.

In addition, Abduo also implemented HSV colour space analysis to further solidify the detection of a vehicle in a parking space (Figure 2). HSV space analysis method is commonly used in image analysis for feature detection and image segmentation [11]. Abduo compared the hue, saturation, and the brightness value coefficients from the calibration image with the current frame and determined whether the change in HSV space for the two input images was above a user-defined threshold. The occupancy of the respective parking space was then updated accordingly. However, HSV space analysis is highly dependent on lighting conditions, as it directly compares the hue, saturation, and the brightness of two images. Therefore, such methods require constant lighting conditions, which is not possible in an outside environment, where the light intensity varies with the time of day and weather conditions. In addition, the comparison of colour values between the two input images can result in false negatives caused by vehicles with a similar colour to that of the empty parking space.

Abduo claims to have achieved a detection rate of 100% when a vehicle was present. However, the paper does not discuss the false negatives caused by the inaccuracy in the HSV space analysis method, or the false positives caused by the movement of a pedestrian from the absolute detection method. Therefore, the 100% obtained sensitivity cannot be confidently supported as it lacks the discussion of other variables in effect.

*Figure 2. Shows the output of the HSV space analysis with an ROI mask on top to detect the presence of a vehicle*

In another paper, Canny Edge Detection and Hough Line Transform were used for automatic assignment of ROI in a car park [12]. Canny Edge Detection is a multistage algorithm commonly used in image processing for extracting features and detecting edges of an object [13]. Hough Line Transform was utilised for detecting the marked lines that differentiated the parking spaces in a car park using a voting system (Figure 3). The outputs from the automatic assignment of ROI were then used to perform background subtraction and HSV space analysis to detect the presence of a car. The automated allocation of ROI restricts the versatility of the system, as some car parks are not marked with clear lines that define the individual parking spaces, making such methods ineffective.

*Figure 3. Shows the output from the Canny Edge Detector to automatically allocate available parking spaces and generate ROI*

In 2016, Garry described a method for utilising Hough Circle Transform to monitor the occupancy of a car park by detecting a wheel of a parked vehicle (Figure 4) [14]. Hough Circle Transform searched for circular objects in the ROI and used a voting system to determine the validity of the findings. The equation used for defining the circle is shown in equation 1.

$$(x-a)^2 + (y-b)^2 = r^2 \qquad (1)$$

Where $(a, b)$ is the centre of the circle and r is the radius. A user-defined threshold was then used to compare the radius of the detected circles to filter the ones that were not the expected size of a wheel. If an expected circle in an ROI was detected, then the parking space was considered occupied. Garry's proposed approach achieved an accuracy of 68%. This poor accuracy was largely due to the angle and the low resolution of the camera, which restricted the view of the wheel.

*Figure 4. Shows the wheel detection using Hough Circle Transform*

All the prior research mentioned above suffered from a high number of false positives, due to objects such as pedestrians and other foreign objects classifying the parking spaces as occupied. Therefore, to reduce the number of falsely classified objects in the parking spaces, a method for differentiating and validating the detected object is required.

### C. *Using a Convolutional Neural Network*

The limitation of occupancy detection methods discussed above can be overcome by using a Convolutional Neural Network (CNN) to validate and accurately classify the detected object in the parking space.

In 2018, Dwivedi proposed a method for using a top-down view of a car park and a self-trained Tensor Flow network to classify the occupancy of the parking spaces [15]. Dwivedi's proposed system used the approach of self-allocating ROI using Canny Edge Detection and Hough Line Transform. Each frame of the video was then processed with Tensor Flow network to determine whether any vehicles were present in each ROI. A custom dataset was used which included 550 images of both vacant and occupied parking spots. Dwivedi's trained model obtained an accuracy of 94% (Figure 5). This proposed system had low false positive rates due to only using deep learning to validate the occupancy of the parking spots. However, processing time per frame was 0.6 seconds, which was due to the resource intensive CNN processing of every frame of the video. This limitation makes the proposed approach impractical for a real-time application, as the output frame rate is less than 2 frames per second.

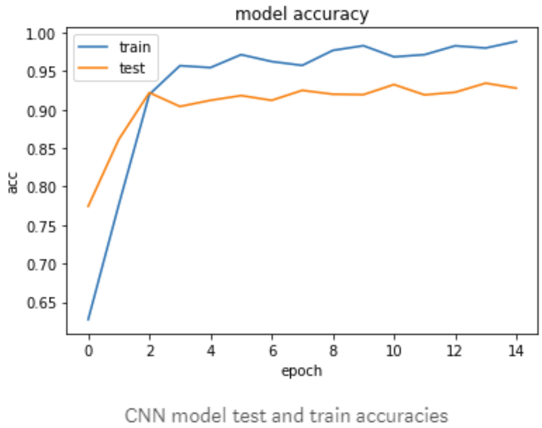


*Figure 5. Shows the accuracy achieved of the trained model to detect cars in a car park*

The use of CNN greatly improved the accuracy of the camera-based system to monitor the occupancy of a car park. However, it increased the processing power required to sample the video, resulting in an impractical system which cannot be utilised in real time.

## III. Proposed Solution

The proposed solution extends upon prior research by combining the vision-based algorithms for detecting movement with object detection to validate the occupancy of the parking space. The proposed approach features three distinct processes: manual ROI declaration for defining the parking spaces, background subtraction for detecting movement, and object detection for validating the movement. These processes are discussed in their relevant sections below and are also shown in Figure 6.

It was assumed that a stationary camera mounted high above the parking lot would be utilised for this approach.

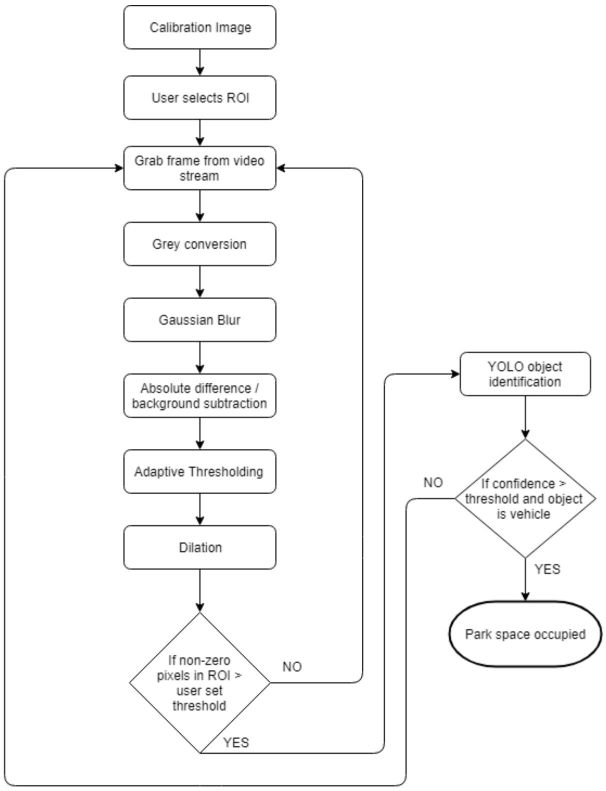


*Figure 6. Shows the flow chart for how the proposed system operates*

*A. Defining region of interests*

The proposed system first requires the user to define the desired parking spaces. This is achieved by displaying a calibration image with unoccupied parking locations. The user is then required to click on the desired corners and create a polygon (Figure 7). The user has the ability to define as many parking spaces to monitor as they desire, depending on the computer's specification to sample the data in real time. Once the user has selected all the desired parking locations to monitor, the ROI coordinates are then saved to a calibration file by pressing the 'a' key.

The manual calibration process of parking spaces is an advantage over the prior research [12], as it enables the system to be versatile and be used in conditions where automatic ROI declaration is not feasible, such as car parks without any marking for individual parking spaces.

The purpose of the calibration image is to provide the system with a point of reference to determine any change between the un-occupied state and the current state. Therefore, it is crucial that the calibration image features empty park spaces, to allow for a comparison point.

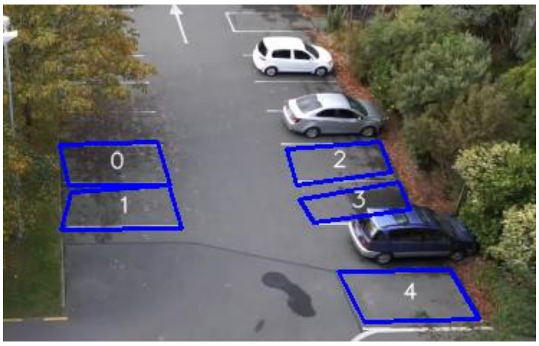


*Figure 7. Shows the region of interests marked by users to declare the parking spaces that are to be monitored*

*B. Detecting movement in parking spaces*

Once the user has defined the parking spaces to monitor, the process of detecting movement in the live video stream is executed. The following algorithms are then applied to the live video stream.

The first stage is to read the calibration file to get the coordinates of the parking spaces. These coordinates are then used to draw a rectangle to display the status of the parking space; where green represents vacant, and red is occupied.

Next, the calibration image and the current frame from the video stream are converted to grey scale. Colour information is not required to process the images and find features. Therefore, the colour information is suppressed as it acts as a form of noise. This results in the calibration image and the current frame to only house data regarding the light and dark pixels, making for easier detection of features without unnecessary information.

Gaussian blur is then applied to reduce noise from the video stream and the calibration image [16]. The equation for Gaussian blur is shown in Equation 2. A kernel size of $21\ X\ 21$ is used to calculate the standard deviation. Gaussian blur is commonly used to suppress high-frequency components from an image that is interpreted as noise, therefore by applying a low pass filter (Gaussian blur), white noise is reduced from the images.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/2\sigma^2} \tag{2}$$

Where $\sigma$ is the standard deviation of the distribution. It is assumed that the distribution has a mean of zero and is centred on the line $x = 0$.

The absolute difference between the two blurred images is then calculated to determine the change between the calibration image (vacant position) and the current frame. This is achieved by subtracting the two images and ensuring that the result is an absolute value. This process is referred to as background subtraction. This results in a matrix with pixels of value '0' when there is no difference between the two input images, or a positive number if a difference is detected. As the difference could only be caused by moving objects due to the camera being stationary, the matrix displays a black background with only moving foreground objects visible (Figure 8).

*Figure 8. Shows the result from absolute difference calculations from the calibration image and the current frame being processed*

Thresholding is then applied to transform the absolute difference image into a binary image with pixel values of only '0' or '1' [17]. This allowed for clearer detection of moving objects as the resulting pixels are all '1'. In addition, thresholding was also utilised for suppressing noise and filtering pixels with low intensity, to get a result which only detects significant movement and not background movement, such as blowing leaves.

Dilation is then applied to the threshold image. Dilation is an operation under the set of mathematical morphology and is commonly used to fill holes between objects [18]. A structuring element of a 10 x 10 matrix of one's was used for convolution with the input image. This resulted in an image where the holes between the detected object's features were filled to create one large detected segment, allowing for easier processing.

Once the processes discussed above were executed, this resulted in a black image that only detected significant moving objects, such as cars, vehicles, and people, whilst ensuring the background noise was suppressed.

Next, the pixels under each marked parking region were checked for non-zero value – which represents checking for any movement in the desired ROI. This was achieved by counting the number of pixels with a value of '1'. This number was then compared with the desired threshold set by the user. The value of the threshold corresponds proportionally to the size of the object that is required to be detected. If the value of non-zero pixels in an ROI exceeds this threshold, an image of the ROI of the current frame is then processed using You Only Look Once (YOLO) object detector, as discussed below.

*C. Object identification*

There could be detected movement in the ROI due to a variety of reasons such as pedestrians, animals, and other foreign objects. Such objects do not necessarily occupy the parking space. However, the movement detection algorithms will classify the parking space as being occupied, which is a limitation of prior research [9] [12]. The proposed approach overcomes this issue by implementing YOLO object detection for validating the detected movement in the ROI.

YOLO was designed to run in real time as it only requires one forward propagation pass through the network to make predictions, combined with non-maximum suppression to only detect an object once. The algorithm can achieve a real-time frame rate upwards of 45 FPS [19]. The traditional approach of the R-CNN system provides a higher accuracy as there are multiple layers involved in the neural network. However, this results in high processing time and low output frame rates, making such systems impractical to be used in real time, which was a limitation of prior research [15].

The use of object detection only when there is significant movement in the ROI, eliminates the need for full-time object detection. In addition, YOLO is only conducted on the ROI image, which is a smaller image including only one object. Therefore, the proposed system functions at a high frame rate with minimal processing power, allowing to be used in real time and on low powered embedded computer systems.

COCO classifier data set is used to validate the detected object. This is a pre-trained classifier with support for a wide range of objects such as vehicles, people, and animals [20]. The proposed approach validates whether the object in the ROI is a vehicle and if the detection's confidence is higher than the user set threshold then the respective parking space state is considered occupied.

## IV. RESULTS

The proposed methods were tested on a system with the following specifications:

- Processor: Intel core i5 4590 @ 3.3 GHz
- Windows 10
- Visual Studio Code
- Python 3.7, OpenCV 4
- Camera: Sony a6300 @ 1920 x 1080 60 FPS
- Video footage: Psychology car park, University of Canterbury (17 Minutes)

The proposed approach was an iterative process which concluded of building the minimum viable system, quantifying the results, then refining the system by adding additional algorithms. This ensured that no redundant algorithms or features were being added that affected the performance and the power consumption of the system.

The results from the iterative testing process, including the methods used, are shown in Table 1.

*Table 1. Shows the methods used in the proposed solution and their respective accuracy.*

| Methods | Accuracy | Avg. FPS |
|---|---|---|
| Background subtraction | 50 % | 60 |
| BG subtraction + thresholding | 72 % | 60 |
| BG subtraction + thresholding + dilation | 78 % | 58 |
| Above + YOLO object detection | 89 % | 50 |

The accuracy percentage shown in Table 1, represents the number of times that a vehicle was successfully detected in the ROIs, and also accounts for the false positives and the false negatives witnessed during the duration of the test video footage. The testing process resulted in no false negatives, depicting that a vehicle was detected 100% of the time it was present. However, the false negatives were high during the first iteration of testing which only consisted of background subtraction approach. The false positives reduced as more methods were added to the system, hence the overall accuracy increased.

Background subtraction alone had the lowest accuracy. This was because there was no suppression of noise or validation of the detected movement, resulting in every little movement triggering the occupancy status.

Thresholding increased the accuracy of the proposed solution as there was an increased suppression of noise, and the movement was easier to detect since it was a binary image; consisting of only '0' and '1' pixel values (Figure 9).

The addition of dilation method further increased the accuracy as holes between the detected objects were filled, and hence a larger uninterrupted object was detected (Figure 10).

Lastly, YOLO object detection increased the accuracy of the proposed solution by a further 11%. This final approach featured the lowest false positives and hence the highest overall accuracy. This was largely because the detected movement was validated to determine whether it was caused by a vehicle or a foreign object (Figure 11).

It is worth noting that the average frames-per-second of the proposed solution decreased by 8 with the inclusion of YOLO object detection. This final frame rate is still higher than that of the traditional R-CNN deep learning approach. The final frame rate did not decrease significantly due to the efficient methodology of the YOLO object detection, and because of only performing object detection when necessary, as discussed in section 3C.

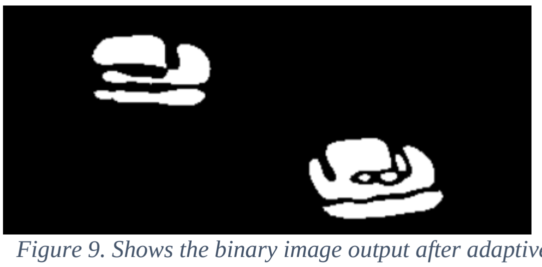

*Figure 9. Shows the binary image output after adaptive thresholding being applied to the proposed approach*

*Figure 10. Shows the output after dilation being applied to the image to fill holes in the detected object*

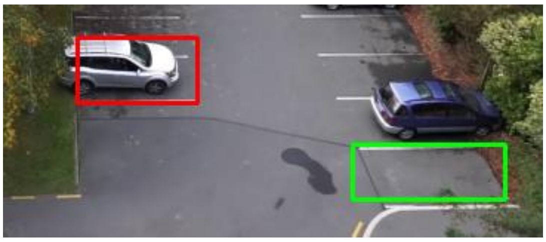

*Figure 11. Shows the output from the YOLO object detection, where the red rectangle represents an occupied parking space, and the green represents vacant*

### A. *Limitation of research*

The frame rate of the proposed solution outlined in Table 1, only represents the system being tested on a high-performance PC, rather than the desired application of low-cost single-board computing. Therefore, without the results from the system running on a low-power computer such as Raspberry Pi, it is not possible to quantify the real-time performance of the system.

The video footage used during the testing of the proposed solution was captured in one setting with optimal light and traffic conditions. Therefore, the result from the system may vary depending on the environmental conditions.

The position and the angle of the camera may result in large vehicles obstructing the view to smaller adjacent cars. This would result in false detection of parking spaces.

A high mounted camera as the one used for testing of the proposed solution is prone to movement due to wind and birds, which can cause false detection and offset the ROI, requiring recalibration of the system.

## V. CONCLUSIONS

This paper proposed a method to efficiently and autonomously monitor the occupancy of a car park using a camera-based system. The proposed solution utilises background subtraction, adaptive thresholding, and dilation, to detect movement in the user-defined park spaces. YOLO object detection was used to validate and determine whether the detected movement in the respective park space was from a vehicle occupying the park or a negligible foreign object.

The proposed solution resulted in a final accuracy of 89% which is higher than the previous research which featured an accuracy of 68% [14]. The increase in accuracy of the proposed solution over the prior research is mainly due to the inclusion of additional methods such as dilation and validation of the movement using the YOLO object detection.

The field of view (FOV) and the angle of the mounted camera features a limitation and a bottleneck for the proposed solution. As the limited FOV can enable a large vehicle obstructing the view of a smaller adjacent car, resulting in a false reading of the occupancy status. Another limitation of the proposed method is that an involuntary movement of the camera caused by a gust of wind or a bird could result in shifted ROIs and hence false detections.

### A. *Future Research*

The following enhancements can be performed as future research into the proposed method to further improve the accuracy and efficiency of the system:

- Testing the system with an array of camera placed at different locations to reduce the effect of large vehicles obstructing the view.
- Sense passive movement of the camera and adjust the ROI positions to counteract this involuntary movement.
- Testing the proposed solution on a low-power single board computer to validate the performance in real-time.
- Training a custom data set for YOLO with images of cars and other foreign objects captured from the desired angle of the camera, to further reduce false detection and increase efficiency.

## REFERENCES


[1] J.Voelcker. "1.2 Billion Vehicles On World's Roads Now, 2 Billion by 2035: Report." Internet: https://www2.greencarreports.com/ news/1093560_1-2-billion vehicles-on-worlds-roads-now-2-billion-by-2035-report, Jul. 29, 2014 [May. 28, 2019]

[2] M. N. Smith. "The number of cars will double worldwide by 2040." Internet: https://www.businessinsider.com/global-transport-use-will-double-by-2040-as-china-and-india-gdp-balloon/?r=AU&IR=T, Apr. 20, 2016 [May. 28, 2019]

[3] G. Cookson and B. Pishuc. "The Impact of Parking Pain in the US, UK, and Germany," Inrix Research, Jul. 2017

[4] A. Hilmani, A. Maizate, and L. Hassouni. "Designing and Managing a Smart Parking System Using Wireless Sensor Networks." *Journal of Sensor and Actuator Networks*, vol. 7, no. 2, p. 24, 2018.

[5] M. Chen and T. Chang. "A Parking Guidance and Information System based on Wireless Sensor Network," in *Proc. IEEE Int. Conf. Information and Automation, Shenzhen*, China, 2011, pp. 601–605.

[6] A. Agrawal, P. Kamde, R. Mourya, and M. Bhanushali. "Car Parking System using IR Sensors." *International Journal of Advances Research in Engineering, Science & Technology*, vol. 4, no. 2, Feb. 2017.

[7] P. Zappi, E. Farella, and L. Benini. "Tracking Motion Direction and Distance With Pyroelectric IR Sensors." in *IEEE Sensors Journal*, vol. 10, no. 9, pp. 1486-1494, Sept. 2010.

[8] Z. Zhang, X. Li, H. Yuan, and F. Yu. "A Street Parking System Using Wireless Sensor Networks." *Int. J. Distributed Sensor Networks*. vol. 2013, pp. 1–10, 2013.

[9] M. Abduo and R. Green. "Enhancing Parking Space Monitoring System," Computer Vision Lab, University of Canterbury, May 2015

[10] Y. Cheng and J. Wang. "A Motion Image Detection Method Based on the Inter-Frame Difference Method." *Applied Mechanics and Materials*, vol. 490-491, pp. 1283-1286, 2014.

[11] H. Cheng, X. Jiang, Y. Sun, and J. Wang. "Color image segmentation: advances and prospects." *Pattern Recognition*, vol. 34, no. 12, pp. 2259-2281, 2001.

[12] N. Hayatdavoudi and R. Green. “Development of Parking Space Monitoring System by Vision Based System,” Computer Vision Lab, University of Canterbury, May 2016

[13] G. M. H. Amer and A. M. Abushaala. “Edge detection methods.” *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, 2015.

[14] N. Garry and R. Green. “Parking Space Occupancy Monitoring Using a Low Frame-Rate,” Computer Vision Lab, University of Canterbury, May 2016.

[15] P. Dwivedi. “Find where to park in real time using OpenCV and Tensorflow.” Internet: https://towardsdatascience.com/find-where-to-park-in-real-time-using-opencv-and-tensorflow-4307a4c3da03, Oct. 16, 2018 [May. 28, 2019]

[16] B. Dhruv, N. Mittal, and M. Modi. “Analysis of different filters for noise reduction in images.” *2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, 2017.

[17] P. Roy, S. Dutta, N. Dey, G. Dey, S. Chakraborty, and R. Ray. “Adaptive thresholding: A comparative study,” in *Proc. 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014.

[18] D. Chudasama, T. Patel, S. Joshi, and G. I. Prajapati. “Image Segmentation using Morphological Operations.” *International Journal of Computer Applications*, vol. 117, no. 18, pp. 16–19, 2015.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection,” in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] “Common Objects in Context,” *COCO*. Internet: http://cocodataset.org [Jun. 05, 2019].