International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–9

http://dx.doi.org/10.12785/ijcds/1570964473

Real-Time Car Parking Detection with Deep Learning in Different Lighting Scenarios

Fatema H. Yusuf[1] **and Mohab A. Mangoud**[1]

[1]*Department of Electrical Engineering,University of Bahrain, Isa Town, Kingdom of Bahrain*
Received 12 October 2023, Revised 18 September 2024, Accepted 19 September 2024

Abstract: This paper presents an intelligent parking system utilizing image processing and deep learning to address parking challenges amidst varying lighting conditions. The escalating number of vehicles on the road has exacerbated the difficulty and time spent in finding available parking spaces. To alleviate this issue, we propose an efficient camera-based system capable of real-time detection of open parking slots using deep learning. Initially, we introduce a simple parking detection technique utilizing image processing; however, its inefficiency in low-light conditions is highlighted. Subsequently, we introduce an efficient AI-powered system, using different versions of YOLO algorithm, which accurately identifies available and occupied parking slots by detecting vehicles within the parking area. The strategically positioned webcam provides comprehensive coverage of the parking area, with an initial image serving as a reference for identifying all parking slots. During operation, the webcam records real-time video footage of the parking area, enabling continuous updates for an accurate count of free and occupied parking slots. Moreover, the model performs well during day and night and under varying weather conditions. The paper details the step-by-step implementation of the system and showcases achieved results under diverse lighting conditions. In conclusion, this research demonstrates the system's effectiveness in mitigating parking challenges through the combination of image processing, deep learning, and real-time video analysis. We have tested and compared our solution against state-of-the-art techniques. Additionally, we highlight the future potential for research to further enhance and advance this innovative system.

Keywords: Smart Parking System, Parking Slots, Image Processing, Object Detection, YOLO

1. Introduction

Recently, there has been a significant increase in the number of vehicles, resulting in traffic congestion. One major contributor to traffic congestion is parking cruising. A recent study [1] shows that between 9% and 56% of traffic is caused by drivers searching for available parking spaces. The cruising for parking wastes valuable time, with an average of 6 minutes spent in the search. In addition to time wasted, this search for parking also has negative environmental impacts, as vehicles emit harmful emissions like CO2, which can adversely affect human health. To address this problem, a real time smart parking system is proposed as a potential solution. This system aims to assist drivers in quickly locating available parking spaces by detecting vehicles in the parking area and counting the number of vacant slots and showing their locations. By implementing such a system, we can reduce traffic congestion, save time, and minimize fuel consumption, ultimately benefiting both individuals and the environment. However, challenges in implementing this system include accuracy of the used detection technique, performance under different lighting conditions and camera orientation.

One of the previously proposed solutions for smart parking systems [2] utilizes the HSV color segmentation method to obtain the background image of the parking area. The image undergoes preprocessing, including median filters and background detection, followed by background subtraction during the detection process. Various image processing techniques such as blurring, thresholding, and filtering are applied to enhance the detection performance. The system focuses on the region of interest (ROI) corresponding to the parking spaces to detect objects. By comparing the ratio of white pixels in each ROI to the total number of pixels with a threshold value, the system classifies whether the parking space is occupied or not. The highest average accuracy of 95.76% is achieved with a minimum threshold value of 0.4. However, this paper used a classical image processing technique for the detection process that is very dependent on the position and angle of the CCTV camera at the parking lot.

In another proposed system [3], a computer vision technique is employed to extract video footage of a

E-mail address: 20173010@stu.uob.edu.bh, mmangoud@stu.uob.edu.bh

specific parking area, which is then processed using MATLAB. Object-Oriented Programming is used to count the available parking slots and determine their status. The results are displayed visually on a website. The authors stated that effects caused by changes in environmental lighting while image processing, was a variable not evaluated in this research. Therefore, in this paper, the performance under different lighting conditions will be investigated.

In recent years, Among the various object detection algorithms, the YOLO (You Only Look Once) framework has stood out for its remarkable balance of speed and accuracy, enabling the rapid and reliable identification of objects in images. YOLO has become a central real-time object detection system for robotics, driverless cars, and video monitoring applications. In [?] authors presented comprehensive survey starting with original YOLO up to YOLOv8 summarizing the essential of YOLO's development. Deep learning YOLO v3 algorithm was utilized in [4] to detect vehicles and parking spaces using three datasets with images captured. An enhanced version of YOLO v3 algorithm is proposed by adding a residual structure to extract features from images containing both parking spaces and vehicles. Four feature maps are employed in the detection process, enabling complex networks to extract additional features. the K-means clustering algorithm is used to cluster and classify car parking spaces in the dataset. PKLot dataset was used.

"PKLot – is a robust dataset for parking lot Classication" (http://web.inf.ufpr.br/vri/news/ parking-lotdatabase). It contains 12,417 images of parking lots in Brazil taken from three different camera feeds of two different parking lots. The images were taken over a 30 day period at 5 minute intervals. Each parking lot image is annotated with the date, time and current weather conditions (either sunny, cloudy, or rainy). The model is tested on a parking space of 63 spaces resulting in a precision of 91.6% and 93.3% and a recall of 87.2% and 90.9% for Yolo and enhanced Yolo respectively. However, it is mentioned that illumination has an impact on the detection effect of the algorithm, and it is necessary to further improve the algorithm.

In [5], modified YOLOv4-tiny (mYOLOv4-tiny) model was introduced to detect vehicles based on real-time traffic flow data. During the classification process, Convolutional Fuzzy Neural Network (CFNN) and Vector-CFNN models and a network mapping fusion method to get enhanced feature fusion and higher classification accuracy. The mYOLOv4-tiny model is a simplified network using YOLO consisting of 24 convolutional layers, and 3 max pooling layers for feature extraction. The precision and recall of the model yielding precision and recall of 97.5% and just above 97%, respectively detection speed of more than 30 frames per second. The BIT-Vehicle Dataset are used which is a public dataset for vehicle classification collated by Beijing Institute of Technology. The dataset contains a total of 9,850 vehicle images, all of which were captured using 2 cameras at different times and locations on highways. These images differ in brightness, proportions, and surface color. The dataset includes six vehicle types: buses, minibuses, minivans, sedans, sports utility vehicles (SUVs), and trucks.

Another proposed computer vision-based smart parking lot occupancy detection system [6] utilizes a Raspberry Pi3 attached to a camera and a model running a simplified Alexnet model which consists of five layers: three convolutional layers followed by two fully connected layers. The mAlexnet model is trained, and the proposed system is tested on PKLot yielding accuracy of 97% and an accuracy of 88% for the created SWUPark across several weather conditions

In [7], system utilizing CNN algorithms is trained to classify the occupancy status of periodically captured parking slots. A mask is applied to identify the slots, and each slot's image is segmented from the overall parking area image. The system employs two CNN architectures, mLeNet and mAlexNet, to determine slot availability in different scenarios: using a single camera for both training and testing from the same angle, and multiple cameras for training and testing from different angles. The system using mAlexNet is compared with another proposed system leveraging RBF kernel SVMs. Two comparative experiments are conducted using different datasets, CNRPark and PKLot. In one experiment, the model is trained and tested within the same dataset, while the other involves training in one dataset and testing in the other. The highest accuracy of 98.1% is achieved when the mAlexNet model is trained and tested on the PKLot dataset. However, this technique did not use an object detection algorithm. Also, results need to be tested under different light condition changes, presence of shadows, and partial occlusions and when tests are performed using images captured from a viewpoint different than the viewpoint used for training.

In [8], an image processing model is created to find free outdoor on-street parking spots in the chosen sample street by using deep learning techniques. Parking regions are defined manually and done in advance. The authors used a custom dataset containing images of parking slots from different viewpoints, labeled, and categorized into two classes to determine the presence of a car. During the classification process, a feature detector kernel is used to traverse images along with a Conv2D layer with a 3x3 kernel, creating 32 different feature maps for each image. Image features, including headlights, mirrors, and windows, are enhanced using Pooling layers. For a specific scenario, as a proof-of-concept, a single roadside camera and one street are used to evaluate the system. The system employs various neural networks with distinct activation functions, achieving its highest accuracy of 92% using two Conv2D layers with 32 kernels of 3x3 sizes along with a MaxPooling layer with 2 by 2 feature extractors.

A smart parking management system [9] has been trained using 100 images of both occupied and vacant parking slots across 4 parking areas. The system employs two cameras—one for detecting available parking slots and accidents within the parking area. One camera utilizes Optical Character Recognition (OCR) to identify vehicle registration plate numbers, which are then processed in the tracking system. The vehicle number acts as an object ID for vehicle detection, implemented using the YOLO v3 algorithm. The system tracks the vehicle while it's in motion and detects when it is parked, achieving an accuracy of over 95%. Additionally, the system integrates with an AWS cloud system to extract parking information, including location and the number of empty slots. This information is then displayed on a mobile application. The other camera uses deep learning to detect accidents within the parking area. By adding more training images, up to 500, to distinguish between vehicle collisions and noncollision scenarios, the system achieves an accuracy above 95%.

In [10] a system has been developed using CNN framework. The dataset used for this problem is PKLot. The paper employs two classifiers to automatically detect available parking spaces: a binary classifier (vacant or occupied) and a multinomial classifier (number of vacant slots). The binary classifier divides the input image into individual spots and rotates them vertically to classify parking slots as either occupied or empty. In contrast, the multinomial classifier does not require image segmentation and has different input sizes and labels compared to the binary classifier. It calculates the number of free slots in the given image, up to a predetermined maximum value of empty slots. If the input image has more empty slots than the predetermined value, the system considers the number of free slots as the predetermined value. The system utilized 70% of the images for training, with 20% for validation and 10% for testing. It has been tested under various weather conditions, including rainy and sunny days, achieving a maximum accuracy of 99.7%. This high accuracy was obtained by employing three convolutional and fully connected layers in the model.

A proposed system [11] automatically detects parking spaces by capturing a series of images of the parking area. Vehicle images are segmented using a Cascade Mask RCNN model. This model is similar to the Mask R-CNN but incorporates a cascade structure for improved accuracy in detection. The segmented vehicles are then processed using a method called a heat map, which determines the positions of the parked vehicles in the image while disregarding moving vehicles. This process yields a list of coordinates for the occupied parking slots. The proposed method employs instance segmentation to identify cars and, using vehicle occurrence, generate a heat map of parking spaces. The results using twelve different subsets from the PKLot and CNRPark- EXT parking lot datasets show that the method achieved an AP25 score up to 95.60% and AP50 score up to 79.90%.

In [12], proposes a vision-based vehicle detection and counting system. A custom dataset containing 11,129 images of three different types of vehicles on a highway is created. These images are captured from various angles and lighting conditions. A highway video is recorded, and each frame (image) is segmented into two areas to enhance accuracy. The detection process involves applying the YOLOv3 algorithm to both areas. Subsequently, the ORB algorithm is employed to determine the vehicle trajectories, providing detailed information about the vehicle, such as its type, number, and driving direction. This information is valuable for traffic analysis. The system is tested on three videos captured from the same angle but with different numbers of frames. It achieved a maximum precision and recall of 0.88 and 0.89, respectively. The previous related work can be summarized in Table I, for performance assessment (Note: P: Precision, R: Recall, AP: Average Precision, F1-Score, Acc: Accuracy) along with the approach used in each paper.

In Section 2, we introduce a smart parking system incorporating image processing techniques tested at both high and low lighting conditions. Section 3 outlines employing proposed deep learning for vacant parking detection technique and applying YOLOv3, providing insights into the methodology employed in model development and its testing across diverse lighting scenarios. Moving on to Section 4, we extend our methodology by applying YOLO-v8 to a larger parking lot sourced from a public dataset, with the addition of a custom dataset created specifically for testing the proposed low lighting conditions without applying bounding boxes. Finally, Section 5 encapsulates the conclusions drawn from our study and outlines potential avenues for future research.

2. Car Parking Detection Using Image Processing

In this section, elementary detection technique using image processing is presented. Figure 1 shown below represents the block diagram of the method used that is divided into two parts: selection and detection. During selection process, the image of the parking area is used as a reference to select all the slots. Whereas in the detector part, live stream video is recorded from a high position with the use of a webcam. The video is segmented into frames and then each frame is exposed to several processes.

A picture of the parking area is taken from a high position and is used as a reference to select the parking slots. Then, it is passed to the model. Within a loop, the dimensions of the slots are optimized, and the slots are outlined in blue. This process is illustrated in Figure 2.

Parking slot detection consists of three processes which are as follows.

TABLE I. Comparison of object detection metrics

Ref	P	R	F1-score	AP	Acc	Approach
[2]	-	-	-	-	95.8	Image processing
[4]	93.3	90.9	93.3	-	-	YOLOv3
[5]	96.3	99.6	97.9	-	-	YOLOv4-tiny
[6]	93.3	90.9	93.3	-	87	YOLOv3
[7]	97	98	88	-	98.1	CNN-mAlexnet
[8]	-	-	-	-	92	
[9]	-	-	-	-	>95	CNN
[10]	-	-	-	-	99.7	B-CNN
[11]	-	-	-	89	95.6	C-Mask R-CNN
[12]	88	89	-	88	-	YOLOv3

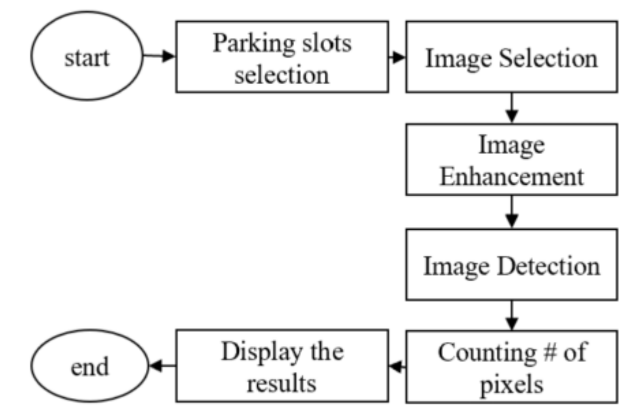

Figure 1. Block diagram of detection using image processing

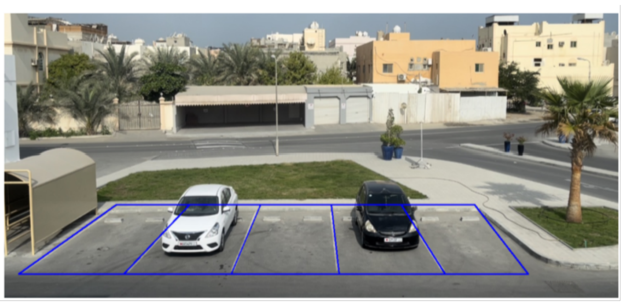

Figure 2. The selection of target parking slots

A. Image Segmentation

In this process, the image of the parking area is segmented into five RGB images, each representing a parking space, and then all are converted to greyscale using the Luminosity formula as follows:

$$Y = 0.299\ \mathrm{R} + 0.587\ \mathrm{G} + 0.114\ \mathrm{B} \tag{1}$$

The results are shown in the following figures.

Figure 3. Cropped (left) Grayscale transformed images (middle) Blured image (right)

B. Image Enhancement

To get rid of unnecessary noises, the image is blurred using Gaussian blurring method in which a Gaussian kernel is used. In [5, eq. (2)], the equation of Gaussian filter kernel is represented. As the kernel, k (that should be given in odd and positive number) increases, the blur increases. The standard deviation of both the horizontal and vertical axes are calculated from the kernel size.

C. Image Detection

The edge of the blurred image is detected using a Sobel Kernel that uses the equations represented in [6, eq. (3)] and [6, eq. (4)] to compute the gradient and direction of the edges.

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))2 + (j-(k+1))2}{2\sigma^2}\right) \tag{2}$$

$$\text{Edge Gradient(G)} : G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$\text{Angle} : \Theta = \tan^{-1}\left(\frac{G_x}{G_y}\right) \tag{4}$$

The result is shown in Figure 4. The number of white pixels that represent in the edge is counted and based on it, the status of the parking slot is determined and then the number of available spaces is displayed.

Figure 4. Edge detetcion technique

The proposed system utilizes PyCharm as the Integrated Development Environment (IDE) and incorporates the OpenCV library for image processing. PyCharm is a comprehensive IDE for Python, encompassing essential programming functions and tools like a code editor, compiler, and debugger. Computer vision (CV) is a field of computer science that involves understanding images and videos, their storage, and data extraction from them. OpenCV, short for Open-Source Computer Vision Library, is a powerful tool that provides a wide array of programming functions, particularly vital in artificial intelligence (AI) and image processing. OpenCV is extensively employed for real-time object detection.

This detection method has been tested in the Market Mall parking area in Bahrain and the results were satisfactory as shown in figure 5 and 6. Figure 5 represents the identification of one parking space showing that it is available and marked in green.

Figure 6 shows that even if a person is walking through the parking space, the status of that space is still considered available (vacant) because the number of white pixels is not large enough. So, only if there is something as large as the vehicle then the system considers the slot to be occupied.

However, when the same model used at evening time, smart parking system was tested that determines the availability of parking spaces based on image processing as shown in Figure 7. If a parking slot contains enough white pixels, it indicates that the slot is occupied by a vehicle. Conversely, if the number of white pixels is low, the slot is considered free. Therefore, it is concluded that this detection method performs well during the daytime. However, it exhibited poor performance at night, as depicted in the following

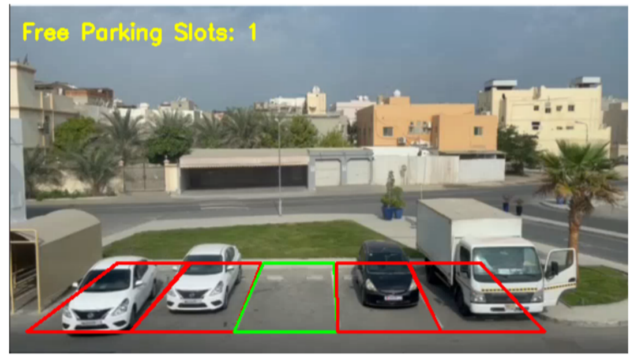

Figure 5. Parking area with four busy slots and one available slot

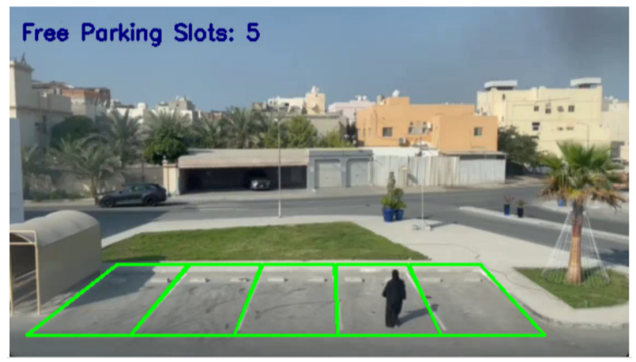

Figure 6. Edge detetcion technique

figure.

Figure 7. A person walking throught the parking slot

Another limitation of the proposed system, which relies on the number of white pixels, is that it may mistakenly classify parking slot as occupied under different weather conditions with rain spot present in the slot. This is because the image of the slot would have a significant number of white pixels, even if it does not actually contain a vehicle. Therefore, AI based on deep learning method was proposed as a robust alternative and it is developed as will be discussed in section 3. Although this method is easy to implement, eco-friendly and cost-efficient, it needs to be further improved in particular at evening or under low lighting conditions.

3. Proposed Deep Learning Based Smart Parking System

The following figure illustrates the block diagram of the system. The workflow starts with a selection process, where all parking slots are initially marked as free (represented by green color boxes). For the detection process, a live video stream of the parking area is captured using a webcam positioned at a high vantage point. The YOLO v3 algorithm, previously trained on the publicly available ”COCO” dataset containing various classes, including different types of vehicles, is utilized to detect vehicles in the parking area. Each frame of the video undergoes the detection process.

The threshold value for the area of an occupied parking slot is set to 1600. Additionally, initial threshold values are set for the confidence, score, and Intersection over Union (IOU). Instead of manually specifying the coordinates for irregularly shaped parking slots, rectangular regions are chosen as representations of the slots. These rectangles are selected within the slots, although they may not align precisely with the exact borders of the slots. The coordinates of these rectangles are then supplied to Python software, and after importing the necessary libraries, they are saved into variables. In the visualization, all the parking slots are marked in green to indicate their initial status as free slots as shown in Figure 9.

After loading the configuration and weight files of the pretrained YOLO v3 model, as well as the labels of the objects, a loop is created to perform object detection on each frame of the video. A 4D blob array is generated to store and process the frames, which are converted into input images. The blob is then set as the input to the network, and all the layer names are retrieved. Next, a for loop iterates over each output layer of the network. The object detection is refined by focusing only on specific types of vehicles such as cars, buses, trucks, motorbikes, and bicycles, while disregarding other objects. Within the loop, the class IDs, scores, and confidences of the detected objects are extracted. Only objects with a confidence equal to or above the defined threshold value are considered. The bounding box coordinates, class IDs, and confidences are updated accordingly in case any disregarded objects are encountered. To avoid multiple bounding boxes around a single object, a non-maxima suppression technique is applied. Bounding box rectangles are then drawn around the detected vehicles labeled with the word “busy”, as illustrated in the provided figure. A polygon is created for each bounding box using its coordinates and stored in a variable called ”POLYGON”. This polygon is then used to calculate the intersection area between each bounding box and each parking slot polygon. If the calculated area exceeds the threshold value for the occupied area, the slot is considered busy and marked in red, while also incrementing the busy counter. Otherwise, if the area is below the threshold, the slot’s status remains free, which is the default. The number of free parking slots is determined by subtracting the number of busy slots from the total number of slots, which is set at the beginning. As discussed before in section 1 and in [?]. YOLO algorithm examines the input image only once to make predictions. It achieves this by utilizing a single neural network and performing a forward propagation

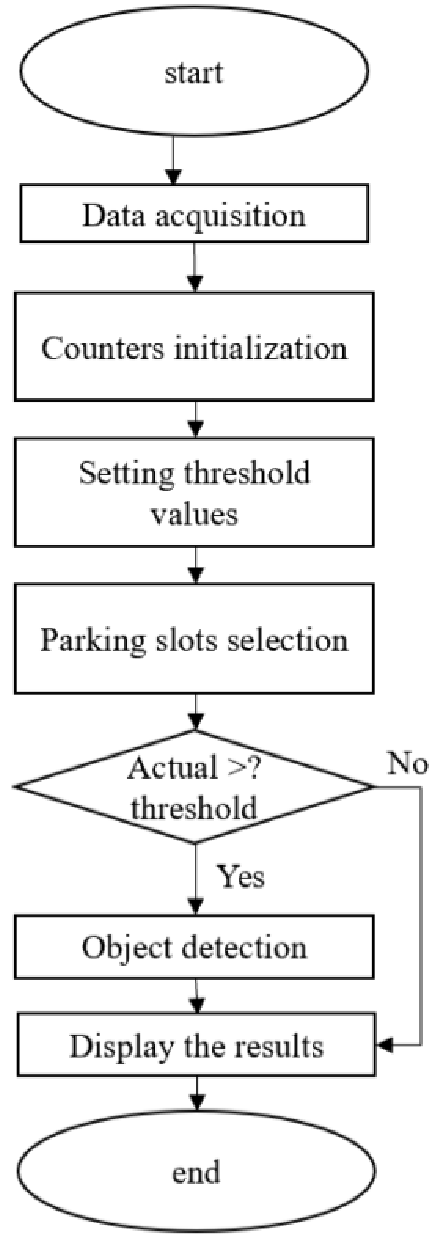

Figure 8. Block Diagram of the proposed / developed AI powered detection method

Figure 9. Parking slots selection

pass. The network divides the input image into grids, with each grid representing a bounding box. The probabilities for each region serve as the weights for the bounding boxes. To prevent multiple bounding boxes from being drawn around the same object, a technique called "non-maxima suppression" is employed. YOLO consists of 24 convolution layers and two fully connected layers, all of which are organized based on their specific use. YOLO v3 accepts RGB images with a size of 416x416 using convolutional layers. The last two dimensions of the output are flattened, obtaining an output of 19x19 grid cells, each containing 425 values. Within Darknet-53, 1x1 detection kernels are utilized with three different sizes. The architecture of YOLO v3 is visualized in the following figure [10].

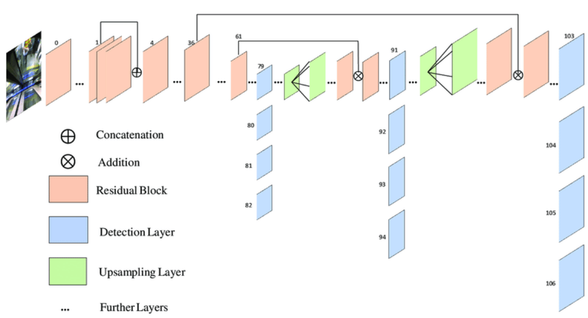

Figure 10. YOLOv3 architecture [4]

The proposed system was tested in The Market Mall parking area in Bahrain, and the results were satisfactory compared to other proposed systems as in [3]. However, it is worth noting that the compared system lacks the capability to detect vehicles in the parking area during nighttime. This limitation may be attributed to the dataset used, which does not include images of vehicles in various weather conditions. As a result, the compared system is unable to effectively detect vehicles at night. Figure 11 illustrates the test results of our model in a parking area with three occupied slots during nighttime. Although the confidence values obtained by our proposed system are relatively low in darker conditions, the system performs well in detecting vehicles at night as shown in the following figures.

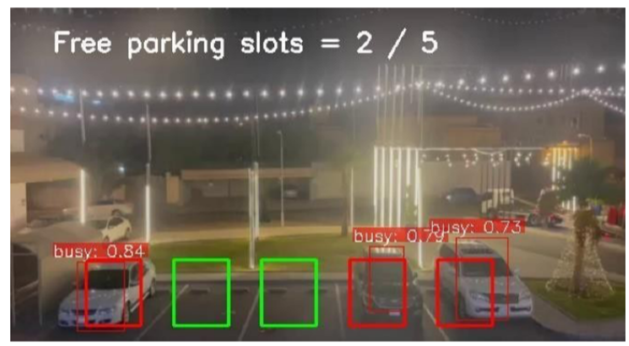

Figure 11. Parking area with three busy slots at different time

Figure 12. Parking area with three busy slots at different time

4. Large Parking Lots Using Divers Lighting Conditions

In this section, the detection method is tested on a larger parking lots with a binary classification Vacant /occupied and with vehicle type recognition (Five types) bus, car, bike, motorbike, Truck. that was included in coco datasets with and without using bounding boxes under different lighting conditions.

While applying the same technique in a larger parking area consisting of 66 slots [13] as shown in Figure 13, the model does not detect all the vehicles as shown in Figure 14. Although the results indicate that the system is performing well using YOLO v3 with an accuracy of 90%, the precision, recall, and F1-score were 1, 0.9, and 0.947, respectively.

YOLOv8 is the latest version of YOLO, launched on January 10th, 2023, by Ultralytics. It has been utilized here to train public datasets and has achieved high accuracy compared to previous versions. Developers can use YOLO v8 through the CLI or Python package, making it more convenient [14].

YOLOv8m has been trained and tested on the 66-slot parking area, obtaining high precision and recall values of 100% and 92.5%, respectively. The F1-score is

Figure 13. Large-scale parking area

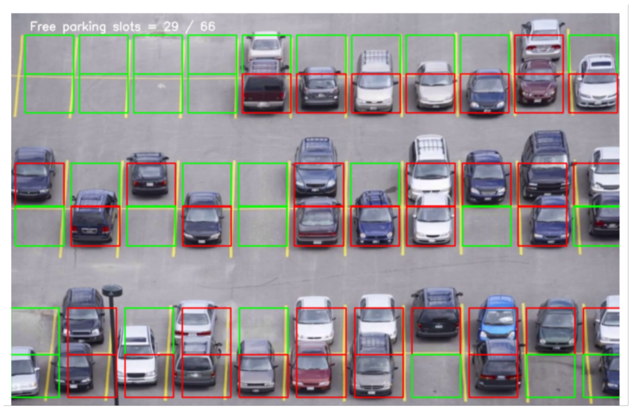

Figure 14. Detection result in a 66-space parking area using YOLO V3

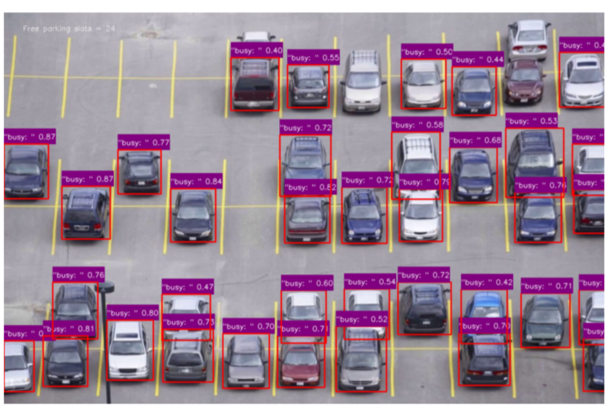

Figure 15. Detection result in a 66-space parking area using YOLO V8m

just above 96%, and the accuracy is 92.5%. The following figure shows the detection result.

The performance improved when using the pretrained YOLOv8s model with he same confidence threshold, the model exhibited better performance achieving a precision of 100% and a recall of 97.5%. The F1-score reached 98.7, and the accuracy also achieved 97.5%. The object detection result is visualized in the figure shown below.

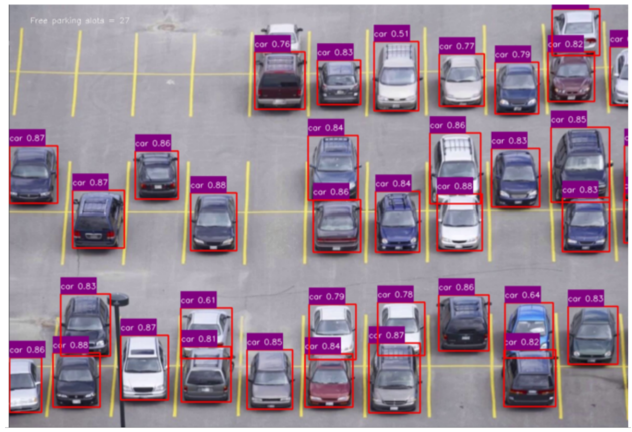

Figure 16. Detection result in a 66-space parking area using YOLO V8s

The following table represents the comparison among the used models based on the evaluation metrics in descending order (P: Precision, R: Recall, AP: Average Precision, F1-Score, Acc: Accuracy)

Next we investigated the orientation of the camera angle with the effect of low lighting. An advanced pretrained YOLO model, YOLO v8, was tested on Figure 17; a 40-space parking area with various types of vehicles at night with the same confidence threshold. Due to the camera angle, it is difficult to select each parking slot, so another selection technique is applied which is considering the whole area of the image excluding the area in the middle (outside the parking area) to be the region of interest. Because of the camera angle and the limited dataset, lacking images of vehicles from different perspectives, the model does not detect all the parked vehicles, although it performs well achieving an accuracy of 90%. The precision, recall and f1-score values are 89.2%, 100%, and 94.3%, respectively.

Figure 17. Detection in a 40-space parking area using YOLO v8s

TABLE II. Comparison of object detection metrics

Model	P	R	F1-score	Acc
YOLOv8s	1.000	0.975	0.987	0.975
YOLOv8m	1.000	0.925	0.961	0.925
YOLOV3	1.000	0.900	0.900	0.900

5. Conclusion and Future Work

Implementing the proposed system can pave the way for reducing parking cruising as the number of vehicles continues to increase. Drivers can benefit from the system by saving time, avoiding frustration while searching for parking spaces, and reducing fuel consumption. Additionally, the system contributes to protecting people from respiratory diseases by minimizing harmful emissions and promoting environmental conservation. While the proposed system successfully operates in different conditions and detects various types of vehicles, further improvements can be made. Integration of a user-friendly website to visualize the availability of parking slots and provide directions to the nearest open slot would enhance usability. Additionally, enhancing the system's confidence by incorporating more datasets with varied lighting conditions would be beneficial. Furthermore, automating the parking slot selection process would make the system scalable and more practical.

References

[1] J. C. X. Y. Yating Zhu, Xiaofei Ye and T. Wang, "Impact of cruising for parking on travel time of traffic flow," *Sustainability*, vol. 12, 2020.

[2] A. S. N. S. S. Awang Hendrianto Pratomo, Wilis Kaswidjanti and S. Saifullah, ""parking detection system using background subtraction and hsv color segmentation," *Bulletin of Electrical Engineering and Informatics*, vol. 10, 2021.

[3] G. R.-A. V. V. E.-G. Diana Laura Gómez-Ruíz, Daphne Espejel-Garcia and A. Villalobos, ""implementation of an available parking space detection system in hectic parking lots," *Journal of Transportation Technologies*, vol. 11, 2021.

[4] X. Ding and R. Yang, "Vehicle and parking space detection based on improved YOLO network model," *J. Phys. Conf. Ser.*, vol. 1325, no. 1, Oct. 2019.

[5] C.-J. Lin and J.-Y. Jhang, "Intelligent traffic-monitoring system based on YOLO and convolutional fuzzy neural networks," *IEEE Access*, vol. 10, pp. 14 120–14 133, 2022.

[6] P. P. Chantri Polprasert, Chaiyaboon Sruayiam and S. Teravetchakarn, """a camera-based smart parking system employing low-complexity deep learning for outdoor," 2019.

[7] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, Jun. 2016.

[8] D. F. Oncevarlikl, K. D. Yildiz, and S. Goren, "Deep learning based on-street parking spot detection for smart cities," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*. IEEE, Sep. 2019.

[9] I. H. Jung, J. M. Lee, and K. Hwang, "Smart parking management system using AI," *Webology*, vol. 19, no. 1, pp. 4629–4638, Jan. 2022.

[10] S. University, "Parking space classification using convolutional neural networks," http://cs231n.stanford.edu/reports/2016/pdfs/280_Report.pdf, accessed on October 25, 2023.

[11] P. R. L. de Almeida, J. H. Alves, L. S. Oliveira, A. G. Hochuli, J. V. Fröhlich, and R. A. Krauel, "Vehicle occurrence-based parking space detection," 2023.

[12] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *Eur. Transp. Res. Rev.*, vol. 11, no. 1, Dec. 2019.

[13] https://www.mottmac.com/transportation-planning/parking, [Accessed 12-04-2024].

[14] "Yolov8: A new state-of-the-art computer vision model," https://yolov8.com/, [Accessed 12-04-2024].