CAR PARKING SYSTEM

Manzo Clarence[1], Friendy[2], Amin Ahmed[3], Obai Ali[4]

[1] Computer Science, Multimedia University, Cyberjaya, Selangor, Malaysia
[2] Computer Science, Multimedia University, Cyberjaya, Selangor, Malaysia
[3] Computer Science, Multimedia University, Cyberjaya, Selangor, Malaysia
[4] Computer Science, Multimedia University, Cyberjaya, Selangor, Malaysia

ABSTRACT

A car parking system is a system that is used to manage the parking of vehicles in a designated area. This system is typically a combination of hardware and software components that work together to provide a convenient and efficient way to park vehicles. In this project, the software component is more focused rather than the hardware component. The software component is responsible for the management and control of the system, which for this project is the calculation of cars in the designated area, the calculation of empty parking space, the detection of parking alignment, and the calculation of cars by color in the designated area. The aim of a car parking system project is to design, develop, and implement a car parking system that meets the needs of users and is efficient and effective. The project involves the use of state-of-the-art technology and advanced algorithms to ensure that the system is reliable.

The benefits of a car parking system include increased convenience for users, improved safety and security, better management of parking resources, and reduced traffic congestion. With the implementation of a car parking system, users can park their vehicles with confidence, knowing that their vehicles are secure and that they can easily retrieve them when they are ready to leave. Additionally, the system helps to reduce traffic congestion by ensuring that vehicles are parked in designated areas and that there is no overcrowding.

In conclusion, the implementation of a car parking system is an important project that will provide many benefits to users, including increased convenience, improved safety and security, and reduced traffic congestion. By investing in a car parking system, organizations can help to ensure that their parking resources are effectively managed and that their users are provided with a convenient and efficient parking experience.

1. INTRODUCTION

Parking management has become a significant challenge in urban areas, as the increasing number of vehicles has led to a shortage of parking spaces and increased traffic congestion. In addition, traditional parking management methods don't have a car counter in a designated parking area or a counter for an empty parking slot. If these things are being done by humans, it will be time-consuming, inefficient, and can cause frustration for workers and customers. These challenges have motivated the need for a more efficient and effective solution: a car parking system.

The increasing demand for convenient and efficient parking solutions has motivated the development of car parking systems. The primary objective of this project is to address the challenges that arise from the increasing number of vehicles in urban areas, including traffic congestion, safety concerns, and the efficient use of parking resources.

With the implementation of a car parking system, the aim is to provide a seamless and hassle-free parking experience for users. The system will help to reduce traffic congestion and ensure that vehicles are parked in designated areas, leading to improved safety and security. Moreover, the system will help to tell the customers how many empty parking spaces are left, how crowded the parking space and which cars are not parked correctly with the parking alignment.

Additionally, the implementation of a car parking system will provide organizations with an opportunity to improve their reputation and enhance their customer service. The system will demonstrate an organization's commitment to providing its users with a convenient and efficient parking experience, which will ultimately lead to increased customer satisfaction and loyalty.

This project is planned to use some of the existing algorithms and methods around visual image processing problems. Which methods to use are up to the members to decide due to the fact that each member has a different task distribution of different car parking system's part. Obai Ali is responsible for creating an algorithm for counting the number of empty parking spaces. Amin Ahmed is responsible for creating the car counter algorithm. Friendy is responsible for creating an algorithm to detect and count the number of parked cars and moving cars. While Manzo Clarence is responsible for creating an algorithm for counting the number of cars by their colors.

2. BACKGROUND STUDY

Kaarthik K., Sridevi A., and Vivek C. make an intelligent parking system starting with an initialization process. [1] In this phase, they have drawn the parking slot manually to help identify the empty spaces later on. And then, they continued to acquire the input image data from CCTV cameras of a parking slot from multiple angles. During the image processing phase, they first did a segmentation process. They converted the images from RGB images to grayscale images and did a thresholding technique, and therefore created a final binary image. In the second phase, they did a morphology to enhance the binary image from unwanted noises and more. Lastly, they did car detection only if the edge and outline boundaries were obtained from the previous step.

Ms. S. Banerjee, Ms. P. Choudekar, and Prof. M.K. Muju took a similar approach to the previous authors. [2] First, images of car parking slots are obtained from CCTV cameras. Then convert the image from RGB to grayscale. But in this one, they did a Power law transformation (gamma correction) afterward. Lastly, they did a Prewitt edge detection operation to detect the cars. In their program, they created a system to allow a car to enter if the parking slot has empty slots. If the car is allowed to enter, it will first be guided to the left side of the parking space if the left side still has some empty slots. If it is full, it will guide the car to the right side of the parking space. Each side only has 10 parking slots. This means, there will be 20 parking slots in total.

H. Al-Kharusi and I. Al-Bahadly took the image of an empty parking space instead of drawing it manually. [3] They use a green circle from the RGB value to mark the empty car spaces. Then they convert the image into HSV and convert it again to grayscale. Then they did a threshold and eroded the empty parking space image. After finishing with the empty parking space image, to obtain the image dataset, they also use a CCTV camera that has no obstructing object. Then, to differentiate objects, see the cluster form in a 3-dimensional graph of an RGB image. They also use multiple threshold values to threshold the part of the image where it is dark or bright. Then a closing operation is performed for image development. By using the coordinates from the initialized empty car park, each car park slot can be analyzed to determine whether a car is there or not. And if the object is big enough, it can be considered as a car and therefore the parking slot will be occupied.

M.Y.I. Idris, E.M. Tamil, Z. Razak, N.M. Noor, and L.W. Kin have made a very interesting and advanced program. [4] Their system is not just detecting cars, but also tells a customer about any parking space vacancy within a car park area. Then the customer can choose the empty parking slots within the area with graphical instructions. And if the customer didn't take any decision in 30 seconds, the program will make an assumption based on the shortest path. There are a lot of features to be mentioned in this program, but that would be too much. The program contains a lot of modules, such as a database updater, image processing, short path, timer, and I/O. Inside each module, there are still many kinds of functions as well. For the purpose of this background study, we would like to focus more on the image processing part. A threshold technique is being done where it will subtract the new incoming image from the background image. Since the program has made an application where the admin can choose a coordinate of the parking slot, it will compare the pixels of that coordinate only instead of the whole parking space. There are not many visual image processing techniques that were elaborated since they were more focused on the application.

Bilel Benjdira, Taha Khursheed, Anis Koubaa, and Kais Ouni, make an algorithm comparison between Convolutional Neural Network and Yolo V3 in the object detection method [5], the input is they retrieve using a video which is taken by a drone, the collection of the image is divided into two sections, which is a training set includes 218 images, 3365 instances, and labeled cars, and a test set includes 52 images and 737 instances of cars. In the training and testing part, they train 200K steps, and use optimized Faster-RCNN and YOLO v3, by setting the right amount of learning rate, maximal object per class and limiting the batch size, and resizing the input image taken by the drone. The result is that both algorithms achieve high accuracy but the performance time affects the Faster-RCNN to consume more time to detect the testing object.

Ersin Kilic and Serkan Ozturk are using the Convolutional Neural Network for car counting. [6] The input of the images is captured using a 40m height fixed camera, they proposed to build a heatmap learner convolutional neural network (HLCNNs), the backbone of the model, consisting of a convolutional layer, adaptation layer. Then they set the VGG-16 model with 3 max pooling layers to 16, 8, and 4 downsampling ratios, then used Adam optimizer to train 3 models which is the previous downsampling ratio, and conclude that 8 downsampling ratios achieve the best score.

Mohd Saad Umair Ansari, Mayur Chaudhary, and Aayush Agarwal create a counting car on the road [7], they start with background subtraction, which subtracts the car from the background, grayscale the image, reason because it is easy and faster the computing process, then add gaussian blur, so the noise in the image is gone, the morphing and dilate the result, then use cv2 built-in contour to detect a car and apply deep learning, which called "region of interest", with this deep learning the car can't be wrongly classified to be another object, and finally they add a threshold to eliminate unrelated color. The result of the highest error in the car counting system is 34%, which is above-average accuracy.

Jae Kyu Suhr and Ho Gi Jung use a fused sensor very advanced to detect an unoccupied parking slot detection and tracking system [8], with the help of the camera, the system can mark the empty parking lot. To prevent missing parking slots from getting marked wrongly or falsely detected, they use Around View Monitor (AVM) to help. The camera is also used for predicting the position of the current position in the parking area. If more than 2 camera is detected, the AVM system can create a 3D point of view so it will increase the accuracy of the position of the car and keep tracking the parking lot, combining the AVM and camera system sequentially and result from the classification decider, is the slot if occupied or not.

3. APPROACH

Because this project is an application-based type, we would like to try to re-implement a car parking system where the features and function of the car parking system are our own idea. Some background study from previous works about creating a car parking system is being done to guide or inspire us to implement a car parking system with a new modified approach. The car parking system will be able to do 4 main functions, which are detecting and counting cars in a parking space, detecting and counting empty parking slots in a parking space, detecting and counting parked cars and moving cars, and detecting and counting cars by their color in a parking space.

First, the program will read the whole input image dataset from a folder into a list called "image_list". Then the program will also read all the ground truths files which will be in the CSV format that we have made manually.

3.1. Detect and count cars by their color

"Colour_Groundtruth.csv" is for the ground truth file of the total number of cars by their colors in a parking space. For this program, it will only detect the 3 colors that most frequently appear in the dataset, which are red, black, and white. The program will detect one color at a time. It will initialize three lists to store each of the colors detected and count the total number of cars. The names of the lists are "redCount" for counted red cars, "whiteCount" for counted white cars, and "blackCount" for counted black cars.

First, to detect the red colored car, use a SLIC superpixels region segmentation with total segments 600, compactness 30, sigma 1, and labeled it with its average colors. Then, use thresholding with threshold value 120, maximum value 255, and threshold type "thresh binary". After that, mask the red colors with the specified upper and lower bound. Finally, count the number of red cars with 8 connectivity connected components and store it into "redCount".

Second, to detect the white colored car, mask the white colors with the specified upper and lower bound. This time, since there will be a lot of unwanted noises, a closing operation should be done. And then dilate vertically to combine the body of the car and the front part of the car that were separated by windshield. Finally, count the number of white cars with 8 connectivity connected components and store it into "whiteCount".

Third, to detect the black colored car, use a SLIC superpixels region segmentation with total segments 600, compactness 30, sigma 1, and labeled it with its average colors. After that, mask the black colors with the specified upper and lower bound. A closing operation should be done to remove all the unwanted noises. Finally, count the number of black cars with 8 connectivity connected components and store it into "blackCount".

3.2. Detect parked car and moving car

The file named "Cars_Groundtruth" is the ground truth for the total number of parked cars and moving cars. The file only includes 3 data, which all refer to total car parking, total moving car, and total car that produce each image respectively.

The first task is to detect the parked car and moving car by creating a mask first, to select which we want to choose for detecting. In this system, we create 2 masks, and later the mask is used for car parking detection and moving car detection. The next process is to apply the mask into the image and then apply a threshold to the image, this is being done to eliminate unwanted color. Then, use a contour for detecting a car in the parking position or in moving position separately. Then filter out the contour so it doesn't get a bad result, like detecting a wrong object as a car. Lastly, count the contour and store it into the "moving_car" and "parking_car" list respectively.

3.3. Detect cars using YOLO algorithm

"Yolo_Groundtruth" is the name of the ground truth file of total number of cars and it include only one type of data which point to the number of cars, To detect and count cars in the dataset, we will use the You Only Look Once (YOLO) object detection algorithm. YOLO is a real-time object detection system that can detect and classify objects in an image or video sequence. We will use the YOLOv3 model, which has been trained on the COCO dataset and can detect 80 different classes of objects, including cars.

The first step in our approach will be to load the YOLOv3 model and the COCO class names. we will convert each image into a blob, which will be fed into the YOLOv3 model. The model will then detect and classify any objects in the frame, and we will extract the bounding boxes of the detected cars.

Next, we will apply non-maximal suppression to remove overlapping bounding boxes and retain only the bounding box with the highest confidence score for each car. This will help to avoid counting the same car multiple times. Finally, we will draw rectangles around the cars and annotate each rectangle with the label "Car". We will also keep track of the number of cars detected in each image of this information in a list.

4. EXPERIMENT

The dataset itself is obtained from an online source. The website is called Kaggle. However, the original dataset contains a very large number of images that can go to thousands of images. Therefore, we have cut down the dataset into just 600 images so it can be computed by the program more efficiently and effectively. As mentioned before, the ground truth dataset is obtained by counting the cars manually by humans. The ground truth dataset is not an image dataset, it is a list of numbers dataset which will be stored in a CSV file format. To evaluate the results, we will compare the detected or estimated total number of cars with the ground truth numbers using a line plot comparison. By doing such, we can compare and see the differences between the estimated and the ground truth data visually. To know the exact statistical number of the differences between the estimated and the ground truth data, we use an euclidean distance to measure how far is the difference between those data.

4.1. Detect and count cars by their color

The line plots comparison for red colored cars, white colored cars, and black colored cars can be seen in figure 4.1.1, figure 4.1.2, and figure 4.1.3. For the purpose of analysis, a sample of 200 images is taken from 600 images. Visually, we can see that the program has detected far from the ground truth values. However we can see a bit of similar patterns between the estimated numbers and the groundtruth numbers, where when the ground truth numbers are dropping, the estimated numbers are also dropping, and when the ground truth numbers are rising, the estimated numbers are also rising.

To know the exact statistical numbers, the euclidean distance for red colored cars is 130.6101, the euclidean distance for white colored cars is 180.5076, and the euclidean distance for black colored cars is 359.8555. From these numbers, we can tell that detecting red colored cars is the most accurate although it is still not that accurate, while detecting black colored cars is the worst accurate.

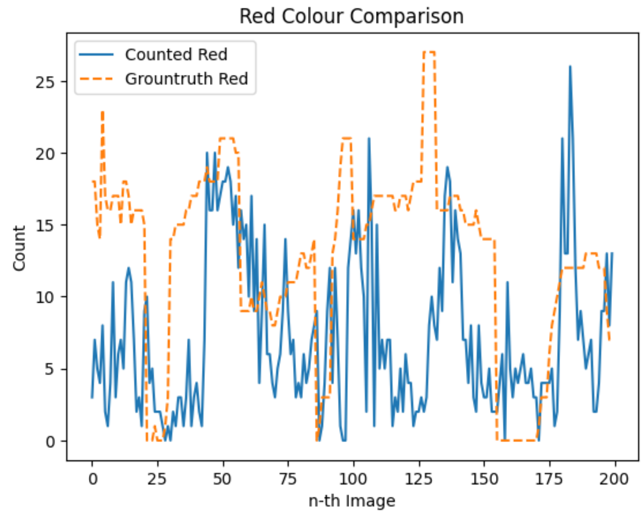


Figure 4.1.1

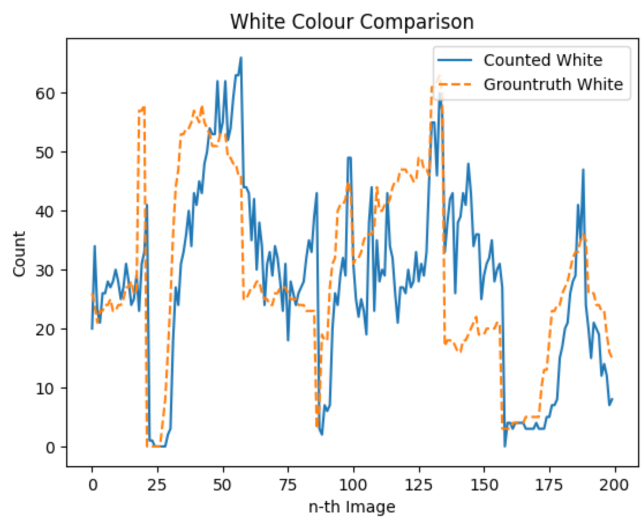


Figure 4.1.2

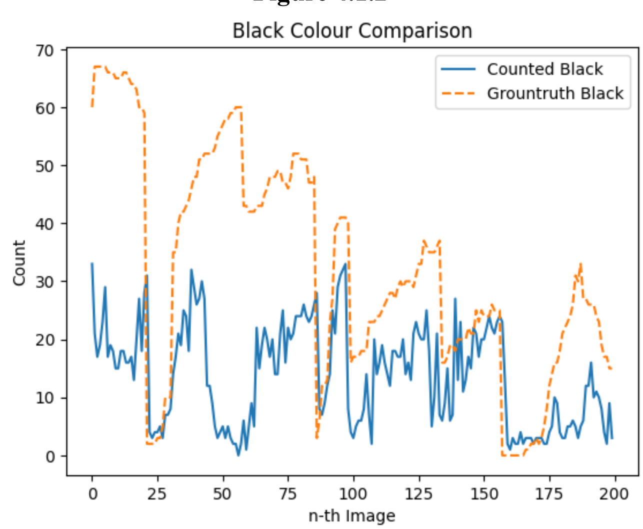


Figure 4.1.3

The comparison of the original images and the resulting images of the detected red colored cars can be seen in figure 4.1.4 and figure 4.1.5.

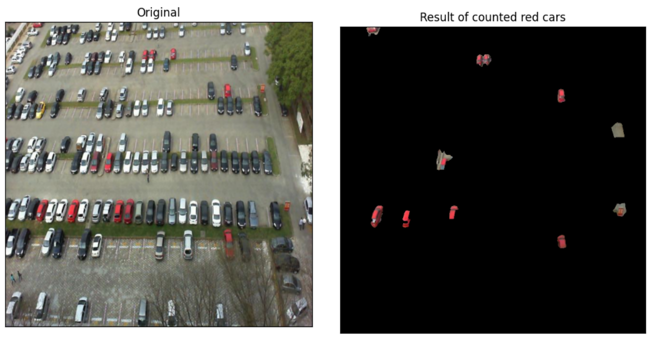


Figure 4.1.4

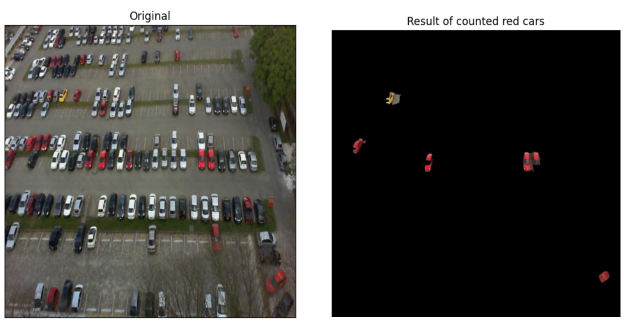


Figure 4.1.5

The comparison of the original images and the resulting images of the detected white colored cars can be seen in figure 4.1.6 and figure 4.1.7.

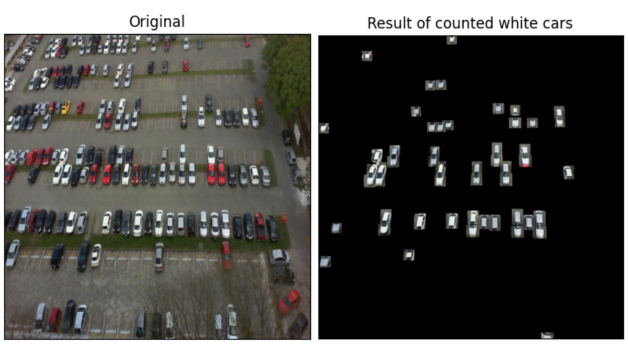


Figure 4.1.6

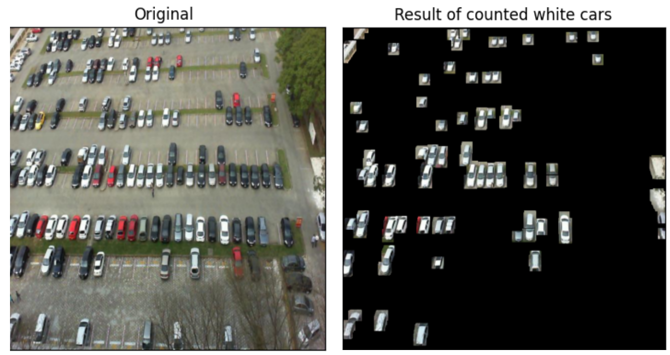


Figure 4.1.7

The comparison of the original images and the resulting images of the detected black colored cars can be seen in figure 4.1.8 and figure 4.1.9.

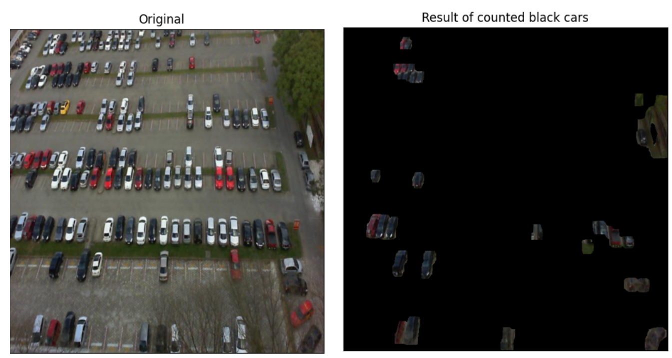


Figure 4.1.8

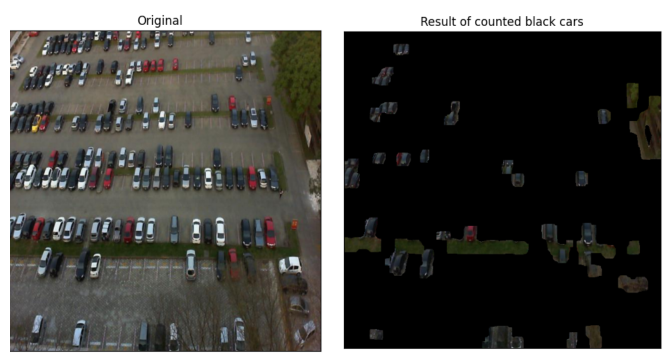


Figure 4.1.9

4.2. Detect parked car and moving car

Figures 4.2.1, 4.2.2, and 4.2.3 show comparison of line plots for parked vehicles, moving cars, and number of cars. A sample of 200 photos from a total of 600 photos was selected for examination. Visually, we can see that the system has a lot of false detection, especially in the moving car, the truth and the estimated value is quite high in the comparison, but, we can detect certain similarities between the estimated number and the basic truth number.

To know the exact statistical numbers, the comparison of the test and the ground truth has been done, the calculated euclidean distance for total counted cars is 1108.432, the euclidean distance for total parking cars is 1013.518, and the euclidean distance for total moving cars is 151.930. From the total moving cars, we can see that the total number of car has the best result, but the total number of moving cars and parking cars has a similarity result, which is a bad distance.

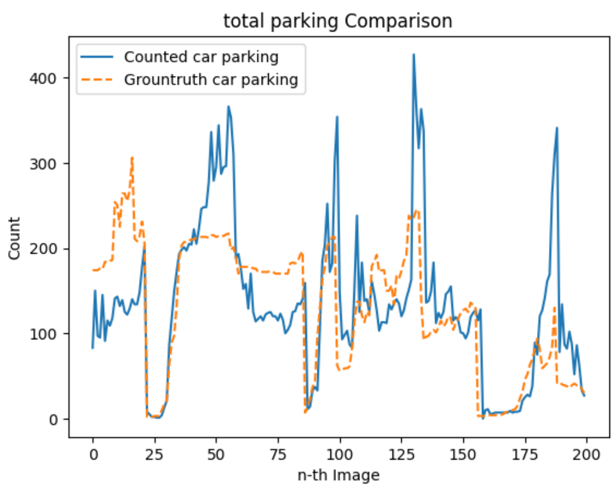


Figure 4.2.1

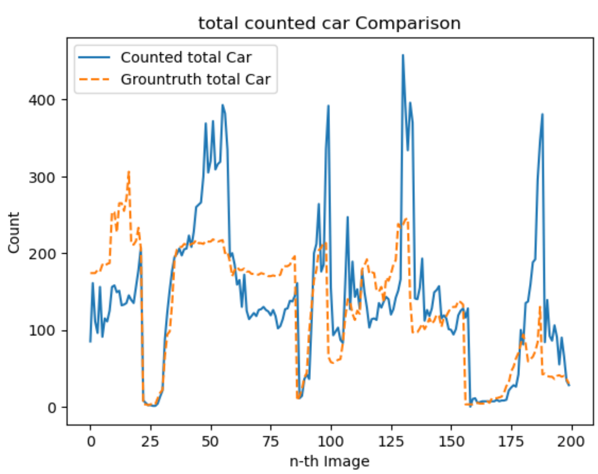


Figure 4.2.2

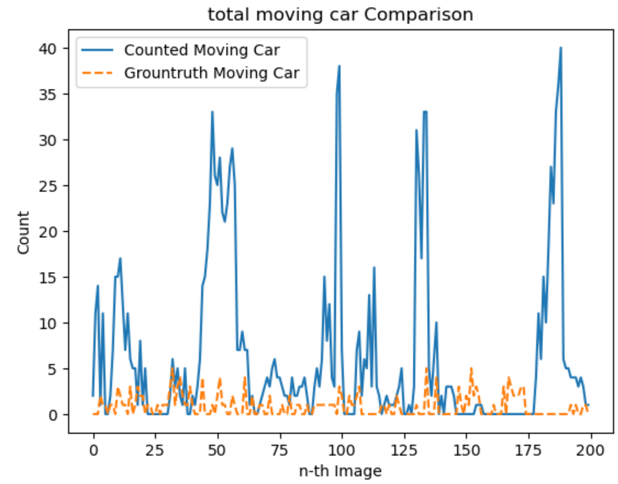
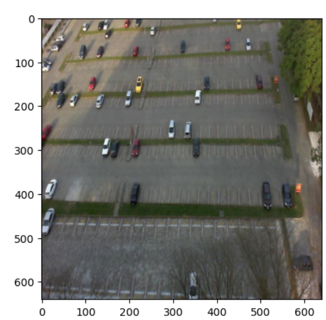


Figure 4.2.3

The comparison of the original images and the resulting images of the detected parking cars can be seen in figure 4.1.4 and figure 4.1.5.



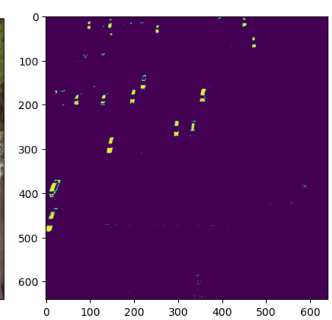
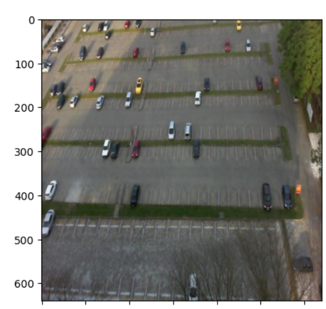


Figure 4.1.4 **Figure 4.1.5**

The comparison of the original images and the resulting images of the detected parking cars can be seen in figure 4.1.6 and figure 4.1.7.



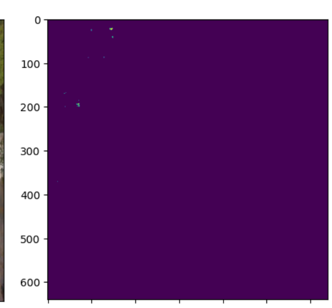


Figure 4.1.6 **Figure 4.1.7**

4.3. Detect cars using YOLO algorithm

Figure 4.3.1 shows comparison of line plots for cars, Out of a total of 600 images, 200 were chosen as a sample for analysis. Visually, we can see that the system makes a lot of false detections, particularly when a car is driving. However, despite the fact that the estimated value and the fundamental truth value are relatively similar, the truth and estimated values are quite high in comparison.

The calculated euclidean distance for all counted cars is 2117.37, To know the actual statistical numbers, the comparison of the test and the ground truth has been done.

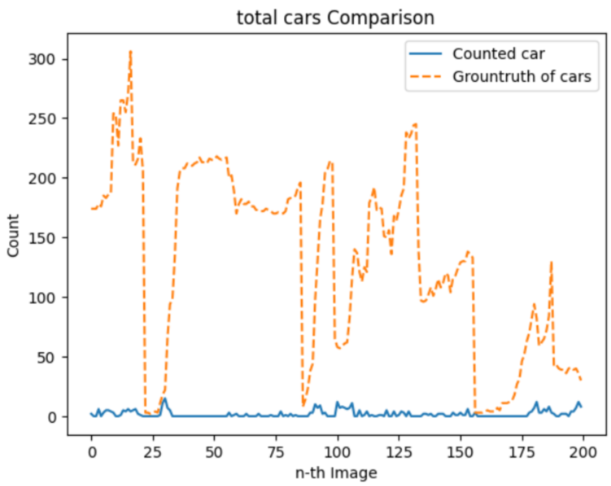


Figure 4.3.1

Show the output as figure 4.3.2 , 4.3.3 and 4.3.4 point to the variety of the result and how different it could be in terms of accuracy.

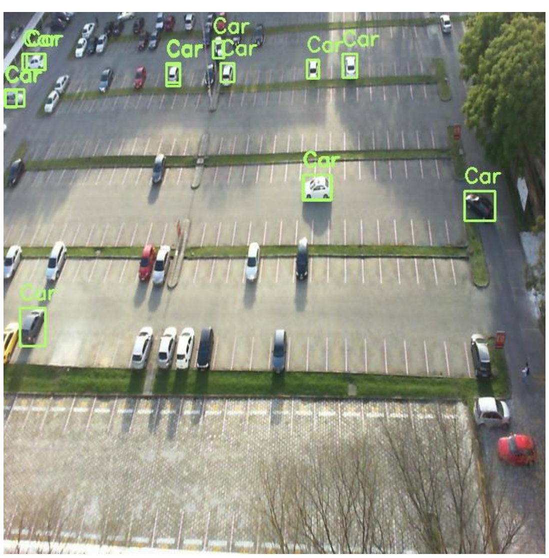


Figure 4.3.2

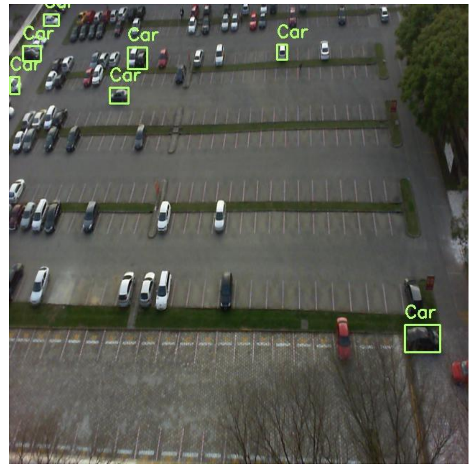


Figure 4.3.3

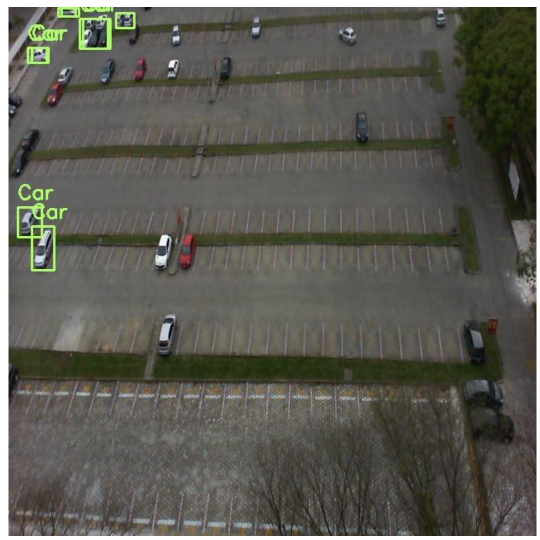


Figure 4.3.4

Figure 4.3.5 and 4.3.6 represent the type of error that can be given as result.

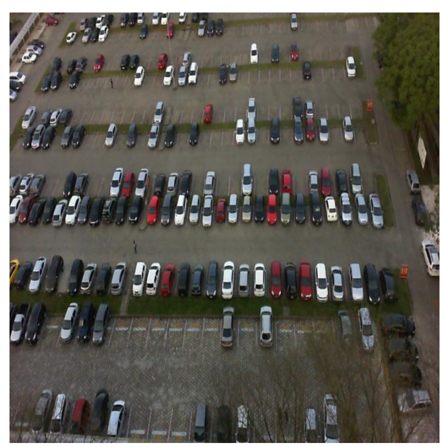


Figure 4.3.5

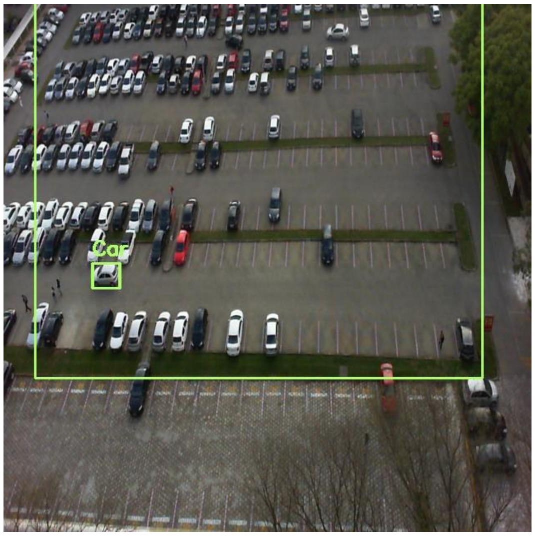


Figure 4.3.6

5. CONCLUSION

In conclusion, the car parking system project has been a success in providing a solution to automate the process of counting number of cars in a parking space, number of empty parking slots in a parking space, number of cars that are not in the parking alignment, and number of cars by their color in a parking space. The use of visual image processing has made the process faster and more convenient for customers. The project team has worked hard and tried their best to ensure that the system meets all the necessary standards and regulations. The car parking system is a prime example of how visual image processing can be used to solve practical problems and improve the daily lives of people.

Although the car parking system is already usable for real-life implementation, the accuracy of the car parking system is still far from good. There are still a lot of future improvements to be done to increase the efficiency and the accuracy of the car parking system. For future ideas, we could use machine learning to train the model and therefore can detect the car efficiently and accurately. Or we can create the User Interface so the operator of the parking system can manage the program more easily rather than using the python jupyter notebook which is not a user friendly interface. Other than that, we can build a better car detection system that could be a real-time car detection system rather than detect a car parking space after the event occurs.

6. REFERENCES

[1] Kaarthik K., Sridevi A., and Vivek C. (2017). Image Processing Based Intelligent Parking System. *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, DOI:10.1109/ICEICE.2017.8191876

[2] Ms. S. Banerjee, Ms. P. Choudekar, and Prof. M.K. Muju. (2011). Real Time Car Parking System Using Image Processing. *2011 3rd International Conference on Electronics Computer Technology,* DOI: 10.1109/ICECTECH.2011.5941663

[3] H. Al-Kharusi and I. Al-Bahadly. (2014). Intelligent Parking Management System Based on Image Processing. *World Journal of Engineering and Technology*, 2014, 2: 55-67. DOI: 10.4236/wjet.2014.22006

[4] M.Y.I. Idris, E.M. Tamil, Z. Razak, N.M. Noor, and L.W. Kin. (2009). Smart Parking System using Image Processing Techniques in Wireless Sensor Network Environment. *Information Technology Journal*, 8: 114-127. DOI: 10.3923/itj.2009.114.127

[5] Bilel Benjdira, Taha Khursheed, Anis Koubaa, Adel Ammar, and Kais Ouni. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. *Processing of the 1st International Conference on Unmanned Vehicle Systems (UVS),* DOI :978-1-5386-9368-1/19/$31.00@ 2019 IEEE

[6] Ersin Kilic and Serkan Ozturk. (2021). An accurate car counting in aerial images based on convolutional neural networks. *Journal of Ambient Intelligence and Humanized Computing,* DOI: 10.1007/s12652-021-03377-5

[7] Mohd Saad Umair Ansari, Mayur Chaudhary, and Aayush Agarwal. (2021). TRAFFIC MANAGEMENT BY VEHICLE COUNTING SYSTEM USING IMAGE PROCESSING, *International Research Journal of Modernization in Engineering Technology and Science Volume:03/Issue:06/June-2021.*

[8] Jae Kyu Suhr and Ho Gi Jung. (2014). Sensor Fusion-Based Vacant Parking Slot Detection and Tracking, *INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 15, NO. 1.* DOI: 10.1109/TITS.2013.2272100