

**ADVANCED SCHOOL OF SYSTEMS AND DATA STUDIES (ASSDAS)**

**DEPARTMENT OF COMPUTER SCIENCE**

**COMPUTER VISION – CS308**

**CAC**

**CASE STUDY THE KUMASI SMART PAID-PARKING-SYSTEM**

**MARSHAL CHIJIOKE NNEJI**

**(ADS16A00084Y)**

DATE: **December, 2022**

**Answers**

**Question 1**

The Kumasi Paid Parking scheme became operational on June, 2 after some three (3) months of intensive public awareness campaigns on local radios, posters, and with stakeholders and interest groups. The Kumasi Metropolitan Assembly (KMA) engaged Gold Street Real Estate limited the scheme operator through an open competitive bidding process for 40% of gross incomes. Although other paid-on-street parking schemes are operational in Accra and Takoradi, the Kumasi scheme is so far established. The area where the scheme is operational has a concentration of shops, restaurants, banks, and offices which can best be described as a pedestrian open mall. There are 1200 on-street stalls and three (3) street sites with a total capacity of 1400; The Prempeh Assembly Hall (capacity of 200 vehicles) and KMA Central Car Parks (1200) are very remote from the CBD. Patrons of the Central Car Parks are transported in a bus procured from the scheme proceeds. The prison’s car park is also within the parking system but it is the least developed with a capacity of 30 cars.

Kumasi paid pack does not as yet have parking garages for off-street parking. Two surface car parks are operational but shoppers prefer to park on street because of convenience and closeness to one’s destination. As of now shoppers perceive that it is difficult to find parking at KMA nearer to one’s destination, a situation reported was worsening congestion in KMA as drivers drove slowly for several minutes looking for parking which was usually found at locations remote from their ultimate destinations.

Few people however remarked that the data used for the comparison (before) were more than four years old and there could have been some improvement in the situation before inception.

**Solution with computer vision (Automated parking)**

Cameras have become popular in cars, with a rear-view camera being the minimum and full surround-view camera systems at the top end. Automotive camera usage began with single viewing camera systems for the driver. However, both the number of cameras and the number of Advance Driver Assistance Systems (ADAS) applications made possible with automotive cameras have increased rapidly in the last five years, mainly because the processing power has increased during this period to enable high levels of real-time processing for computer vision functions. Some examples include applications such as back-over protection, lane departure warning, front-collision warning, or stereo cameras for more complete depth estimation of the environment ahead of the vehicle.

Automated parking is a good commercial starting point to deploy automated driving in a more restricted environment like KMA. Firstly, it involves low-speed maneuvering with a low risk of high-impact accidents. Secondly, it is a more controlled environment with fewer scene variations and corner cases. Stable deployment of automated parking in the real world and analysis of performance statistics is an important step towards going to higher levels of autonomy. The first-generation parking systems were semi-automated using ultrasonics or radar. Cameras are recently augmenting them to provide a more robust and versatile solution.

**Question 2**

The possible solution (automated parking) can reduce the number of illegal and wrongful parking in KMA.

The challenge of parking assistant systems is to rely on and accurately detect parking slots to allow parking maneuvers with a minimum number of independent movements. The aim is to provide a more complete review of the use of computer vision in parking. Also, the objective of an automated parking system is to deliver a robust, safe, comfortable, and most importantly useful function to the driver enabling time-saving, accurate, and collision-free parking. Current systems on the market rely solely on range sensor data, typically ultrasonics, for slot detection, remeasurement, and collision avoidance during automated parking.

Opportunities:

* The price of a single parking space in the SMART system can be lowered by up to 50% in comparison to the price of building an underground parking space.
* THOUGHTFUL INVESTMENT. Installation of Smart Parking in the KMA car park location, which is the bane of drivers, is a guarantee of success and big profits.
* OPTIMIZED PARKING – Users find the best spot available, saving time, resources, and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.
* Smart Parking is a SELF-FINANCED project. The introduction of parking charges allows for a return on the investment even within a few years.
* MORE COMMERCIAL INVESTMENTS in city centers. The use of Smart Parking is a possibility to change the allocation of huge squares from the KMA car parks to new investments.
* REDUCED TRAFFIC – Traffic flow increases as fewer cars are required to drive around in search of an open parking space
* INNOVATIVE PARKING WITH MODERN ADVERTISEMENT. Installation of the LCD screen, LED display, or backlit large format advertisements is a guaranteed profit, faster return on the investment, and an increase of interest among customers.
* INCREASED SAFETY – Parking lot employees and security guards contain real-time lot data that can help prevent parking violations and suspicious activity. License plate recognition cameras can gather pertinent footage. Also, decreased spot-searching traffic on the streets can reduce accidents caused by the distraction of searching for parking.
* INTEGRATED PAYMENTS AND POS – Returning users can replace daily, manual cash payments with account invoicing and application payments from their phone. This could also enable customer loyalty programs and valuable user feedback.

**Question 3.**

Flow diagram of how image processing will be used;

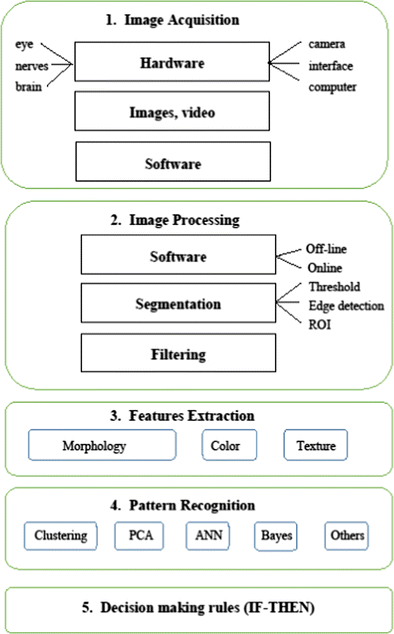


Image Acquisition: Image acquisition often entails the capture of an image by a sensor, like a camera. If there is an output that isn't digital, an analog-to-digital converter is used to change it. Pre-processing, such as image scaling, is also a part of this process.

Image processing: Image processing, as the word suggests, is applying different algorithms and techniques to manipulate or modify the image for making it suitable for the task and use case. Almost every one of us has used image processing for one task or another, for example, when we use portrait mode to click a selfie, we are using image processing to blur the background.

Features extraction: In computer vision, a feature is a measurable piece of data in your image which is unique to this specific object. It may be a distinct color in an image or a specific shape such as a line, edge, or an image segment. A good feature is used to distinguish objects from one another. For example, if I give you a feature like a wheel, and ask you to guess whether the object is a motorcycle or a dog. What would your guess be? A motorcycle. Correct! In this case, the wheel is a strong feature that clearly distinguishes between motorcycles and dogs. If I give you the same feature (a wheel) and ask you to guess whether the object is a bicycle or a motorcycle. In this case, this feature isn’t strong enough to distinguish between both objects. Then we need to look for more features like a mirror, license plate, or maybe a pedal that collectively describes an object.

Pattern recognition: Pattern recognition is the identification and classification of input data, such as text, speech, and images, using machine learning algorithms by delineation of patterns in the given data.

**Question 4**

A camera needs some sort of measurable energy to take a picture. In this context, light or, more generally, electromagnetic waves, are the energy of interest. A photon, which has no mass and whose electric and magnetic flux varies sinusoidally, is frequently used to describe an electromagnetic wave (EM wave). A photon is frequently characterized in three ways:

1. A photon can be described by its energy E (measured in eV)
2. A photon can be described by its frequency f(H2)
3. A photon can be described by its wave length λ(m) E = (hc) / λ E = hf

The most important method of image sensing and acquisition is the quantum detector, which depends on the energy of absorbed photons being utilized to boost electrons from their stable state to a better condition above a threshold of energy. When this occurs, the material's properties are changed in a quantifiable way. Planck and Einstein established the following relationship between the incident photon's λ and the E it carries:

E = (hc) / λ

On impact the photon transfers every or none of this quantum of energy to the electron.

The images are produced by a combination of a lighting source and energy that is reflected or absorbed by the components of the scene being photographed. Radar, X-ray, infrared, ultrasonic, computer-generated energy patterns, and other sources of energy could all be used to create lighting. We utilize a sensor compatible with the characteristics of illumination to sense the image. Image acquisition is the process used with image sensors.

Flame power is altered in a digital image via the sensor. The idea is that the combination of input electrical energy and sensor material that is aware of the real energy that is being monitored converts incoming light energy into voltage.

2-D function f represents an image (x, y). An image practically must be a non-zero, finite quantity, which is:

1. < f (x, y) < “*infinite*

It is also discussed that for an image f (x, y), we have two factors:

* The amount of source illumination incident on the scene being imaged. Let us represent it by: a (x, y).
* The amount of illumination reflected or absorbed by the object in the scene. Let us represent it by: b (x, y).

Then f (x, y) can be represented by:

f (x, y) = a (x, y). b (x, y)

Where 0 < a (x, y) < “*infinite*

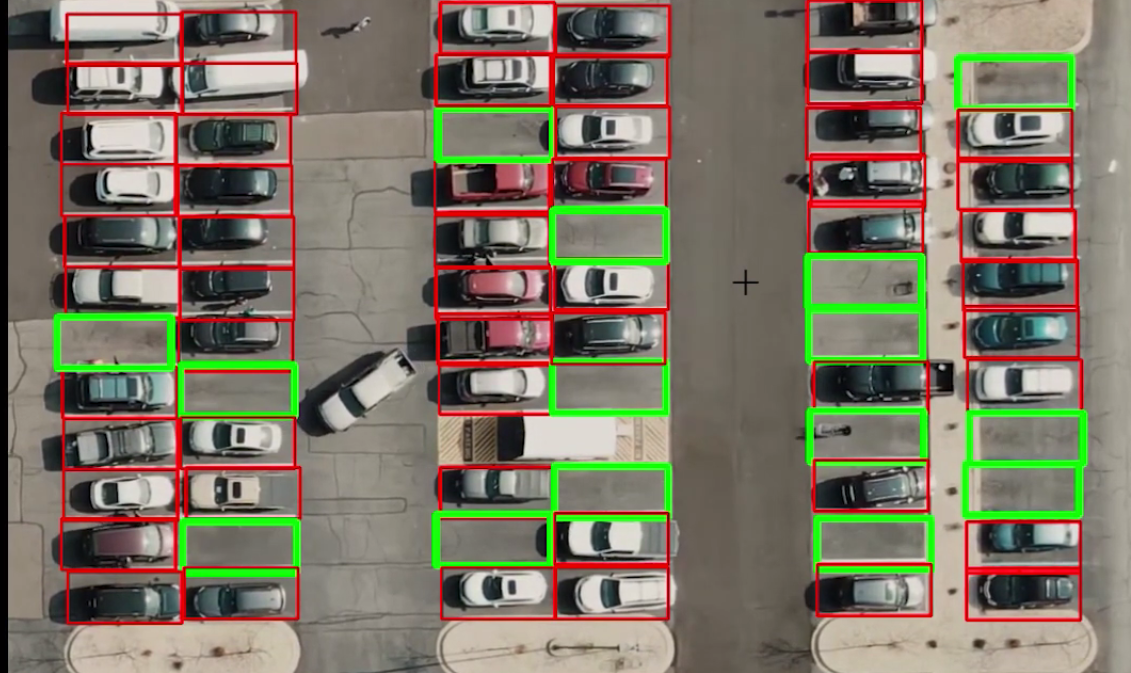
It implies that illumination will be a non-zero, finite quantity and that its amount depends on the source of illumination.

and 0 < b (x, y) < 1

Here 0 indicates no reflection or total absorption and 1 means no absorption or total reflection.

**Acquisition process;**

* Typically, image acquisition involves capturing an image by a sensor such as a camera. If a non-digital form of output exists, it is converted to a digital form using an analog to a digital converter. This process also includes pre-processing, such as image scaling.
* The process that is related to image manipulation to achieve relevant results for specified tasks to be performed is known as image enhancement. Ideally, this process relates to image filtering by performing tasks such as noise removal, contrast adjustment, brightness, and sharpening of the images for improving the quality of the image that were captured originally.
* Image restoration involves improving the appearance of an image that may have been degraded by mathematical and probabilistic models. An ideal example would be the reduction of blurring in an image.
* The extraction of features from an image with a color-based approach.
* It involves representing images in terms of various resolution available that is generally used for image compression. This is useful for image data compression as well.
* The representation is associated with displaying image output in the form of a boundary or a region. It can involve characteristics of shapes in corners or regional representations like the texture or skeletal shapes. On the other hand, the description is most commonly known as feature selection, responsible for extracting meaningful information from an image. The information extracted can help to differentiate between classes of objects from one another accurately.
* The process of assigning labels to an object depending on its description for classification purposes. This is a very important step for Computer Vision. To train models, a large enough corpus of images needs to be processed and labelled, so that the Computer Vision model can be utilize to detect similar objects in other images.



**Question 5**

Image Degradation comes in many forms such as motion blur, noise, and camera misfocus. In cases like motion blur, it is possible to come up with a very good estimate of the actual blurring function and "undo" the blur to restore the original image. In cases where the image is corrupted by noise, the best we may hope to do is to compensate for the degradation it caused. In this project, i will introduce several of the methods and also implement the method used in this project to restore the images identified.

**Methods**

Inverse Filter: In this method we look at an image assuming a known blurring function. We will see that restoration is good when noise is not present and not so good when it is.

Weiner Filtering: In this section we implement image restoration using wiener filtering, which provides us with the optimal trade-off between de-noising and inverse filtering. We will see that the result is in general better than with straight inverse filtering.

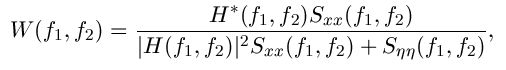
Wavelet Restoration: We implement three wavelet-based algorithms to restore the image.

Blind Deconvolution: In this method, we assume nothing about the image. We do not have any information about the blurring function or on the additive noise. We will see that restoring an image when we know nothing about it is very hard.

**Method used**

The wiener filtering will be used.

The Wiener filtering is optimal in terms of the mean square error. In other words, it minimizes the overall mean square error in the process of inverse filtering and noise smoothing. The Wiener filtering is a linear estimation of the original image. The approach is based on a stochastic framework. The orthogonality principle implies that the Wiener filter in the Fourier domain can be expressed as follows:



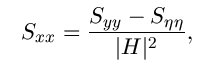
where  are respectively power spectra of the original image and the additive noise, and is the blurring filter. It is easy to see that the Wiener filter has two separate parts, an inverse filtering part and a noise smoothing part. It not only performs the deconvolution by inverse filtering (high pass filtering) but also removes the noise with a compression operation (low pass filtering).

**Implementation**

To implement the Wiener filter in practice we have to estimate the power spectra of the original image and the additive noise. For white additive noise the power spectrum is equal to the variance of the noise. Many methods can be used to estimate the original image's power spectrum. A direct estimate is the periodogram estimate of the power spectrum computed from the observation:



where Y (k, l) is the DFT of the observation. The advantage of the estimate is that it can be implemented very easily without worrying about the singularity of the inverse filtering. Another estimate which leads to a cascade implementation of the inverse filtering and the noise smoothing is



which is a straightforward result of the fact:  The power spectrum can be estimated directly from the observation using the periodogram estimate. This estimate results in a cascade implementation of inverse filtering and noise smoothing:



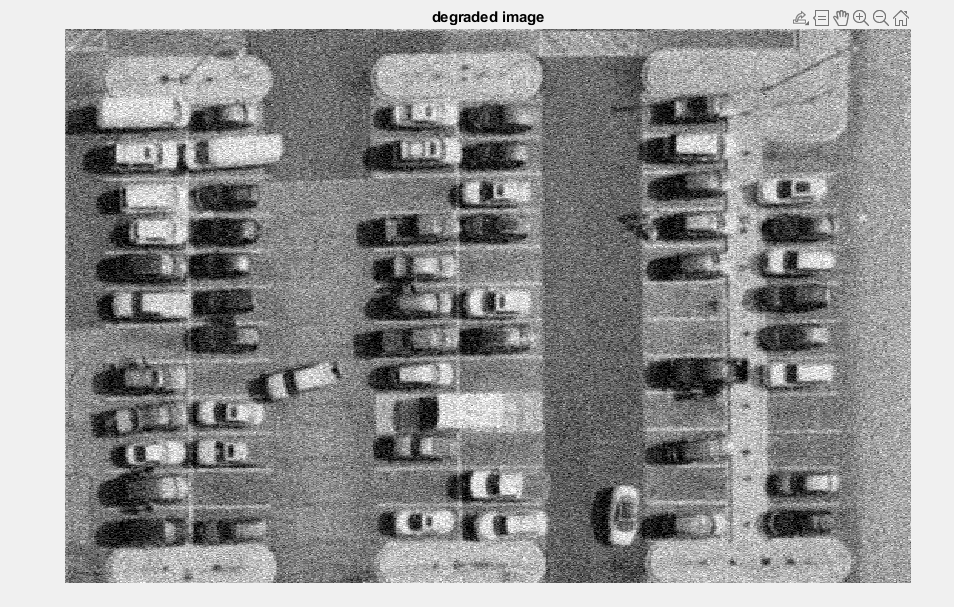
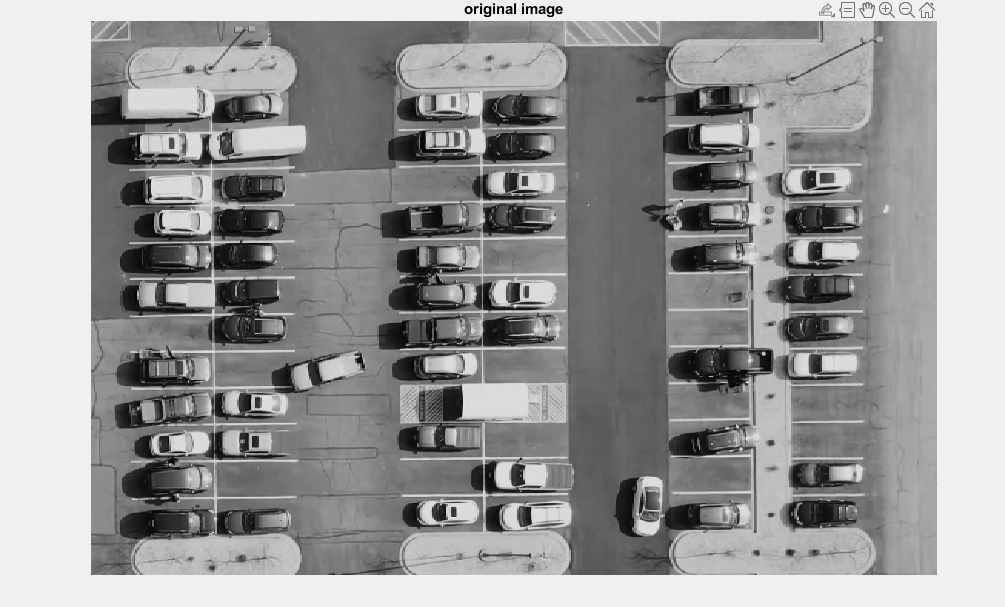
The disadvantage of this implementation is that when the inverse filter is singular, we have to use generalized inverse filtering. People also suggest the power spectrum of the original image can be estimated based on a model such as the model.

Image been used;



**Experimental Result output**

**Code is in zip file…**



**CS308 – CAC PART 2**

**Answers**

**Question 1**.

In order to ensure a smooth operation of this intelligent system, it is important to properly initialize the system. This can be done by setting up the parameters of the system, such as the size of the parking lot, the number of parking spaces, the type of parking spaces, etc. Additionally, it is also important to properly calibrate the sensors and cameras that will be used in the system.

The process of initialization is essential for any computer system; nevertheless, it is of the utmost significance for a system that is going to be used for essential activities such as parking. During initialization, the system's settings are configured, including things like the dimensions of the parking lot, the total number of parking spots, the various types of parking spots, and so on.

In addition to this, it is essential to ensure that the sensors and cameras that will be incorporated into the system have been calibrated correctly. Because of this, the system will be able to detect automobiles and parking spots with a high degree of precision.

You have the option of manually or automatically doing the initialization. During manual startup, a human operator is responsible for configuring the system. This can be accomplished by providing the system with the appropriate values for its parameters. During the automatic startup process, the system will automatically recognize the system's parameters. This is something that can be accomplished by the utilization of sensors, such as cameras, that can detect the parameters of the system.

**Question 2**

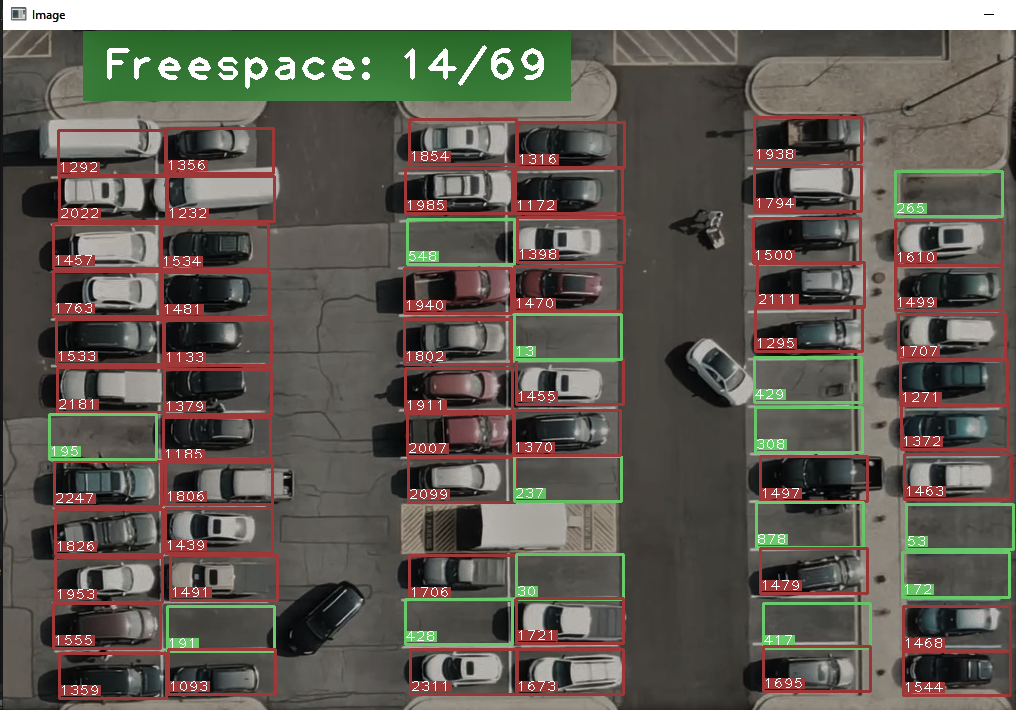
Object/video detection is key to the building of this smart model. The detection technique must consist of sub-tasks such as counting the number of cars parked, finding the relative size of the parking space, or finding the relative distance between the parked cars. Various computer vision algorithms that can be used to handle such tasks include object detection algorithms, such as Haar Cascade or YOLO, or video analysis algorithms, such as optical flow.

The process of constructing this intelligent model relies heavily on the detection of objects and video. The method of detection must include a series of subsidiary duties, such as determining the relative size of the parking spot, determining the relative distance between the automobiles that are parked, and counting the number of cars that are parked. Object detection algorithms, such as Haar Cascade or YOLO, or video analysis algorithms, such as optical flow, are two examples of the different kinds of computer vision algorithms that can be used to handle tasks of this nature.

In order to recognize moving or still things in pictures or videos, such as cars, computer programs called object/video detection methods are utilized. These algorithms can be utilized to carry out a variety of tasks, such as determining the relative size of a parking space or counting the number of cars that are parked in a particular area. There are a wide variety of object/video detection techniques, some of which include YOLO and Haar Cascade. The Haar Cascade is a well-known technique for detecting objects in photos, and it is frequently utilized for this purpose. Haar cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. This algorithm is not so complex and can run in real-time. We can train a haar-cascade detector to detect various objects like cars, bikes, buildings, fruits, etc.

YOLO is a well-known algorithm for detecting objects, and it is typically applied to videos in order to do so. YOLO or ‘You Only Look Once’ is an algorithm that provides real-time object detection using neural networks. This algorithm is known for its speed and accuracy. The algorithm can be used to detect people, animals, traffic signals, etc. YOLO uses convolution neural networks or CNNs to perform real-time object detection. The CNN model predicts the class probabilities for the detected objects and applies bounding boxes for the detected objects in an input image. And as per the name, the algorithm only requires a single forward propagation through the model for object detection and prediction in an input image.

Image output after the use of the computer vision algorithm



The total parking spacing is 69 as seen in the image above. From the image we can see that 14 free space (green) is still available. The counting shown in the image detects when a car is available or not. It was set that if car is in a parking space it should count above 900 and if a car is absent, it should count below 900. With this algorithm it is easy to know when a car or parking space is available.

Code:

#counts below 900 if car is available else do otherwise  
 if count < 900:  
 color = (0, 255, 0)  
 thickness = 2  
 spaceCounter += 1  
 else:  
 color = (0, 0, 255)  
 thickness = 2  
 #styling the rectangle for available space (green) or not (red)  
 cv2.rectangle(img, pos, (pos[0] + width, pos[1] + height), color, thickness)  
 cvzone.putTextRect(img, str(count), (x, y + height - 3), scale=1,  
 thickness=1, offset=0, colorR=color)  
 #space counting --  
cvzone.putTextRect(img, f'Freespace: {spaceCounter}/{len(posList)}', (100, 50), scale=3,  
 thickness=3, offset=20, colorR=(0,200,0))

**Question 3.**

When developing a video detection system for your model, you are necessary to highlight moving cars in the video and record the coordinates of those cars. This issue can be solved utilizing a variety of approaches, such as background subtraction or frame differencing, to name just two of the many possibilities.

The common practice of subtracting the background from a video in order to locate moving items is known as background subtraction. The backdrop is removed from the image using this method, which results in the desired effect. This will ensure that the only things visible in the frame are those that are moving. Another common method that can be utilized in the process of locating moving objects inside a video is known as frame differencing.

Frame Differences method is an algorithm to identify an object's motion. Using this algorithm, we could differentiate an object moving in the environment. Background subtraction is one of the methods suitable to further improve frame differences thus increasing its effectiveness and precision.

This method operates by contrasting the currently active frame with the frame that came before it. This will draw attention to any discrepancies that exist between the two frames, which will be the items that are moving.



From the above image we can see that the objects (cars) were detected but in this case only when the object (car) is in the packing space.

**Question 4**.

Separation of parked cars and empty spaces is an important concept in image processing to be used in this smart system. This can be done by thresholding the images using the Gray-scale image equation Gray=0.229\*R +0.587\*G+0.11\*B.

This intelligent system will make use of an essential idea in image processing known as "separation," which involves separating parked cars from empty areas. The photos can be thresholder in order to get this result by utilizing the equation for grayscale images, Gray=0.229\*R+0.587\*G+0.11\*B.

Image segmentation is a common use of the widely used image processing technique known as thresholding. The image is thresholder with the use of a threshold value in order for this method to work. The equation for a grayscale image is Gray = 0.229\*R +0.587\*G +0.11\*B; you may use this equation to determine this threshold value. With the help of this equation, the threshold value for each pixel in the image may be determined. Following the application of this threshold setting, Adaptive thresholding was used. It is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions. In OpenCV, you can perform Adaptive threshold operation on an image using the method adaptiveThreshold () of the Imgproc class.

So, we get different thresholds for different regions of the same image which gives better results for images with varying illumination.

In addition to the parameters described, the method cv.adaptiveThreshold takes an input parameters:

The adaptiveMethod decides how the threshold value is calculated:

* cv.ADAPTIVE\_THRESH\_GAUSSIAN\_C: The threshold value is a gaussian-weighted sum of the neighborhood values minus the constant C.

The block Size determines the size of the neighborhood area and C is a constant that is subtracted from the mean or weighted sum of the neighborhood pixels.

The code below was used for the threshold implementation

Gray = 0.229\*R +0.587\*G +0.11\*B = RGB values must be integers between 0 and 255.

imgThreshold = cv2.adaptiveThreshold(imgBlur, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C,  
 cv2.THRESH\_BINARY\_INV, 25, 16)

The image below shows our default image threshold (adaptive) output.



Note the thick lines in the image. The thick lines represent available cars in the park. By applying adaptive thresholding, we can threshold local regions of the image (rather than using a global value of our threshold parameter, T). Doing so dramatically improves our foreground and segmentation results.

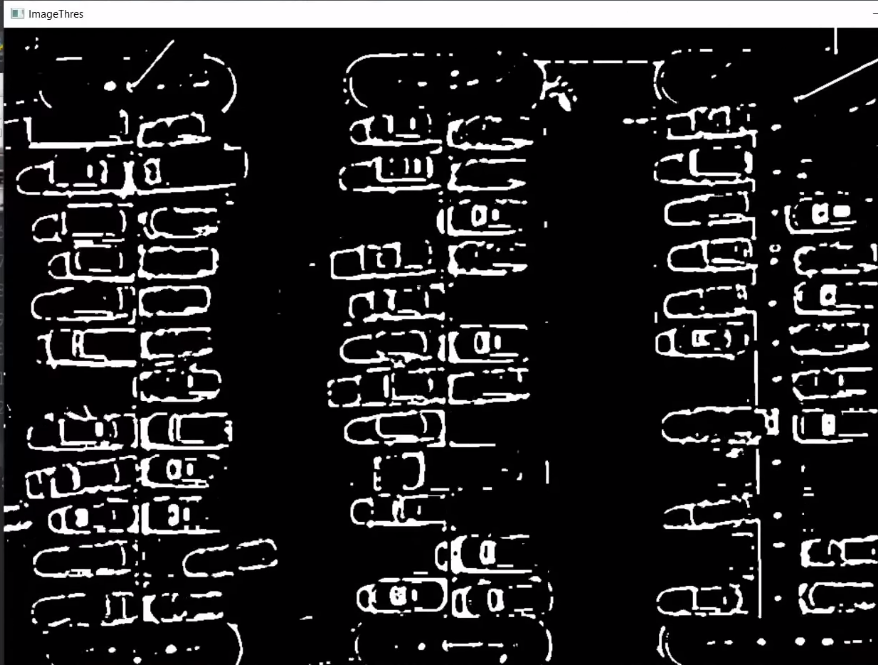
**Question 5.**

Mostly, after thresholding an image, there will be some unnecessary fragments in the image as seen in the image after adaptive threshold was implemented. Salt and pepper noise can be seen in the image, it will make sense when the noise is removed. Salt and pepper noise is composed of pepper noise and salt noise, and is generally generated due to factors such as external environment and device performance during image transmission and acquisition. The black pixels corresponding to pepper noise usually appear in the brighter areas of the image, and the white pixels corresponding to salt noise usually appear in the darker areas of the image. The two kinds of noise appear at the same time and are randomly distributed, and the image will show a noisy effect of black and white noise. Salt and pepper noise is usually not conducive to image edge extraction, restoration, and other operations. Since the pixel value of the noise point is quite different from the pixel value of its neighbors, it can usually be removed by a median filter. The standard median filter is a typical nonlinear filtering method with a simple idea and fast processing, which is most suitable for processing impulse noise such as salt and pepper noise. The image consists of thousands of pixels, and the pixels polluted by noise are often isolated from each other. To remove these isolated pixels, a simple idea is to replace the pixel value with its surrounding pixel value. The median filter is based on this idea, which establishes a window of size n\*n. It scans each pixel from left to right and from top to bottom, sorts the n2 pixels contained by the pixel value, and selects the intermediate value to replace the pixel value of the processing pixel, thereby removing impulse noise. This simple processing works well for images with low noise contamination density, but as the noise density increases, the noises are no longer isolated. To ensure the removal effect, the window size needs to be enlarged, which also leads to blurred edges and details.

The code for the noise removal using median

imgMedian = cv2.medianBlur(imgThreshold, 5)  
kernel = np.ones((3, 3), np.uint8)  
imgDilate = cv2.dilate(imgMedian, kernel, iterations=1)

the output after the removal is seen in the image below;



**Question 6**

For Output 1 display: Observation of the Whole parking area, and

For Output 2 display: Observation of each lane in the parking area and the vacant lane.

1. Open zip file
2. Extract
3. Open folder in code editor (PyCharm was used)
4. Run code (main) for output.