Project Goal

To find number of vacant spots in a parking lot.

Approaches

1) Using OpenCV to detect and draw contours in the image. This technique involves traditional computer vision algorithms and does not involve any deep learning techniques.

   1.1) Do some pre-processing on images i.e. converting it to binary, finding edges using canny edge detector or any other and thresholding the image.
   1.2) On the image obtained from above step use OpenCV "cv2.findContours" function. This will find contours in the given image which means the number of objects in the image.
   1.3) Draw boundary boxes around the contours detected from step 1.2 above using "cv2.drawContours" function of OpenCV.
   1.4) The area of boxes drawn for vehicles will be larger than other small objects in the image. So, we find area of all the contours detected in the image and threshold them to some value which we feel will be the area of car (it depends on the position and angle of camera and input image that we feed to algorithm)
   1.5) We can discard the contours below the threshold and keep rest of them. The ones we keep will be equal to number of vehicles present in the parking lot.
   1.6) Subtract the value obtained from above step from total available spots. This will give us number of spots still available.
   1.7) References
   1.7)1. https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html
   1.7)2. https://www.programcreek.com/python/example/70440/cv2.findContours
   1.7)3. https://www.amazon.com/Learning-OpenCV-Computer-Vision-Library/dp/0596516134

2) Use **YOLO deep learning architecture** to detect vehicles in an image. This approach assumes knowledge of deep
   learning and state-of-the-art CNN architectures for object detection and recognition.

   2.1) YOLO (You Only Look Once) is a state-of-the-art deep CNN architecture used to detect and locate object.
   2.2) It can identify around 90 different classes of objects which includes car, trucks, and other kind of vehicles.
   2.3) Write a python script that implements YOLO architecture.
   2.4) Run YOLO algorithm on your input image. It will detect all the objects in your image and draw bounding boxes around it.
   2.5) Out of all the objects identified, we can output the objects which have classes labelled as car, truck, motorbike, or any other vehicle class.

2.6) Discard all other bounding boxes but vehicles.

2.7) Count number of those output boxes and subtract it from available spots.

2.8) References

   a. https://pjreddie.com/darknet/yolo/
   b. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

3) Use Tensorflow Object Detection API to detect objects in image. Rest of the process is same as Approach 2
   a. References
      https://github.com/tensorflow/models/tree/master/research/object_detection