

# Department of Infrastructure Engineering

## GEOM90038 Advanced Imaging

### Lab Assignment 4: Parking occupancy detection from CCTV images

**Due for submission at 10:00 pm on Friday 27th May 2022**

#### The task

With the availability of the inexpensive cameras (and CCTV camera that are used for surveillance), computer vision methods have developed over the years for automatic object detection and classification. The state-of-the-art methods have achieved remarkable performance for object detection and classification, and include machine learning algorithms, such as convolutional neural networks. The aim of the assignment is to perform automatic parking detection from images, and to delineate the parking spaces automatically using machine learning.

You will be provided a dataset that contains images from a publicly available dataset (PKLot) including images of occupied and unoccupied parking spaces. Using this dataset, an object detector (FasterRCNN model) will be trained to detect cars in images. Subsequently, the trained model will be used to detect cars and delineate parking spaces in another dataset (Barry street) automatically. Your task will be to perform evaluation of the object detection algorithm, improve the parking space detections and subsequently plot the final parking space delineations.



Figure 1. A screenshot of the parking occupancy detection of Barry Street data.

## Resources required

You will use MATLAB 2020a for the assignment. You can download it using the link ([https://au.mathworks.com/downloads/web\\_downloads/download\\_release?release=R2020a](https://au.mathworks.com/downloads/web_downloads/download_release?release=R2020a)). Please make sure to install all the packages (except the Simulink and related packages). Or alternatively you can use myUniApps (MATLAB 2020b). An .mlx file (to be opened using MATLAB) is provided in the LMS that will download the data and other required files for the operations. You can browse the data from the link (<https://rmit.figshare.com/ndownloader/files/24753887>). Please make sure to increase the java heap memory of MATLAB as described in the .mlx file. Another document (part of a book chapter) containing the background and the description of the process (including the dataset) is available on LMS.

## The procedure

You will use the provided MATLAB live script that has two tutorials in it, and a tool for visualising the dataset. In the first tutorial you will be able to fine-tune a pre-trained network (see book chapter for details) as a car classifier that can be used to classify the occupied and empty parking slots. In the second tutorial you will be able to delineate the parking spaces automatically using spatio-temporal reasoning and using a density-based clustering algorithm. Thereafter, you need to write a MATLAB code for performing evaluation (calculating precision and recall), and to improve the delineation accuracy by using the statistics of the detections and assumptions of parking spaces. Following are the steps:

**1. Visualise dataset:** After downloading the .mlx live script, run the first section to download the dataset and unzip. The second section will load the bounding boxes of PKLot dataset and will annotate a few occupied parking spaces and will show an example occupied and empty parking slot. In the next section of the code, you will be able to visualise Barry street dataset and the parking spaces.

**2. Create a car classifier:** This step corresponds to Tutorial 1 of the live script where a pre-trained classifier is fine-tuned with PKLot dataset containing images and labels. The input to the model is an image and the output is a decision, whether the parking slot is occupied or empty. Note, in its default, the provided code does not fine-tune the model, instead it loads the fine-tuned model for the experiments. To run it on-line one can set the option “train” to “true”. You can use the trained car classifier to test with Barry street dataset, and subsequently can estimate the accuracy, and some of the wrong estimations.

**3. Delineate the parking spaces automatically and visualise:** This step corresponds to Tutorial 2 of the live script where you will perform automatic parking slot delineation. A pre-trained object detector is fine-tuned with PKLot containing images and bounding boxes. The input to the model is an image of the whole parking space, and the output is the bounding boxes of the cars on the image. Similar to the previous step, you can set the “train” option to “true” to fine-tune the model on-line, note, however, this needs a GPU for training, otherwise it might take a very long time. You can use this trained car detector to detect cars for Barry Street images.

However, you might realise, not all cars are detected. Also, not all parking spaces are occupied for the car detector to delineate them into parking spaces. Therefore, you can run the trained car detector on all the images of Barry Street dataset, and create a cluster around each parking space. Later, you can use a density-based cluster algorithm (dbScan) to find the location of the parking spaces.

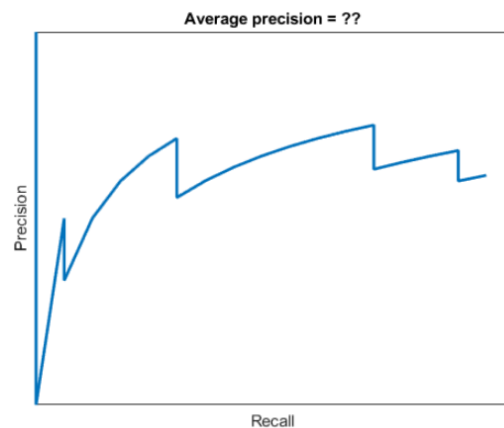
**Important note:** For the next steps, the code for performing the step **must** be accompanied, otherwise this will lead to a penalisation.

#### 4. Perform evaluation

To perform the evaluation, you should write a MATLAB code to plot the precision vs recall of detections. You might realise that the value of precision is very poor, and in the next task you will improve it by using the statistics of the detections, and assumptions related to parking of a vehicle.

**The pseudocode:**

- i. Adjust the variable “classifiedMean” to take into account of the shift of 141 pixels along X axis, and 58 pixels on Y axis.
- ii. Create a blank table in the format that is accepted by the function “evaluateDetectionPrecision” and copy the box and scores of the transformed classifiedMean variable to the new table.
- iii. Calculate average precision, recall and precision, and subsequently plot the precision-recall curve as shown above.
- iv. The goundtruth bounding boxes can be found in the file “GroundTruthBarryStreet.mat”

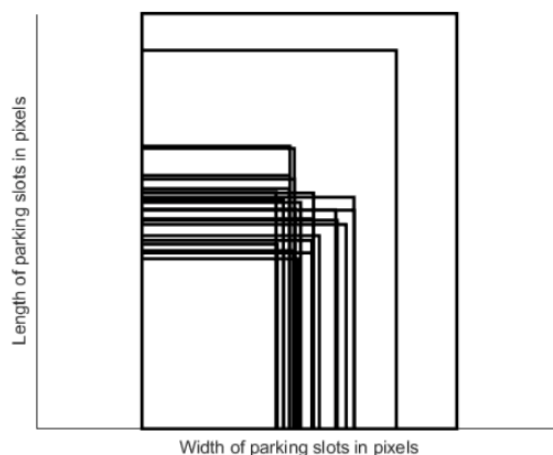


#### 5. Improve the delineation accuracy

Write a MATLAB code and perform the following steps:

##### 5.1 Plot the statistics of the bounding boxes:

Plot the bounding boxes for visualisation. You can rotate the bounding boxes to create an aspect ratio  $>1$ , where the aspect ratio is defined by length/ width of the bounding boxes. For the rotated bounding boxes, the X and Y coordinates, and the length and width of the bounding boxes should be interchanged. You can use the “rectangle” function of MATLAB for the plot.



**5.2 Calculate the average length and width of the parking slots:** Subsequently, calculate the mean length and width of the parking spaces for the whole parking area.

**5.3 Post process the length of the parking slots based on assumptions:** Assume that:

- The length and width of the parking slots remain constant throughout the image. Therefore, use the average length and width for all the parking spaces
- The cars take 80% of the length of the parking space, and therefore, increase the length of the detections by 25% of the original length.

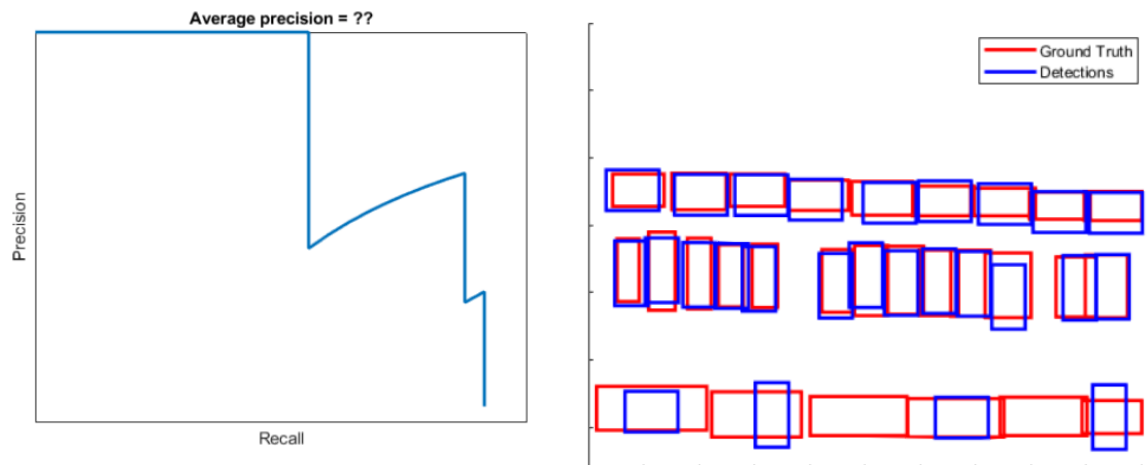
**The pseudocode:**

1. Adjust the variable "classifiedMean" to take into account of the shift of 141 pixels along X axis, and 58 pixels on Y axis.
2. For each bounding box in the transformed "classifiedMean" variable do:
  - a. Check if aspect ratio is greater than 1, then do
    - i. Calculate "DeltaX" as the difference between the bounding box length and the calculated average length of the car times a factor of 1.25
    - ii. Calculate "DeltaY" as the difference between the bounding box width and the calculated average width
    - iii. To calculate the new X coordinate of the bounding box DeltaX/2 should be added
    - iv. To calculate the new Y coordinate of the bounding box DeltaY/2 should be added
    - v. Allocate the average updated length (1.25\*average length) and average width for the new bounding boxes

Else do:

- i. Calculate DeltaX as the difference between the bounding box width and the calculated average width
- ii. Calculate DeltaY as the difference between the bounding box length and the calculated average length of the car times a factor of 1.25
- iii. To calculate the new X coordinate of the bounding box DeltaX/2 should be added
- iv. To calculate the new Y coordinate of the bounding box DeltaY/2 should be added
- v. Allocate the average updated length (1.25\*average length) and average width for the new bounding boxes

**5.4 Recalculate the precision vs recall and show final bounding boxes:** Subsequent to post-processing, plot the precision and recall values again to show the improvement. Follow steps ii-iv of Task 4 to plot the precision-recall curve.



Use function “rectangle” to visualise the new bounding boxes, and you might need to reverse the Y axis of MATLAB in order to generate the above plot.

## Submission

Write a scientific report and include the following content:

1. Introduction: Covering the motivation of the assignment, the history of machine learning, computer vision and object detectors, and the evaluation matrices used for accuracy assessment.
2. Methods and results: Describe the processes you performed for completing each task, including visualise dataset, create a car detector, delineate the parking spaces automatically, perform evaluation and improve the accuracy performance. Include the screenshots of each process and any intermediate results.
3. Discussion: Describe about the accuracy evaluation of the parking slots, and show the improvement based on the made assumptions. Also, describe the challenges and the shortcomings of the performed method and propose scopes of improvements.
4. Conclusions: Provide a summary of your findings, highlighting the accuracy gain achieved before and after the pre-processing, and how it can be further improved.
5. Include the code snippets used for the calculations.

Submit a digital version of your report, via LMS and in pdf format only.

## Marking rubric

Introduction and object detection review	10%
Proper description of the method	10%
<ul style="list-style-type: none"><li>• Visualise dataset</li><li>• Create a car detector</li><li>• Delineate the parking spaces automatically</li></ul>	
Results	
<ul style="list-style-type: none"><li>• Precision vs Recall</li></ul>	10%
<ul style="list-style-type: none"><li>• Plot the statistics of the bounding boxes</li></ul>	10%
<ul style="list-style-type: none"><li>• Calculate the average length and width of the parking slots</li></ul>	10%
<ul style="list-style-type: none"><li>• Post process the length of the parking slots on the 80% assumption</li></ul>	20%
<ul style="list-style-type: none"><li>• Recalculate the precision vs recall and visualise bounding boxes</li></ul>	10%
Discussion	10%
Conclusions	10%