# Grocery Store Object Detection

In this assignment you will be working on a dataset containing images of grocery products. Your goal is to create an algorithm in Python that is able to detect and classify the grocery products within the images. This means that your algorithm should output bounding boxes and classes for products it can find in the given images. Note that you are not allowed to use external data to solve this assignment.

****

Data info:

* **training\_info.json**

This file provides you with the ground truth annotations of the bounding boxes and classes for many images. You’ll see that the json has 3 main keys.

* + - * ‘images’ provides you with info on the images such as filename, width, height,…
      * ‘annotations’ contains the annotations. You’ll notice that more info is given than you’ll need as this data set can also be used for dense segmentation. You’ll have to focus on *bbox* and *category\_id*.
      * ‘categories’: provides you with more info on the categories you will have to detect in the images.
* **testing.json**

Contains a list of images. For these images you will have to predict bounding boxes and matching classes.

* **images**

You are given 31000 images. Only for a subset of these images annotations are provided (see training\_info.json). Additionally, you only have to provide bounding boxes for a subset of the images (see testing.json). You will notice that you have a lot of images that are not listed in the two json files. It is up to you to use the remaining images or not.

## Deliverables

### CSV file with your predictions of the test set:

Submit a csv containing the follow info:

**image\_name,id,confidence,x\_center\_scaled,y\_center\_scaled,bbox\_width,bbox\_height**

D2S\_000800,19,0.937525,0.350391,0.704687,0.322461,0.41380200000000006

D2S\_000800,19,0.928084,0.769141,0.536458,0.244531,0.45078100000000004

D2S\_000800,19,0.9075049999999999,0.48007799999999995,0.297135,0.185156,0.41849

Each row specifies a bounding box your algorithm has detected.

* **Image**\_**name**: the name of the image in which you detected the bounding box, note that no suffix is present.
* **Id**: this is the class id.
* **Confidence**: a number between 0 and 1 indicating the confidence your algorithm assigns to the detected bounding box & class.
* **x\_center\_scaled**: the x coordinate of the center of the bounding box, normalized by the width of the image.
* **y\_center\_scaled**: the y coordinate of the center of the bounding box, normalized by the height of the image.
* **bbox\_width:** the width of the bounding box, normalized by the width of the image.
* **bbox\_height:** the height of the bounding box, normalized by the height of the image.

### Code:

* + Provide us with your code how you solved this assignment.
  + Provide us with info on your python environment so that we can reproduce your results if needed.

### Presentation:

Prepare a 15 min presentation where you talk about how you solved this assignment.