

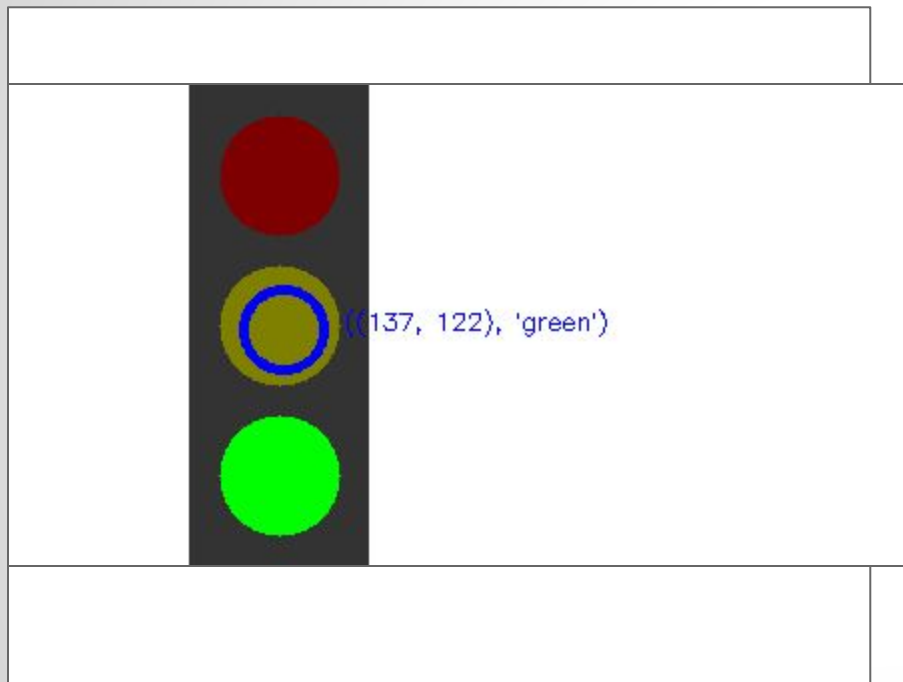
Computer Vision

Spring 2018

Problem Set #2

Darragh Hanley
darragh.hanley@gmail.com

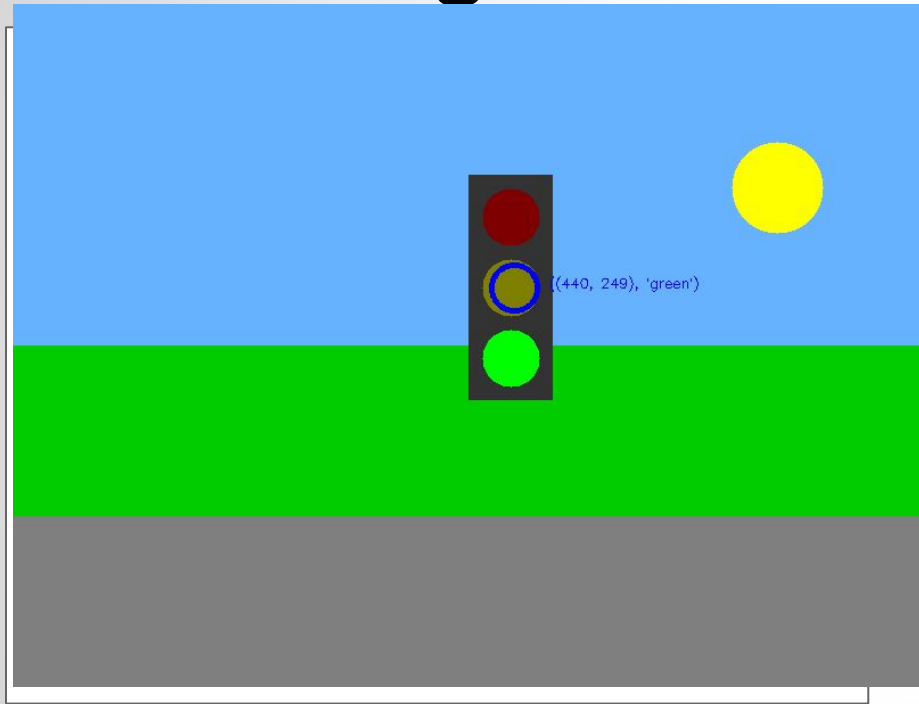
Traffic Light Detection



(137,122), green

ps2-1-a-1.png

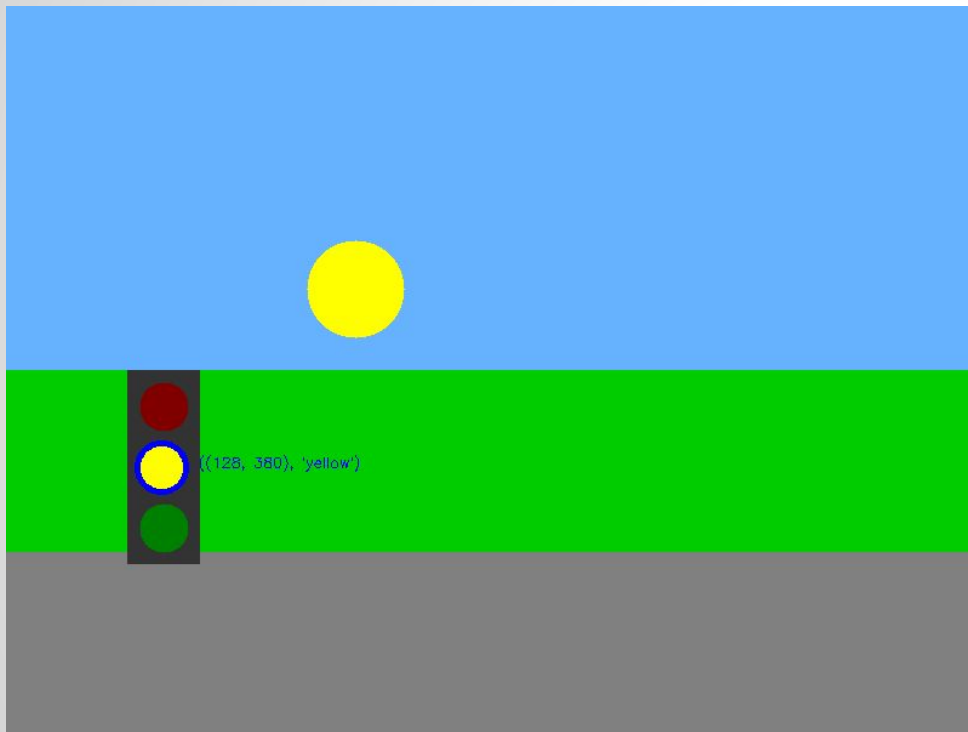
Traffic Light Detection



(440, 249) green

ps2-1-a-2.png

Traffic Light Detection

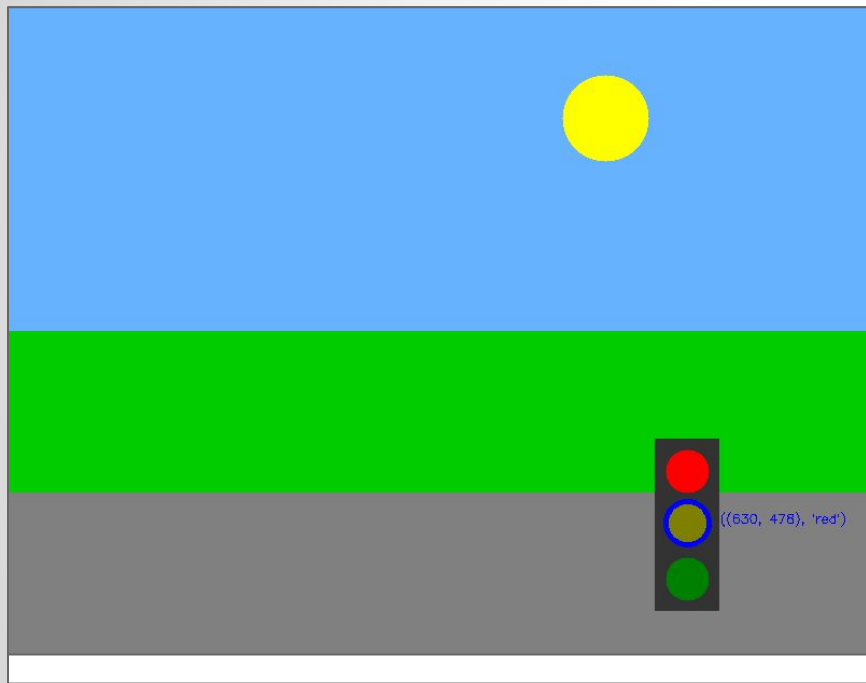


(128, 380), yellow

(128, 380), yellow

ps2-1-a-3.png

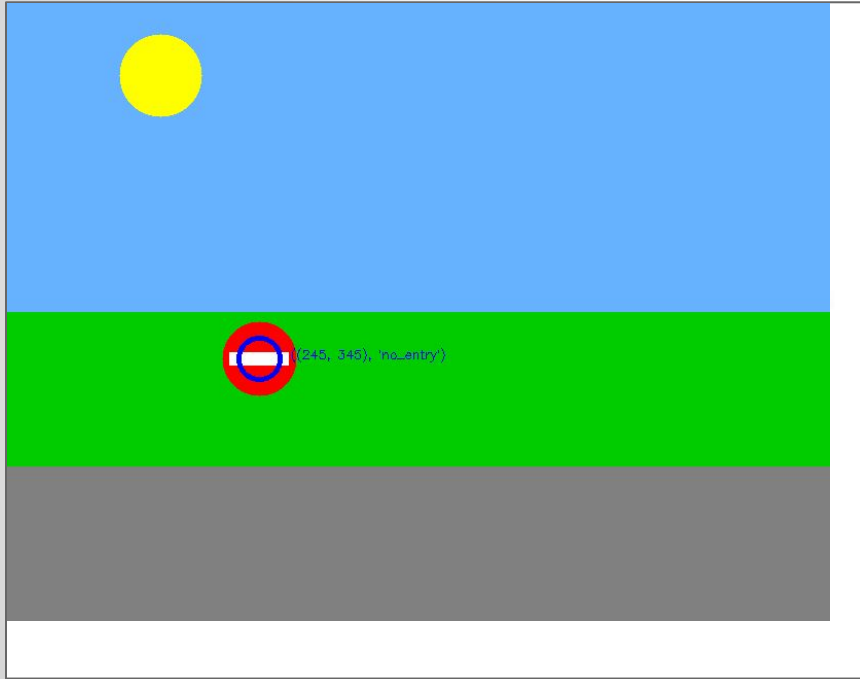
Traffic Light Detection



(630, 478) red

ps2-1-a-4.png

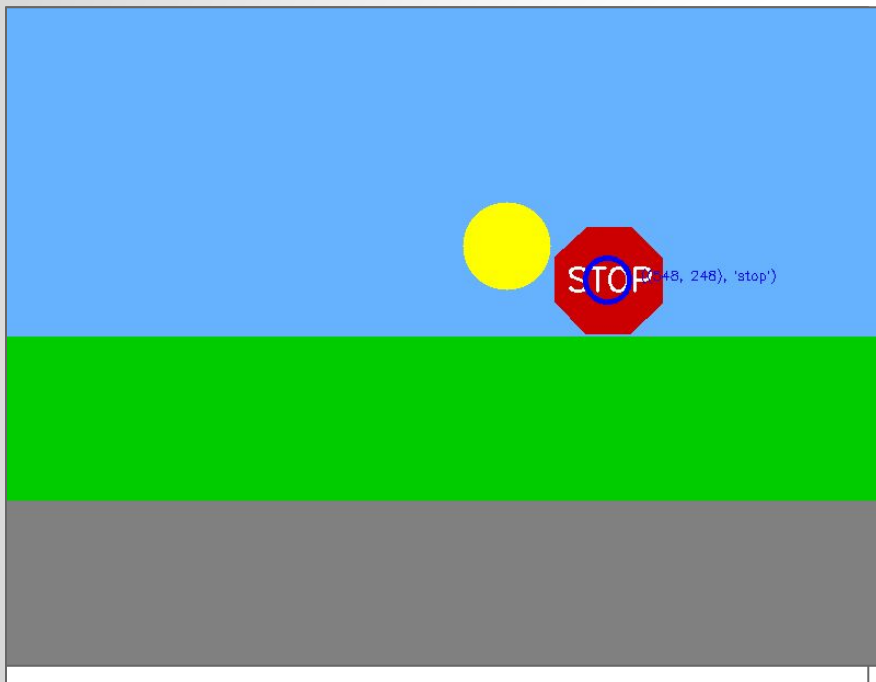
Traffic Sign Detection - Do not enter



(245, 345) no_entry

ps2-2-a-1.png

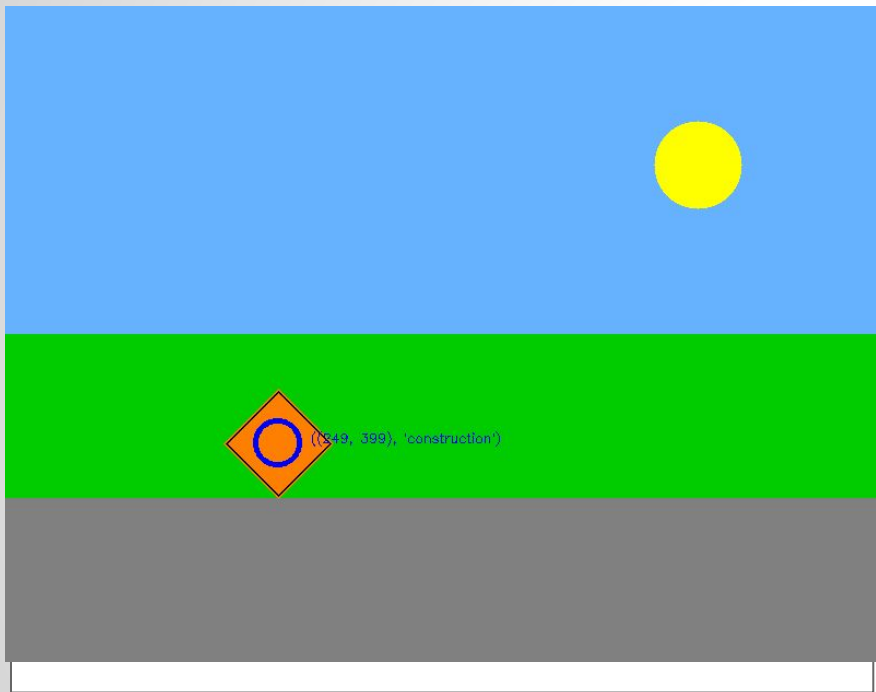
Traffic Sign Detection - Stop



(548,248) stop

ps2-2-a-2.png

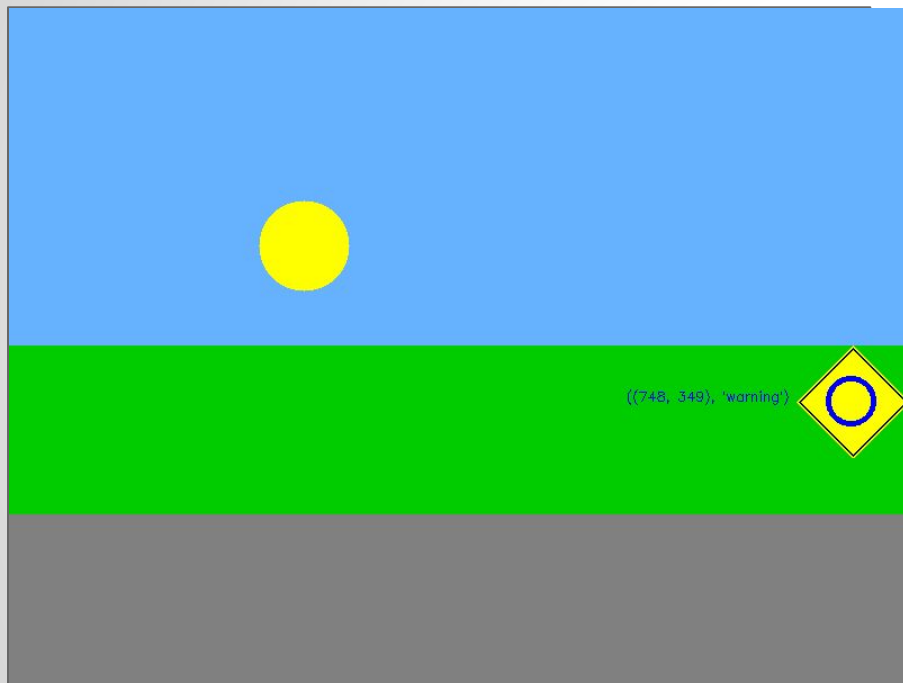
Traffic Sign Detection - Construction



(249, 399) construction

ps2-2-a-3.png

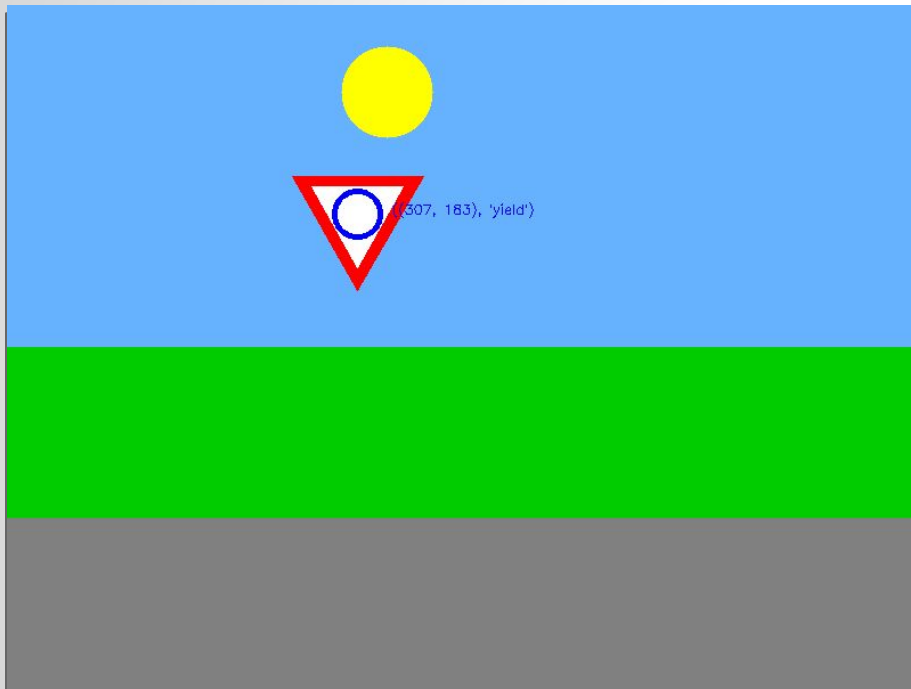
Traffic Sign Detection - Warning



(748,349) warning

ps2-2-a-4.png

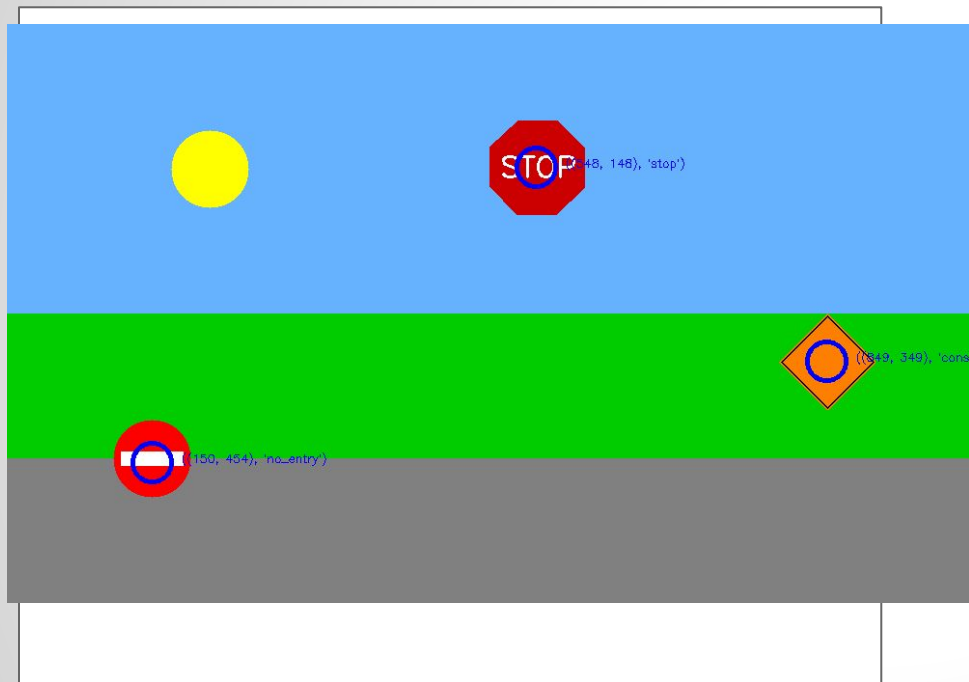
Traffic Sign Detection - Yield



(307, 183), yield

ps2-2-a-5.png

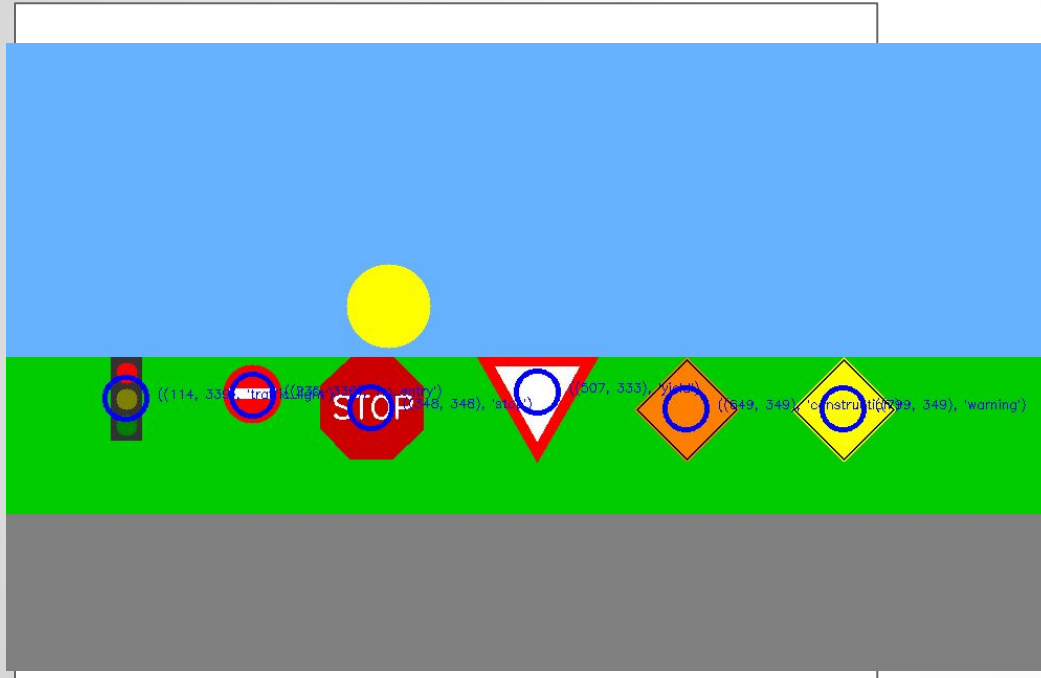
Multiple sign detection



(548, 148), stop
(150, 454), no_entry
(849, 349), construction

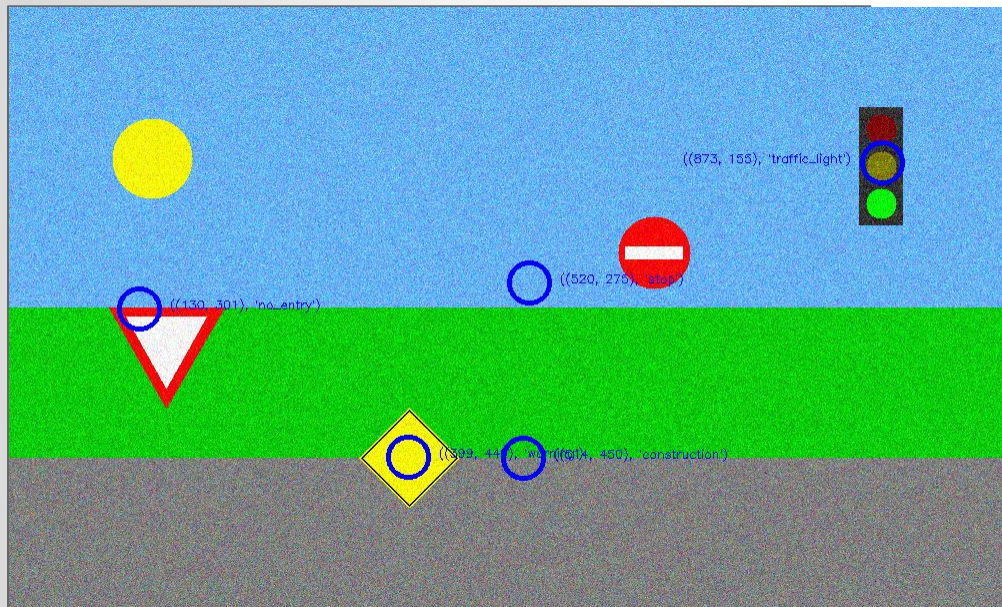
ps2-3-a-1.png

Multiple sign detection



(649, 349) construction
(799, 349) warning
(507, 333) yield
(348, 348) stop
(235, 336) no_entry
(114, 339) traffic light

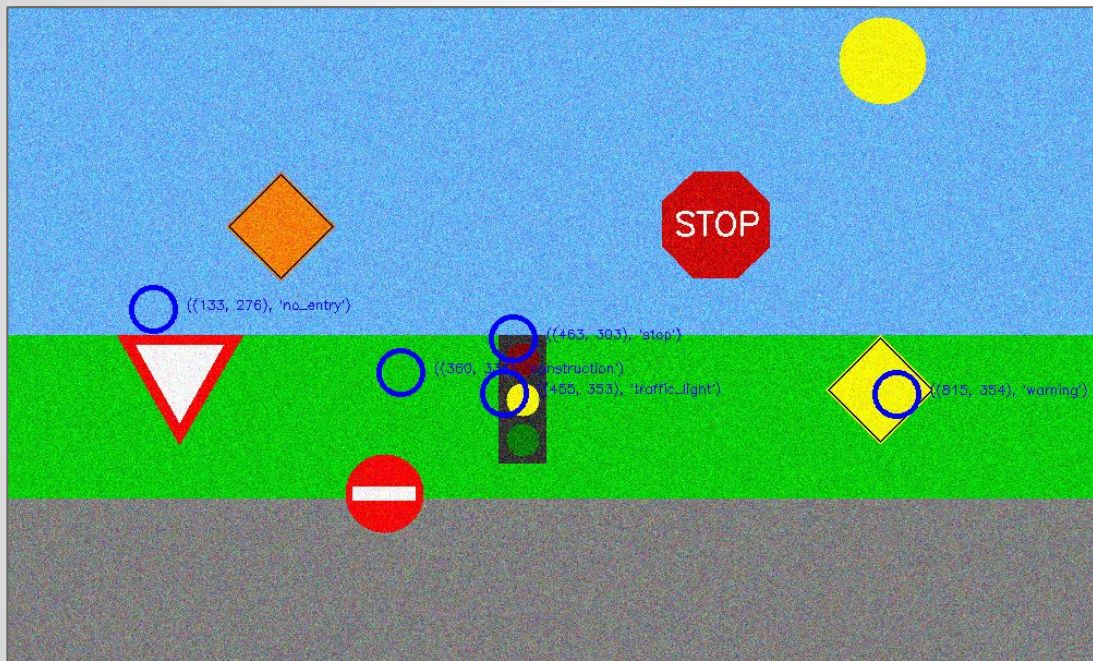
Multiple sign detection with noise



(399, 440) warning
(873, 155) traffic light

ps2-4-a-1.png

Multiple sign detection with noise



(815, 354) warning
(455, 353) traffic light

Challenge Problem



Input image



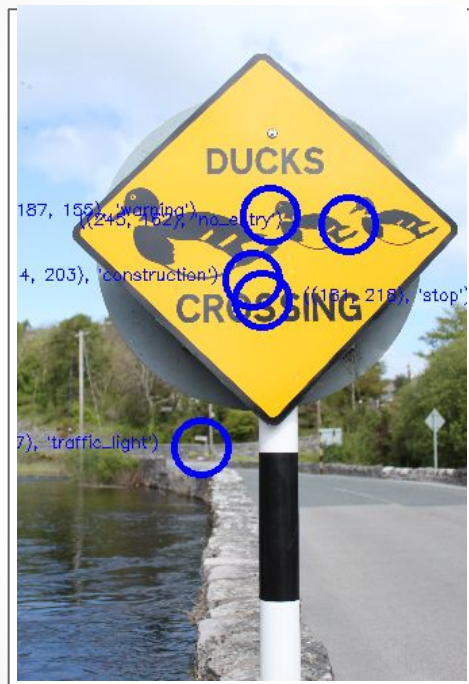
Output image

$(153, 246)$ yield

Challenge Problem



Input image



Output image

(187, 155) warning

Challenge Problem



Input image



Output image

(371, 125)
construction

Challenge Problem

Describe what you had to do to adapt your code for this task. How does the difference between simulated and real-world images affect your method?

I needed to downsize the images and keep the aspect ratio.

It is difficult in a real world scenario as the lines are often not straight and the colors not so consistent as that of the simulated images.

Challenge Problem

If you used other functions/methods, explain why that was better(or why your previous implementation did not work)

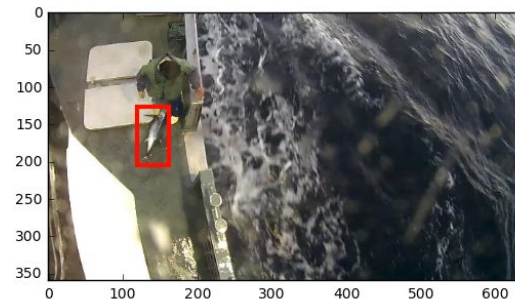
I liked the dilation and the denoising of noisy images. These were effective in improving precision.

I did not try other images within the cv2 range; however, I have run object detection with deep learning and given sufficient data found it to be a lot more practical. For example see an image in here using yolo v2 from a kaggle competition I did to identify fish:

<https://github.com/darraghdog/fish/blob/master/scripts/test%20bbox%20and%20crop.ipynb>

Nevertheless; without sufficient training data opencv could be the better choice.

```
In [9]: show_bb(3)
```



```
n [10]: plot(val0[3])
```

