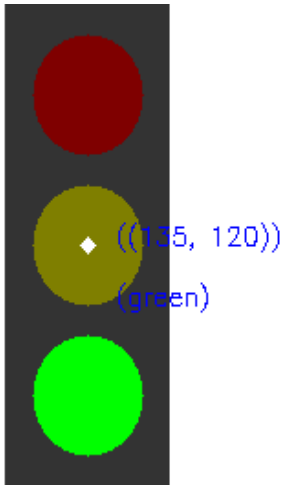
Computer Vision (SPRING 2019) Problem Set #2

Jijun HU
jijun.hu.0930@gatech.edu

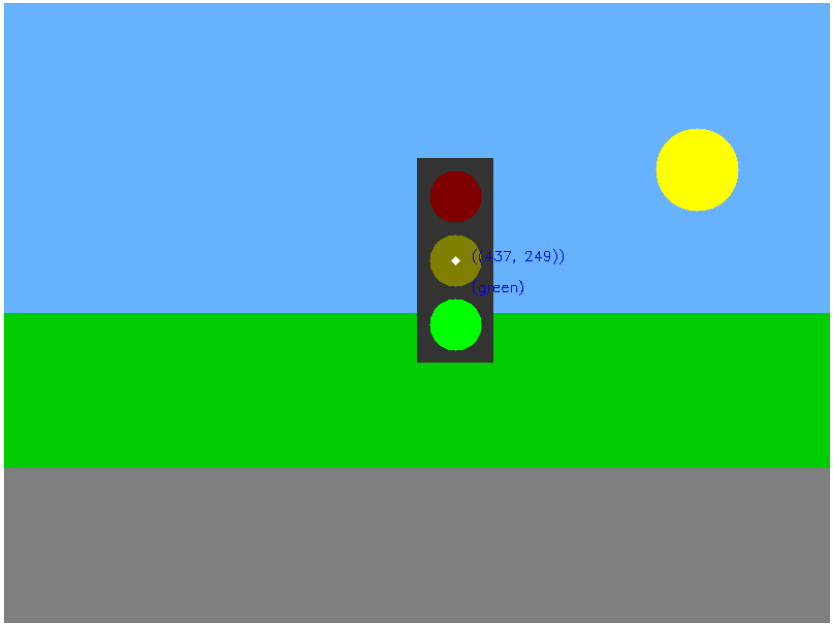
Traffic Light Detection



Coordinates and State:
Coordinates - (135,120)
State - green

ps2-1-a-1.png

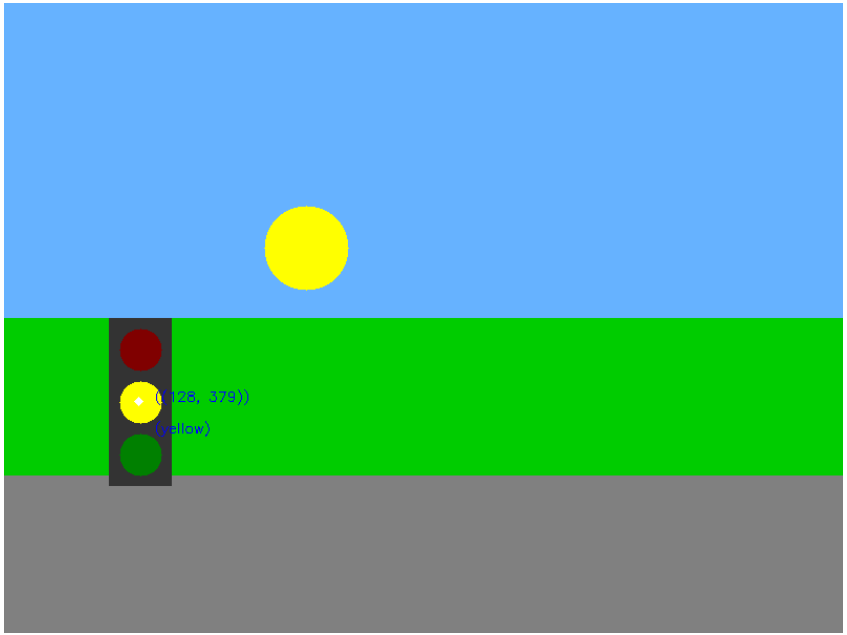
Traffic Light Detection



ps2-1-a-2.png

Coordinates and State:
Coordinates – (437, 249)
State - green

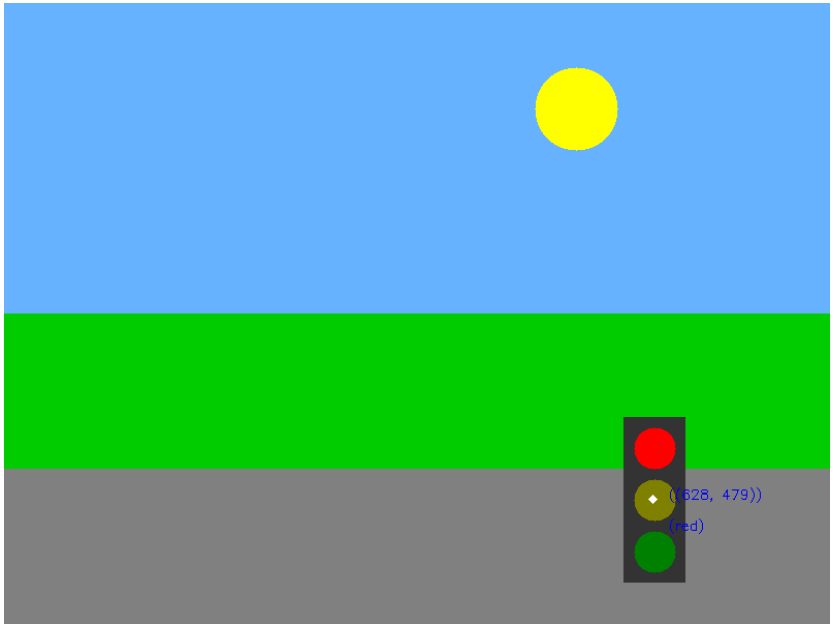
Traffic Light Detection



Coordinates and State:
Coordinates – (128, 379)
State - yellow

ps2-1-a-3.png

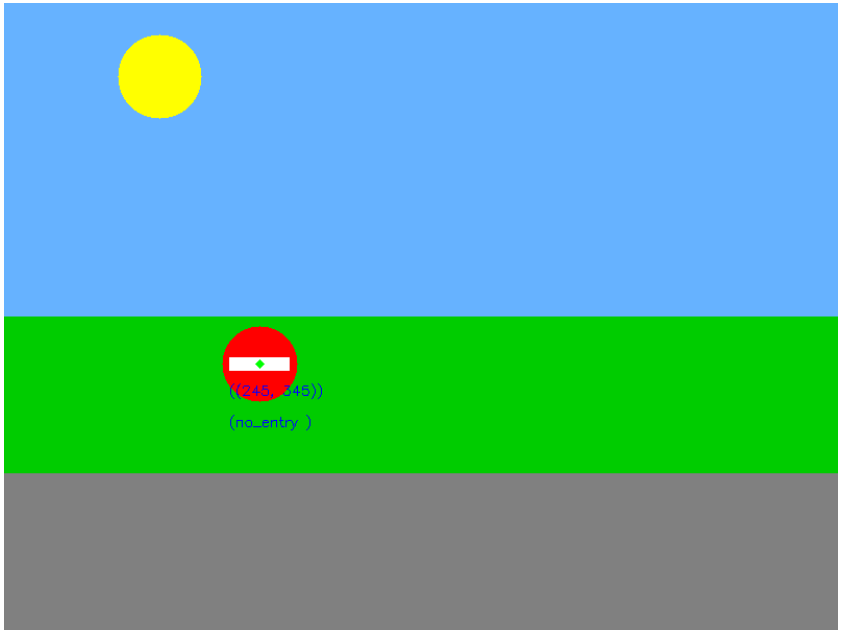
Traffic Light Detection



Coordinates and State:
Coordinates – (628, 479)
State - red

ps2-1-a-4.png

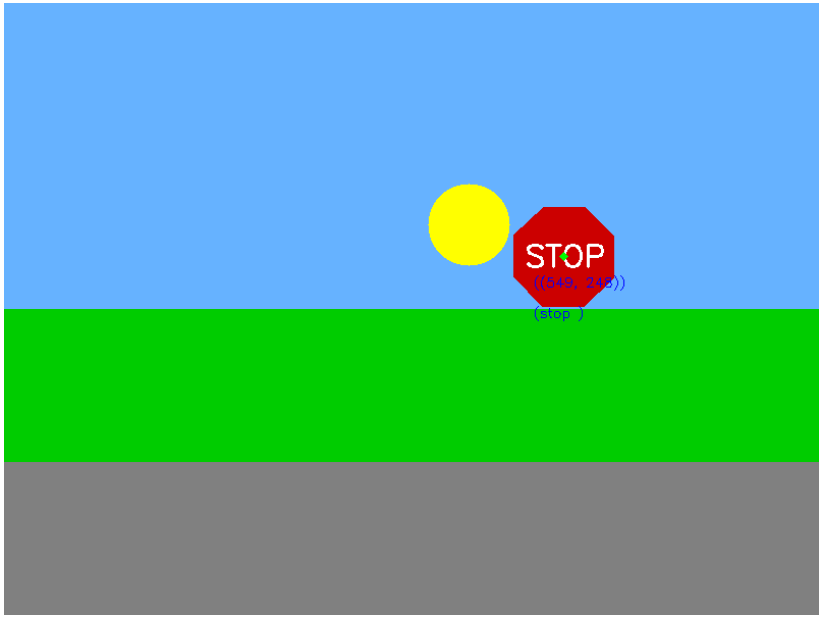
Traffic Sign Detection - Do Not Enter



Coordinates:
Coordinates – (245, 345)

ps2-2-a-1.png

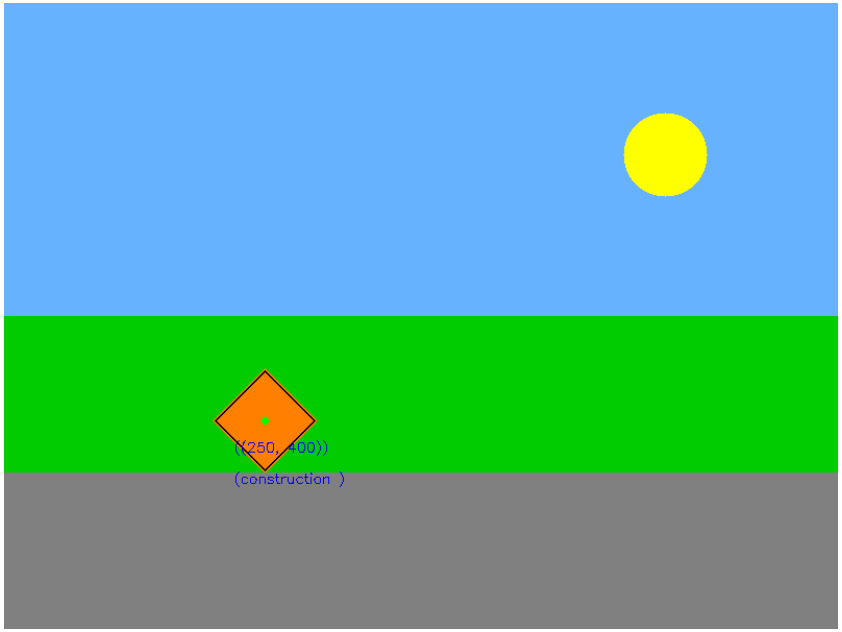
Traffic Sign Detection - Stop



Coordinates:
Coordinates – (549, 248)

ps2-2-a-2.png

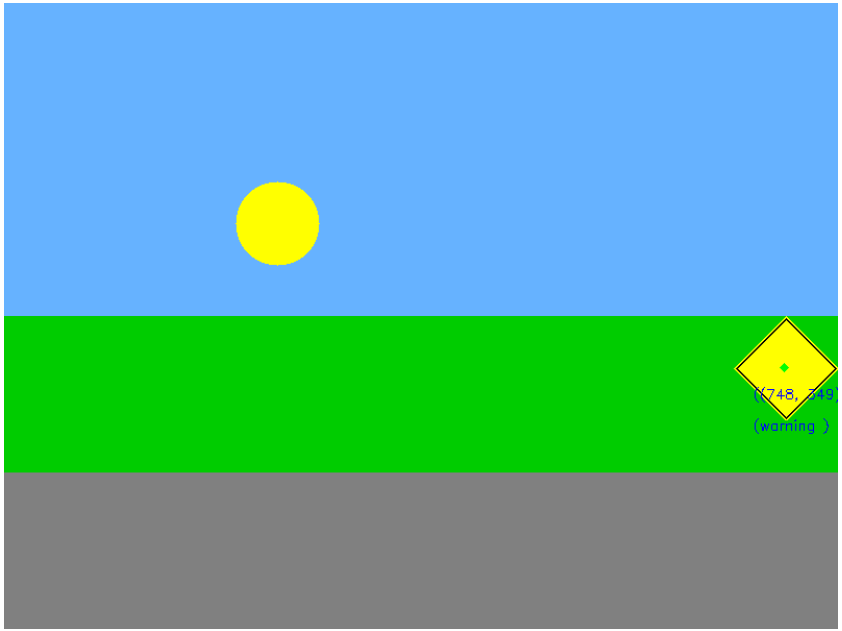
Traffic Sign Detection - Construction



Coordinates:
Coordinates – (250, 400)

ps2-2-a-3.png

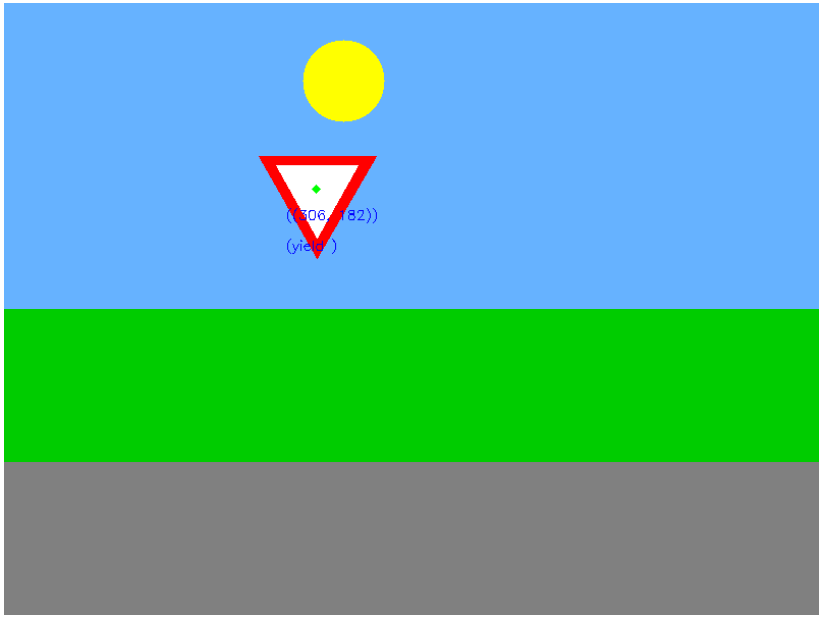
Traffic Sign Detection - Warning



Coordinates:
Coordinates – (748, 349)

ps2-2-a-4.png

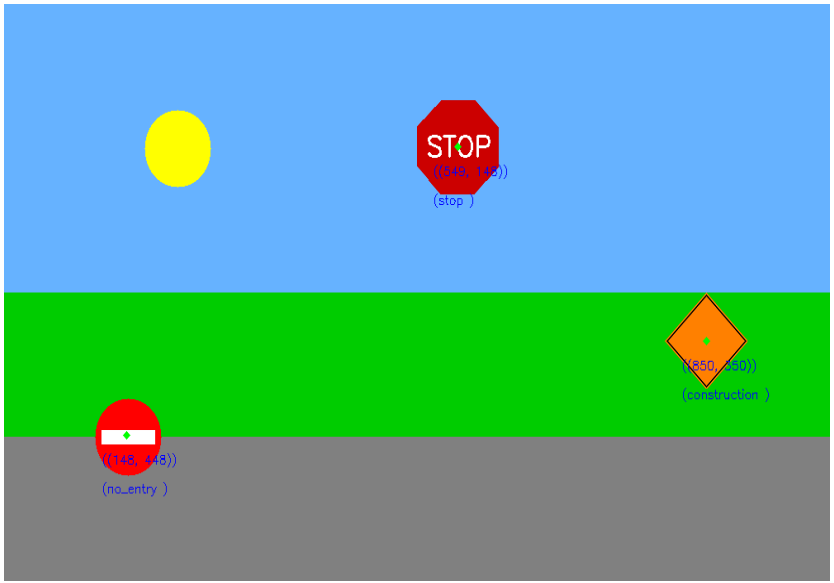
Traffic Sign Detection - Yield



Coordinates:
Coordinates – (306, 182)

ps2-2-a-5.png

Multiple sign detection

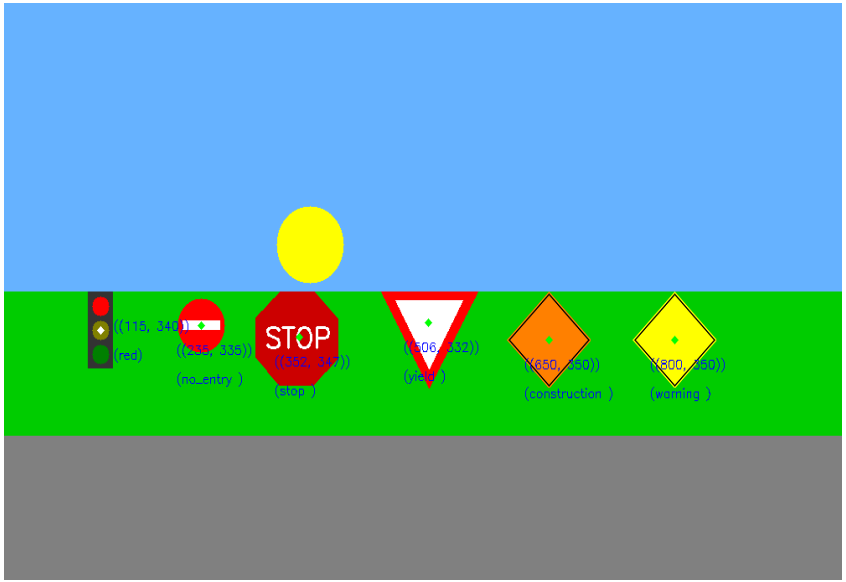


ps2-3-a-1.png

Coordinates and Names

no entry	– (148, 448)
Stop	– (549, 148)
construction	– (850, 350)

Multiple sign detection

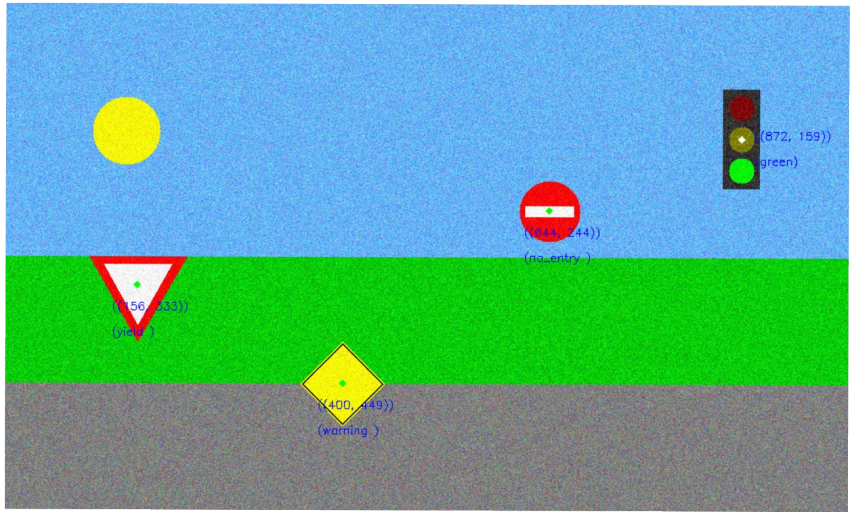


ps2-3-a-2.png

Coordinates and Names

Traffic light	– (115, 340), ‘red’
No entry	– (235, 335)
Stop	– (352, 347)
Yield	– (506, 332)
Construction	– (650, 350)
Warning	– (800, 350)

Multiple sign detection with noise

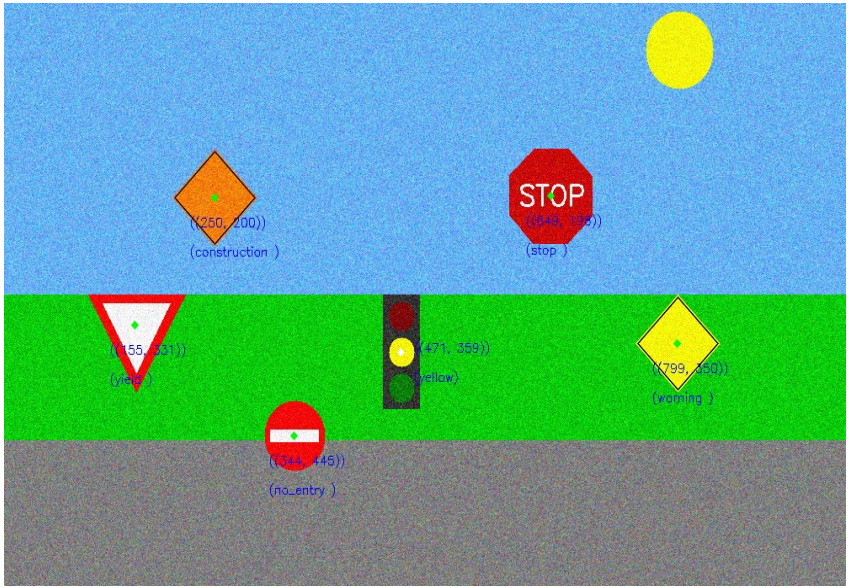


Coordinates and Names

Traffic light	– (872, 159), ‘green’
No entry	– (644, 244)
Yield	– (156, 333)
Warning	– (400, 449)

ps2-4-a-1.png

Multiple sign detection with noise



ps2-4-a-2.png

Coordinates and Names

Traffic light	– (471, 359), ‘yellow’
No entry	– (344, 445)
Stop	– (649, 198)
Yield	– (155, 331)
Construction	– (250, 200)
Warning	– (799, 350)

Challenge problem - A

Do not Enter:
(114, 108)

ps2-5-a-1

Challenge problem - A

Yield

(116, 71)

ps2-5-a-2

Challenge problem - A

Coordinates and Name:

ps2-5-a-3

Challenge problem - B

Coordinates and Name:

ps2-5-b-1

Challenge problem - B

Coordinates and Name:

ps2-5-b-2

Challenge problem - B

Coordinates and Name:

ps2-5-b-3

Challenge problem - Text

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

I feel the real image do contain much more noise in the image, and because when taking the photo, we always have different angles, that makes my conditions in my initial implementation too strong, I tried to loose to restriction a lot, but still cannot make it work for all the signs in different situation.