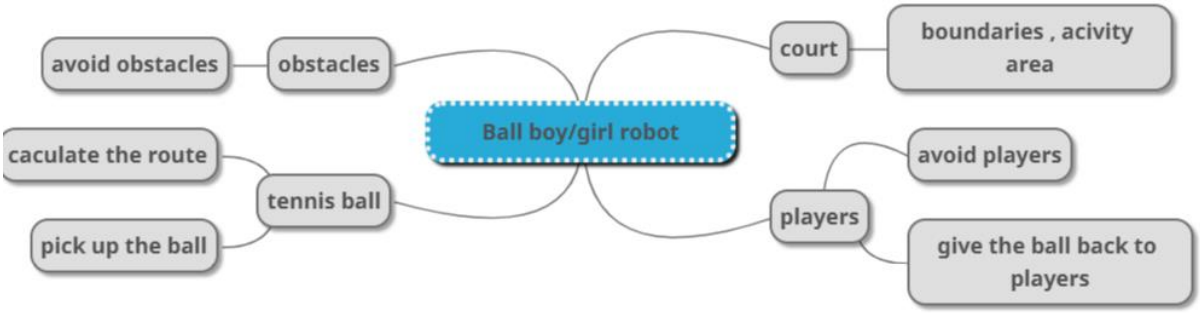
CSC420

Assignment4

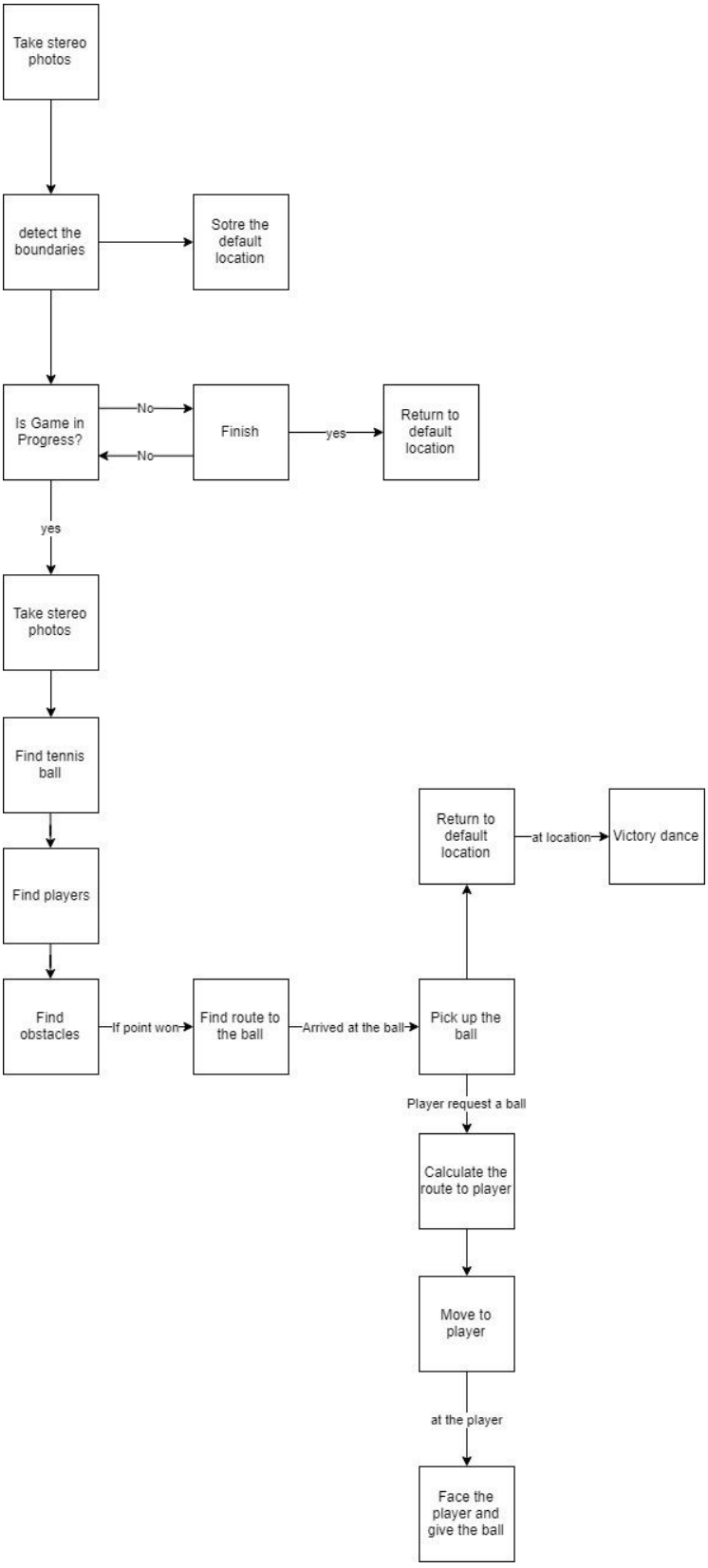
Zhili Xu

Question1:

a) Brainstorm chart



b) robot process pipeline:



c) pseudo-code for ball girl/boy robot

```
def func(match):
    boundaries = cannyEdgeDetection(tennis court)
    data = trainCNN('tennis ball', 'players', 'obstacles')
    game = match.status()
    while ingame:
        photo = StereoPhoto()
        ball = data.findBalls()
        players = data.findPlayers()
        obstacles = data.findObstacles()

        if game.afterPlayerWonAPoint:
            while not robot.atTheBall():
                ballLocationRoute = findRoute(ball, players, obstacles)
                robot.moveToLocation(ballLocationRoute)

            if robot.atTheBall():
                robot.grabObject(ball)

            if robot.hasBall() and game.playerRequestABall():
                playerLocationRoute = findRoute(player, players, obstacles)
                robot.moveToLocation(playerLocationRoute)
                robot.faceDirection(player)

        robot.moveToLocation(defaultLocation)
        robot.victoryDanceAtLocation(defaultLocation)
    # game finished
    while court.hasBall():
        ballLocationRoute = findRoute(ball, players, obstacles)
        robot.moveToLocation(ballLocationRoute)
        if robot.atTheBall():
            robot.grabObject(ball)
    robot.moveToLocation(defaultLocation)
```

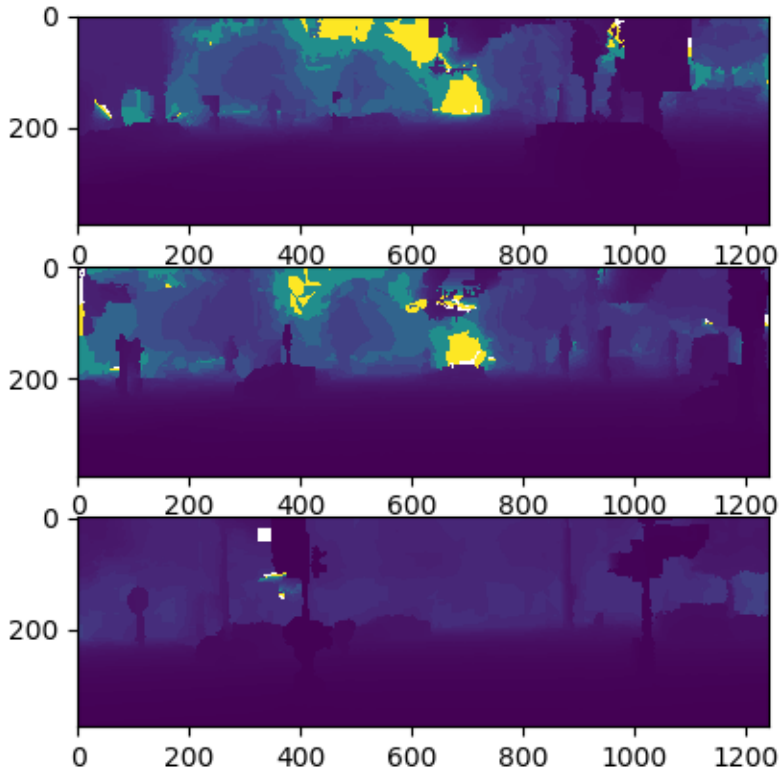
Question2:

In this question, I am using imageAI interface (link: https://github.com/OlafenwaMoses/ImageAI) and RetinaNet object detection model.

a) Depth:

```
import cv2

import numpy as np
import matplotlib.pyplot as plt

im1 = '004945'
im2 = '004964'
im3 = '005002'
results = 'data/test/results/'
left_disparity = '_left_disparity.png'
calib = 'data/test/calib'
allcalib = '_allcalib.txt'


im1_left = results + im1 + left_disparity
im2_left = results + im2 + left_disparity
im3_left = results + im3 + left_disparity

im1_calib = calib + im1 + allcalib
im2_calib = calib + im2 + allcalib
im3_calib = calib + im3 + allcalib


im1_dis = cv2.imread(im1_left, 0)
cols, rows= im1_dis.shape
im2_dis = cv2.imread(im2_left, 0)
im3_dis = cv2.imread(im3_left, 0)

im_dis = [im1_dis/256, im2_dis/256, im3_dis/256]

f = 721.537700
b = 0.5327119288 * 1000

fig = plt.figure()


for n in range(3):
```

```
    depth = np.zeros((cols, rows))
    for i in range(rows):
        for j in range(cols):
            depth[j,i] = (f-b)/im_dis[n][j,i]
            if (im_dis[n][j,i] == 0):
                depth[j,i] = 0
    fig.add_subplot(3,1, n+1)
    plt.imshow(depth)
plt.show()
```



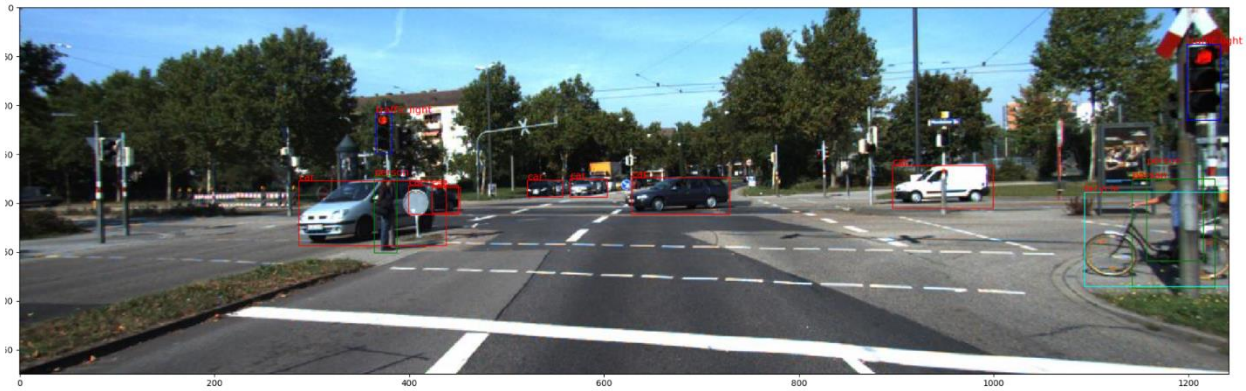
b) Object detection:

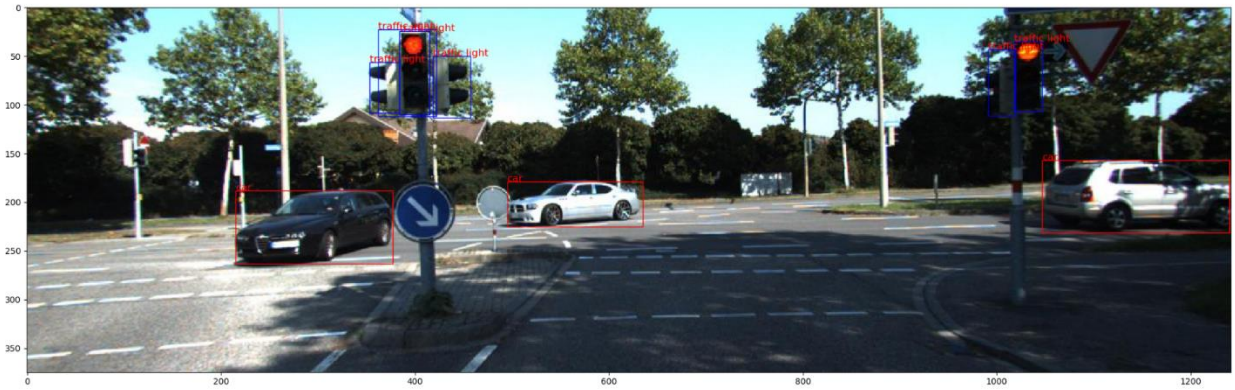
```
from imageai.Detection import ObjectDetection
import os
import numpy as np

execution_path = os.getcwd()


detector = ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath( os.path.join(execution_path ,
"resnet50_coco_best_v2.0.1.h5"))
detector.loadModel()


custom_objects = detector.CustomObjects(person=True, traffic_light=True,
bicycle=True, car=True)
detections = detector.detectCustomObjectsFromImage(custom_objects=custom_objects,
input_image=os.path.join(execution_path , "004945.jpg"),
output_image_path=os.path.join(execution_path , "left1.jpg"),
minimum_percentage_probability=30, display_percentage_probability=False,
display_object_name=False)

file = open('im1object.txt', 'w')
for eachObject in detections:
    string = eachObject["name"] + ': '
    file.write(string)
    for number in eachObject["box_points"]:
        file.write(str(number))
        file.write(' ')
    file.write("\n")
file.close()
```

Object detection data is store in ‘im1object.txt’, ‘im2object.txt’, and im3object.txt’ files separately.

c) visualization

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from PIL import Image
import numpy as np

# Create figure and axes
fig,ax = plt.subplots(1)

# Display the image

im = np.array(Image.open('004945.jpg'), dtype=np.uint8)
# im = np.array(Image.open('004964.jpg'), dtype=np.uint8)

# im = np.array(Image.open('005002.jpg'), dtype=np.uint8)
ax.imshow(im)
string = 'im1object.txt'
file = open(string, 'r')
for line in file:
    data = line.split(':')
    digit = [int(s) for s in data[1].split() if s.isdigit()]
    if data[0] == 'traffic light':
        rect = patches.Rectangle((digit[0],digit[1]),(digit[2] -
digit[0]),(digit[3]-digit[1]),linewidth=1,edgecolor='b',facecolor='none')
        ax.text(digit[0],digit[1], 'traffic light', color = 'red', fontsize=12)
    elif data[0] == 'car':
        rect = patches.Rectangle((digit[0],digit[1]),(digit[2] -
digit[0]),(digit[3]-digit[1]),linewidth=1,edgecolor='r',facecolor='none')
        ax.text(digit[0],digit[1], 'car', color = 'red', fontsize=12)
    elif data[0] == 'person':
        rect = patches.Rectangle((digit[0],digit[1]),(digit[2] -
digit[0]),(digit[3]-digit[1]),linewidth=1,edgecolor='g',facecolor='none')
        ax.text(digit[0],digit[1], 'person', color = 'red', fontsize=12)
    else:
        rect = patches.Rectangle((digit[0],digit[1]),(digit[2] -
digit[0]),(digit[3]-digit[1]),linewidth=1,edgecolor='cyan',facecolor='none')
        ax.text(digit[0],digit[1], 'bicycle', color = 'red', fontsize=12)

    ax.add_patch(rect)
plt.show()
```





I also submitted detail picture on markus (‘2c1, 2c2, 2c3’)

d) Center of mass

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math

im1 = '004945'
im2 = '004964'
im3 = '005002'
results = 'data/test/results/'
left_disparity = '_left_disparity.png'
calib = 'data/test/calib'
allcalib = '_allcalib.txt'


im1_left = results + im1 + left_disparity
im2_left = results + im2 + left_disparity
im3_left = results + im3 + left_disparity


im1_dis = cv2.imread(im1_left, 0)
im2_dis = cv2.imread(im2_left, 0)
im3_dis = cv2.imread(im3_left, 0)
cols, rows= im3_dis.shape
im_dis = [im1_dis, im2_dis, im3_dis]

f = 721.537700
b = 0.5327119288 * 1000
px = 609.559300
py = 172.854000


# # fig = plt.figure()


for n in range(3):
    depth = np.zeros((cols, rows))
    for i in range(rows):
        for j in range(cols):
            depth[j,i] = (f-b)/im_dis[n][j,i]
            if (im_dis[n][j,i] == 0):
                depth[j,i] = 0

    string = 'im'+str(n+1)+'object.txt'
    file1 = open(string, 'r')
```

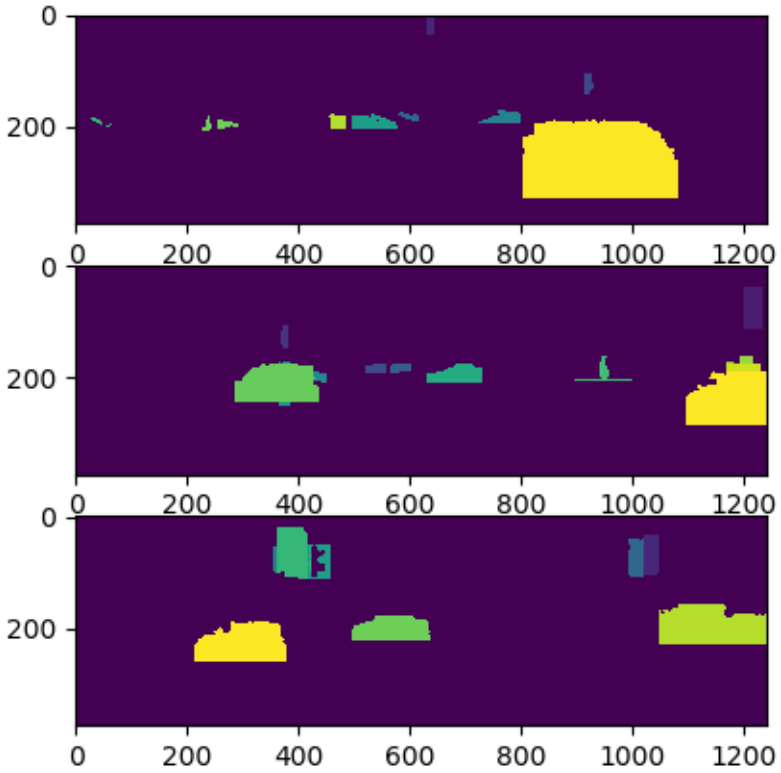
```
    data = []
    for line in file1:
        string = line.split(':')
        data.append(string)
    file1.close()

    digit = []
    for d in data:
        digit.append( [int(s) for s in d[1].split() if s.isdigit()])

    centers = []
    for obj in digit:
        xcent = round((obj[0] + obj[2])/2)
        ycent = round((obj[1] + obj[3])/2)
        z = depth[ycent, xcent]
        x = (xcent - px) * z / f
        y = (ycent - py) * z / f
        com = (x,y,z)
        centers.append(com)

    string = 'im'+str(n+1)+'com.txt'
    file2 = open(string, 'w')
    for cord in centers:
        for i in cord:
            file2.write(str(i))
            file2.write(' ')
        file2.write('\n')
    file2.close()
```

Find the image center of (x,y) coordinate, and then use depth information to find the (z) coordinate. Then use the formula we can find 3D location coordinate. The center of mass information is store in files ‘im1com.txt’, ‘im2com.txt’, ‘im3com.txt’.

e) segmentation:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math

im1 = '004945'
im2 = '004964'
im3 = '005002'
results = 'data/test/results/'
left_disparity = '_left_disparity.png'
calib = 'data/test/calib'
allcalib = '_allcalib.txt'


im1_left = results + im1 + left_disparity
im2_left = results + im2 + left_disparity
im3_left = results + im3 + left_disparity


im1_dis = cv2.imread(im1_left, 0)
im2_dis = cv2.imread(im2_left, 0)
im3_dis = cv2.imread(im3_left, 0)
im1 = 'data/test/left.004945.jpg'
im2 = 'data/test/left.004964.jpg'
im3 = 'data/test/left.005002.jpg'
im3 = cv2.imread(im3, 0)
cols, rows= im3_dis.shape
im_dis = [im1_dis, im2_dis, im3_dis]

f = 721.537700
b = 0.5327119288 * 1000
px = 609.559300
py = 172.854000

fig = plt.figure()
for n in range(3):
    depth = np.zeros((cols, rows))
    for i in range(rows):
        for j in range(cols):
            depth[j,i] = (f-b)/im_dis[n][j,i]
            if (im_dis[n][j,i] == 0):
                depth[j,i] = 0

    segmentation = np.zeros((cols, rows))
```

```
    string = 'im' + str(n+1) + 'com.txt'
    file1 = open(string, 'r')

    com = []
    for line in file1:
        com.append ([float(s) for s in line.split()])

    file1.close()

    string = 'im' + str(n+1) + 'object.txt'
    file2 = open(string, 'r')
    data = []
    for line in file2:
        data.append([int(s) for s in line.split() if s.isdigit()])


    for obj in data:
        c = com[data.index(obj)]
        for i in range(obj[0], obj[2], 1):
            for j  in range(obj[1], obj[3], 1):
                z = depth[min(j,cols-1), min(i,rows-1)]
                x = (i - px)*z/f
                y = (j - py)*z/f
                cord = np.array((x,y,z))
                if (np.linalg.norm(c - cord) <= 3):
                    segmentation[min(j,cols-1),min(i,rows-1)] = data.index(obj)+1
    fig.add_subplot(3,1, n+1)

    plt.imshow(segmentation)
    file2.close()
plt.show()
```



from top to bottom are image segmentations for: ‘004945.jpg’, ‘004964.jpg’, ‘005002.jpg’

f) description of the scene:

```
import numpy as np


im1 = '004945'
im2 = '004964'
im3 = '005002'

for n in range(3):
    string = 'im' + str(n+1)+'object.txt'
    file1 = open(string, 'r')
    car=[]
    light=[]
    person=[]
    bicycle=[]
    index = 0
    for line in file1:
        name = line.split(':')[0]
        if (name=='car'):
            car.append(index)
        elif name =='traffic light':
            light.append(index)
        elif name =='bicycle':
            bicycle.append(index)
        else:
            person.append(index)
        index+=1
    file1.close()

    string = 'im' + str(n+1)+'com.txt'
    file2 = open(string, 'r')
    index = 0
    cardist = []
    lightdist = []
    persondist = []
    bicycledist = []
    for line in file2:
        dist = [float(s) for s in line.split()]
        if (index in car):
            if not cardist or cardist[0] > np.linalg.norm(dist):
                cardist = (np.linalg.norm(dist), dist[0])
        elif (index in light):
            if not lightdist or lightdist[0] > np.linalg.norm(dist):
                lightdist = (np.linalg.norm(dist), dist[0])
        elif (index in person):
```

```
            if not persondist or persondist[0] > np.linalg.norm(dist):
                persondist = (np.linalg.norm(dist), dist[0])
        else:
            if not bicycledist or bicycledist[0] > np.linalg.norm(dist):
                bicycledist = (np.linalg.norm(dist), dist[0])
        index+=1

    if (n==0):
        print ('\nin scene 004945.jpg')
    elif(n==1):
        print ('\nin scene 004964.jpg')
    else:
        print ('\nin scene 005002.jpg')

    print ('number of car is {}'.format(len(car)))
    if (cardist):
        if cardist[1] > 0:
            print ('closest is {} meters away to your right'.format(cardist[0]))
        else:
            print ('closest is {} meters away to your left'.format(cardist[0]))

    print ('number of traffic light is {}'.format(len(light)))
    if (lightdist):
        if lightdist[1] > 0:
            print ('closest is {} meters away to your
right'.format(lightdist[0]))
        else:
            print ('closest is {} meters away to your left'.format(lightdist[0]))

    print ('number of person is {}'.format(len(person)))
    if (persondist):
        if persondist[1] > 0:
            print ('closest is {} meters away to your
right'.format(persondist[0]))
        else:
            print ('closest is {} meters away to your
left'.format(persondist[0]))

    print ('number of bicycle is {}'.format(len(bicycle)))
    if (bicycledist):
        if bicycledist[1] > 0:
            print ('closest is {} meters away to your
right\n'.format(bicycledist[0]))
        else:
```

```
            print ('closest is {} meters away to your
left\n'.format(bicycledist[0]))
```

Output of the codes:

in scene 004945.jpg

number of car is 7

closest is 3.7348413183479052 meters away to your right

number of traffic light is 2

closest is 9.360302939519473 meters away to your right

number of person is 0

number of bicycle is 0

in scene 004964.jpg

number of car is 7

closest is 9.517486368237229 meters away to your left

number of traffic light is 2

closest is 5.635488195108173 meters away to your right

number of person is 3

closest is 6.047034671370383 meters away to your right

number of bicycle is 1

closest is 6.838956617303014 meters away to your right

in scene 005002.jpg

number of car is 3

closest is 8.2541778894947 meters away to your left

number of traffic light is 6

closest is 3.8935348512241266 meters away to your left

number of person is 0

number of bicycle is 0