CODE

CODE

import cv2

import os

import numpy as np

def getContours(img,i,img1):

h = img.shape[0]

w = img.shape[1]

for j in range(h\*8//10,h-1): #make lower pixels black so as to divide the segmented image into 'lanes'(works for

#my outputs (only fro some fo the images) as i didnt consider the white parts of the road

#when making the segmented image(without this the side lanes would come into the view from

#the side)

img[j,:]=0

contours,hierarchy = cv2.findContours(img,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_NONE)

for cnt in contours:

area=cv2.contourArea(cnt)

if area>5000: #get contours from the segmented image

minh=h//2

py=[]

for j in range(0,h):

if img[j,w//2]==255 or img[j,w//4]==255 or img[j,3\*w//4]==255:

minh=j #get approx minimum value of h where white pixel exists on segmented image

break

minh=minh+10

for j in range(0,w): #get x axis values where pixel is white when y axis is little below minh

#minh was increased by 10 as with minh the views were coming out as highly distorted

#as only few pixels at that height were white

#this gives us the upper set of points

if img[minh,j]:

py.append(j)

print(py)

minh=minh-10

pts1 = np.float32([[min(py),minh], [max(py),minh], [0,h], [w,h]]) #logically 0,h and w,h are relatively good

#estimates for two of the points when getting the

#lower set of points for the transform

print(pts1)

pts2 = np.float32([[0, 0], [w,0], [0,h], [w,h]])

matrix=cv2.getPerspectiveTransform(pts1,pts2) #get transformation matrix

print(matrix)

tf=cv2.warpPerspective(img1,matrix,(300,300)) #get birds eye view

cv2.imshow("tf"+str(i),tf)

return tf

f1=os.listdir("Resources")

avg=0

i=0

for imgname in f1:

path="Resources/"+imgname #take given images and segmented images

patho="output3/"+imgname

img=cv2.imread(patho,0)

img1 = cv2.imread(path)

i=i+1

print(i,"--------")

tf=getContours(img,i,img1)#birds eye view

cv2.imwrite("/Users/soumojitbhattacharya/PycharmProjects/road\_agv\_task/perspective/" + str(imgname), tf)

#add to file

cv2.waitKey(0)