PA#1: Implement Lane Detection Algorithm

Heechul Jung, Kyungpook National University 17-May-2020

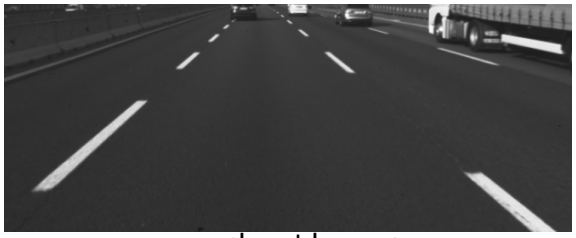
Prerequisite

- Programming Language: I recommend the Python language but you can use any kind of programming languages. e.g. Matlab, C, C++, JAVA ....
- Do not use any computer vision libraries to implement convolution, RANSAC, edge detection, or thresholding. e.g. OpenCV, Matlab Image Processing Toolkit ....
- Submit your code in a ".zip" file, and submit your report (".pdf" file) to explain kindly your implementation and to discuss your results. (Due: 11:59pm, 07-June-2020)
- Putting comments in your code.

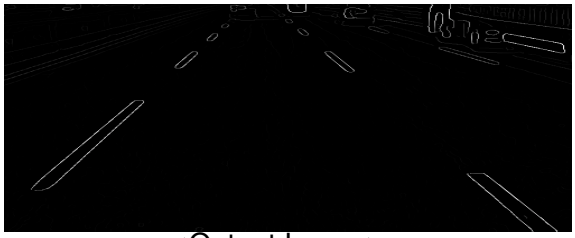
Task 0. Implement a function that convolves image with a given kernel. (20 pt)

Hint: https://en.wikipedia.org/wiki/Kernel_(image_processing)

Task 1. Detect Edges from the given image (lane.png) using the convolution function you implemented in Task 0 (20 pt)



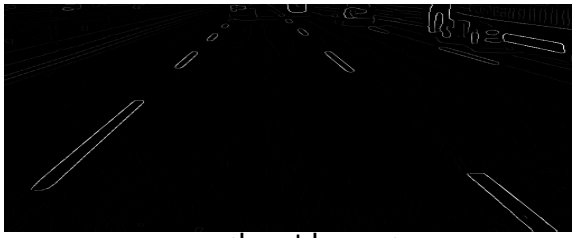
<Input Image>



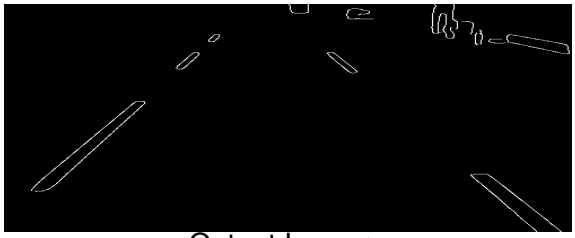
<Output Image>

Hint: You can use any edge detectors such as Sobel, Laplacian or Canny edge detectors.

Task 2. Thresholding the edge images. (10 pt)

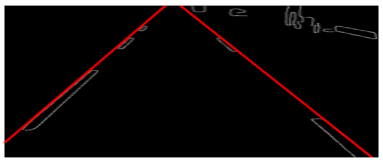


<Input Image>

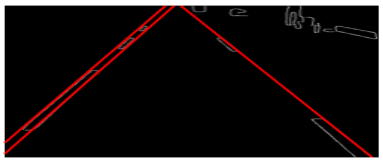


<Output Image>

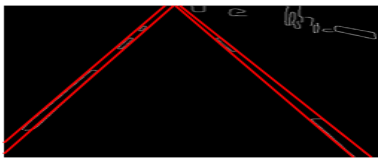
Task 3. Perform a RANSAC algorithm to detect lines which constitutes the lane on the Edge Image, and Draw the lines on the image. All the examples below are good results for this task.(50 pt)



<Result Image #1>



<Result Image #2>



<Result Image #3>

Hint: https://en.wikipedia.org/wiki/Random_sample_consensus

Hint: you can use cv2.line() to draw a line on the image.