My pipeline is made of three functions: get_vertices, get_images, and get_lines. Get_vertices returns the vertices when given the image, get_images returns the names of the the test images in a list, and get_lines is the main part of the pipeline and does the grayscaling, gaussian blurring, canny filtering, selection of the region of interest, and hough lines. The draw_lines function averages all the positive slope lines to find the left side of the lane, and all the negative ones to find the right side. Then, it draws lines on the edge of the lane, and draws a blue polygon in between them.
(I got the blue polygon idea from https://github.com/gardenermike/finding-lane-lines/blob/master/P1.ipynb)

My pipeline is not very good at dealing with curves. When I attempted the challenge, it never noticed the lane very far in front of itself and often got confused about where the lane was. Also, on occasion, it will not notice any lines.

If the pipeline used averaging of lane locations between frames in the videos, it would be more stable and be able to deal with frames with unclear lines. Also, improving the hough parameters would likely make it be able to filter out non lane boundary lines better, increasing the stability.