My pipeline consisted of 6 steps.

1. I converted the images to grayscale.
2. Then to smoothen the image i applied a gaussian blur with kernel size of 5.
3. Applied canny edge detection on smoothened image with low_threshold of 50 and high_threshold of 150.
4. Identified the region of interest by masking away unwanted portions of the image.
5. Retrieved Hough lines.
6. Applied lines to original image

In order to draw a single line on the left and right lanes, I added a new helper function draw_lines_and_merge() and then passed obtained average lines to draw_lines()

1. First i clasiffed left and right lanes with the help of left and right slopes.
2. Computed average and named it as final lines
3. passed final lines to draw_lines function
4. plot the extrapolated (final) line.

My pipeline currently works only for straight lanes, i tried it on the challenge video which has curved lanes and found that my pipeline failed to pass that, i'm still trying to figure out the problem and fix it.

There are many things which can be improved, here in Bangalore i tried to test it by taking a different image and video by capturing some local road lane images, but the problem is due to bad maintenance most of the lanes are erased and it was difficult to find a good straight road which has proper lane markings. So i feel we should not completely relay on lanes instead we need to find the path for the vehicle based on vehicle dimensions.