Lane Detection - Documentation

Instructions

- The program to be run is `main.py`.
- It asks the level of the video as a prompt (1,2 or 3) and accordingly detects the path in that video and shows it approppriately.
- The processed and raw videos can be viewed in the respective folders.
- All the functions used for this task are in the `functions.py` file.

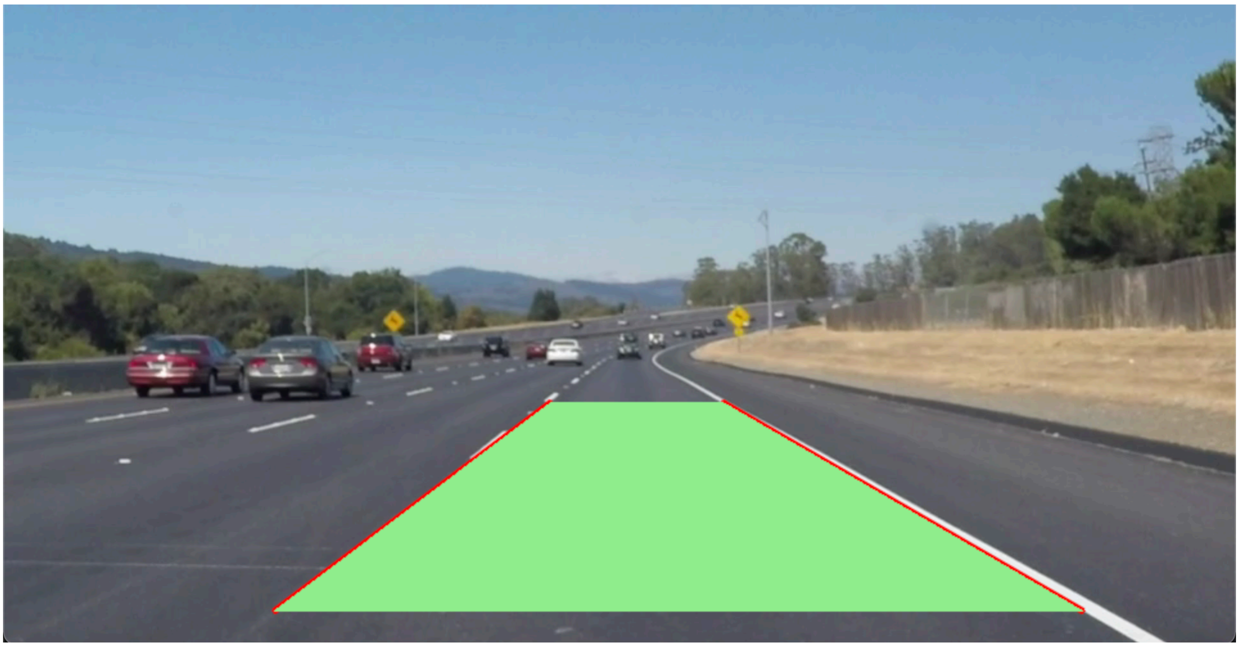
Process and Methods Used

- Opening the video frame-by-frame using OpenCV library
- Processing the video to isolate the lane lines:
    - **Note:** I Isolated the white and yellow lines separately due to the different coloration of the lines in the video.
    - I first converted it to HSV for better yellow color detection.
    - Created a masked image by first creating a mask based on lower and uppper HSV limits in case of yellow and simply the brightness (0 to 255) in case of white (in this case, the image was converted to grayscale beforehand), then performed bitwise AND on the original image.
    - Noise reduction was performed by using a Gaussian Blur.
    - Canny edge detection was performed on this to detect the edges of the image.
    - These edges might also contain background objects, so we define a **ROI (region of interest)** to only focus on the lane lines, and this ROI was essentially the lower 40% of the frame.
    - Further noise reduction and enhancement was performed by using erosion and dilation.
    - Hough transform was performed to detect the lines from this final processed video frame.
- The yellow and white lines which were detected were combined together, making sure either of them were initially empty - this was done by using the last detected lines instead, in case if it was empty. As the frames are just a few milliseconds apart, this is not a problem.
- Now that all the lines are detected, I determined the left-most and right-most lines by comparing the slopes.
- These two lines were drawn on to the video:
    - Each line was extended such that the y-coordinates were between 0.65 and 0.95 of the height of the image (rounded to the nearest integer) - this was done using the equation of straight line.
    - Now these two lines were drawn, as well as a trapezium that joins them, to denote the lane.

Result

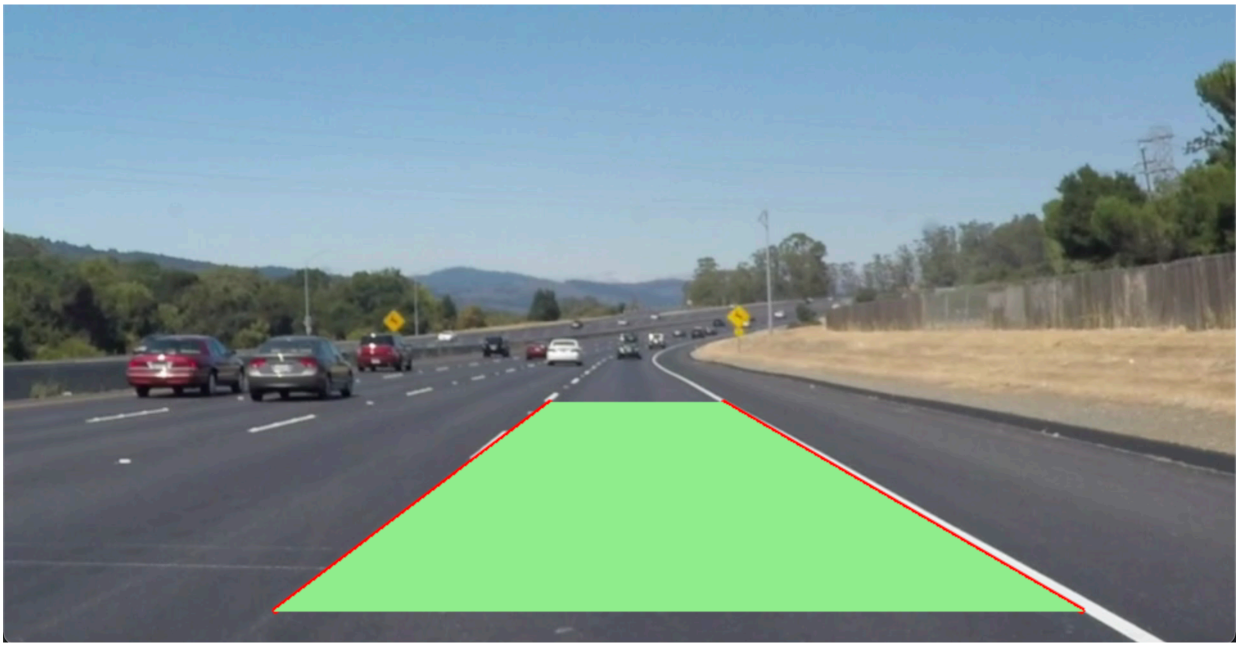
I was able to obtain a near-steady visualisation of the path using the above method, with some minor variations arising due to irregular lightning conditions, shaking of the camera, etc.

Here are some stills from each video:



Level 1

Level 2



Level 3

Another method that could have been implemented to avoid the above mentioned irregularities would have meen to modify the contrast of the video accordingly to account for the fluctuation in lighting.

References Used

- Color Image Segmentation — Image Processing
- What is Image Segmentation?
- Guide used for HSV ranges