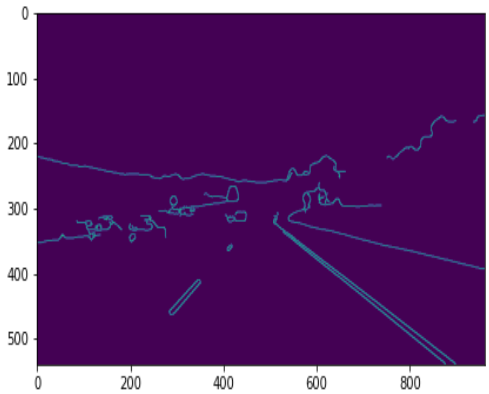
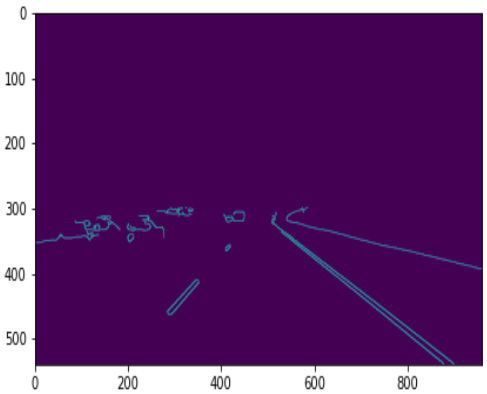
LANE DETECTION REPORT

Problem Statement: We need to fit a straight line to the lane in the image using Hough Transform

Approach: First I completed the image processing to reduce and filter out any Noice followed by isolating the region of interest using edge detection and then I applied the Hough lines algorithm to find and fit the straight line. I tried various filtering methods (like Laplacian) and finally settled with median blur as it was yielding the best result.



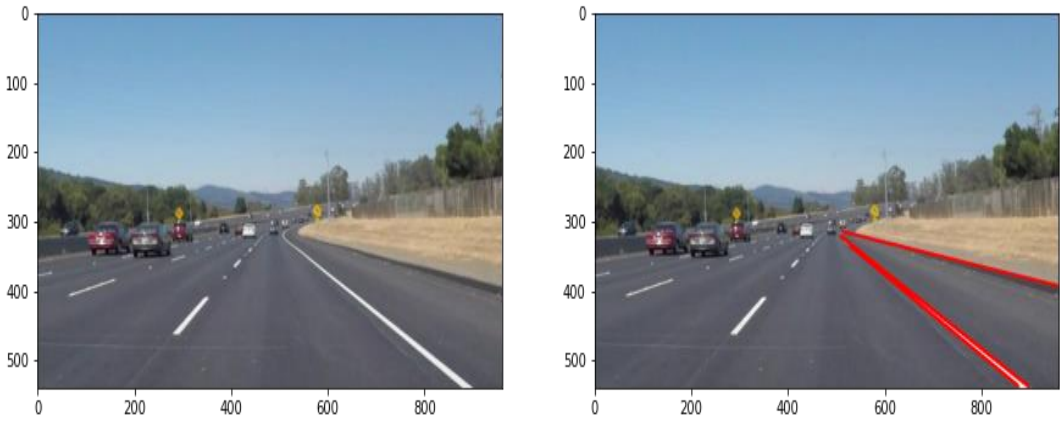


Hough line transform algorithm: Consider a point (x, y) in the xy-plane and the general equation of a straight line in slope-intercept

form, yi=ax+b. Infinitely many lines pass through (x, y), but they all satisfy the equation y=ax+bf or varying values of a and b. However, writing this equation as b=−xia+yi and considering the ab-plane (also called parameter space) yields the equation of a single line for a fixed pair (x, y). Furthermore, a second point (x, y) also has a line in parameter space associated with it, and, unless they are parallel, this line intersects the line associated with (x, y) at some point (a', b'), where a' is the slope and b' the intercept of the line containing both (x, y) and (x, y) in the xy-plane. In fact, all the points on this line have lines in parameter space that intersect at(a,b). In principle, the parameter-space lines corresponding to all points (xk, yk) in the xy-plane could be plotted, and the principal lines in that plane could be found by identifying points in parameter space where large numbers of parameter-space lines intersect.

In simple terms, Hough transform works when a set of discrete pixels are given, and it is to be checked whether these pixels lie on a straight line. if the pixels are found to lie on a straight line, then a line is drawn joining all these points with the help of Hough transform. In OpenCV Hough lines was implemented using the parameter space between (ρ,θ)), ρ=xcosθ)+ysinθ).

Final output:

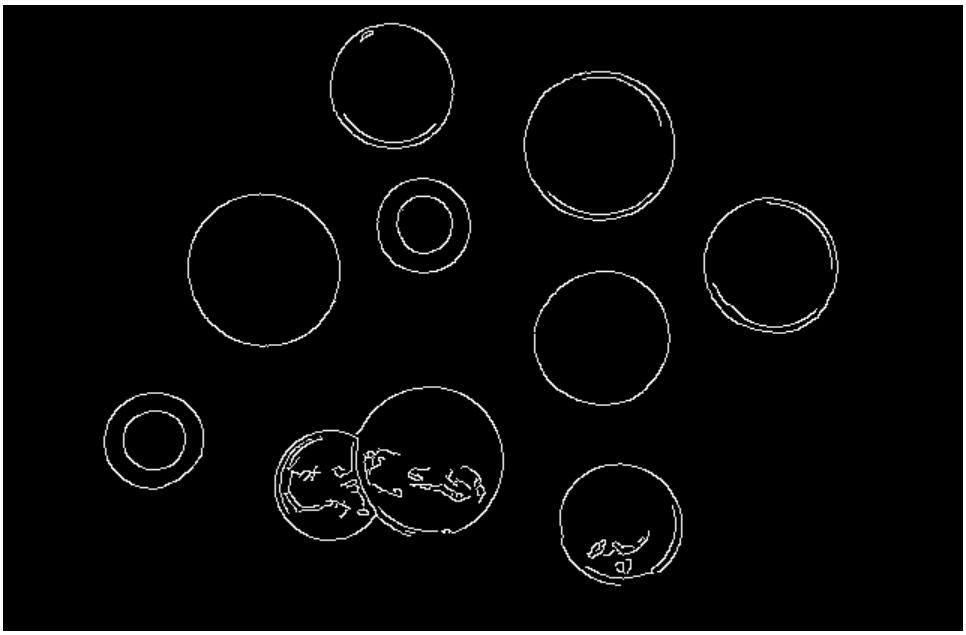


Result: our algorithm was on par with the inbuilt OpenCV function for Hough lines. We could further continue to progress by implementing algorithms to detect and form straight lines using just the non-continuous lane markers

Coin Detection Report

Problem Statement: we are given an image with coins; we need to use Hough Transform for Circles to detect the coins.

Approach: We first did image processing by using gaussian blur after converting the original image into grayscale and then I used gaussian blur. In Gaussian Blur operation, the image is convolved with a Gaussian filter instead of the box filter. The Gaussian filter is a low-pass filter that removes the high-frequency components. It was followed by using Canny edge detector to detect edges and isolate the coins region.



Result: The final output has been sensitive to noise even after filtering a can be seen from the final output. The algorithm also was getting confused when there were concentric circles. Otherwise, the algorithm worked well, especially when there were no nearby sharp edges of other coins to distract.