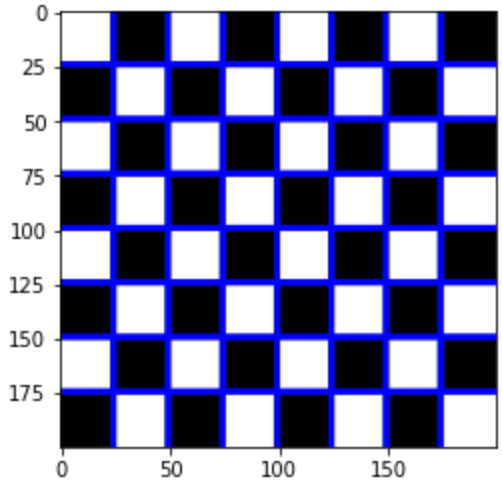
Hough-Transform-Implementation

Implemented the Hough transform algorithm for detecting the lines as well as the circles(coins). In this task I have implemented the Hough transform algorithm for detecting the lines as well as the circles(coins). Hough for Lines -: Hough transform lets every point on image vote. Using mathematical properties of the transform, this voting allows us to figure out the prominent lines in the image. Concept-: For Line detection lines are formed using 2 parameters(p and θ), where p is the length of normal from the origin and θ is the angle between normal and X-axis. The angle θ can vary from -90 degree to 90 degree and the length p can vary from 0 to the diagonal length. The equation for the line in P-θ space is given by p=x1 Cosθ+y1 Sinθ Where (x1,y1) is a point from where the line passes. A line in XY space is equivalent to the point in the p-θ space but a point in XY space is now equivalent to the sinusoidal wave in the p- θ space space.

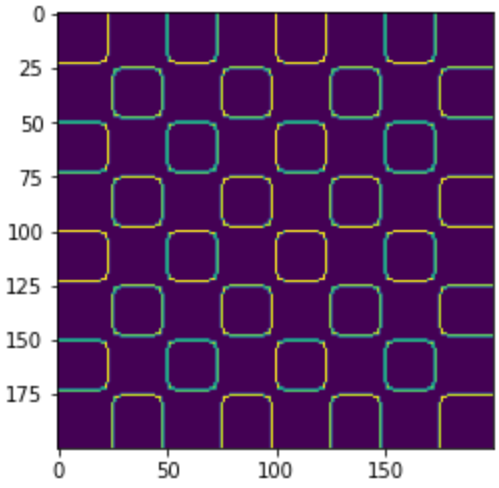
Implementation-:

1)For a given image we need to detect the edges first, I have used sobel operator for detecting the edges across the images.

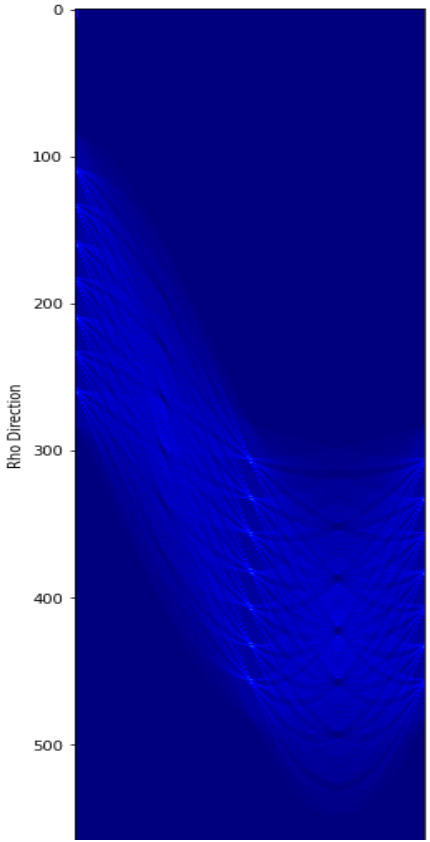


2)Now once the edges have been detected, we would be creating a p- θ space(accumulator) which would be a 2-D Array, such that its rows would represent the p and columns would be representing theta ,from -90 degree to 90 degree. The hough space is given as-:

3)We need to loop through every pixel location of edge detected image, and check if there is a non zero value. If the pixel value is non zero we compute p for that point location for all theta(-90 degree to 90 degree).Now for every theta and the computed p value we increase the already existing value in that particular p- θ cell by 1.



Approach Used-: I have computed indexes for all the cells in the accumulator who have got value more than 90 votes. These indexes if plotted using the method cv2.line will be giving lot of lines over the screen in all the directions . Thus to filter the black lines which overlap red lines, I have chosen the lines which have got θ=-2. This will give the cluster of black lines over each red line . To resolve that I have taken the centre line(line at median position) for each cluster, thus this gives me 6 black lines which overlap with the red lines in the image as shown above.



Blue Lines Detection-: Number of blue lines detected=8.

Approach Used-: I have computed indexes for all the cells in the accumulator who have got value more than 120 votes. This would be giving lots of lines on the image. I have filtered black lines lines at θ =-36 degree and have used median logic(as used for red line) for getting a single line from the cluster of lines.

Approach Used-: I have computed indexes for all the cells in the accumulator who have got value more than 0.529 * maximum value of the (a,b,r)space, which will be local maximum value. Thus plotting these indexes using cv2.circle method. The value for the radius has been taken from 21-23(inclusive), which has been been figured using heuristic approach.