



## Step4:

Using Canny operator from cv2,detect the edges of the image.

## Step5:

Using the HoughLinesP(),detect line co-ordinates for every points in the images.Using For loop,draw the lines on the found co-ordinates.Display the image.

## Program:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
image = cv2.imread('imgme.jpg')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) # Convert image to RGB
for displaying
    plt.title("Input Image")
    plt.axis('off')
    plt.imshow(gray_image, cmap='gray')
    plt.title("Grayscale Image")
    plt.axis('off')
edges = cv2.Canny(gray_image, 13, 130)

plt.imshow(edges, cmap='gray')
plt.title("Canny Edge Detector")
plt.axis('off')
lines = cv2.HoughLinesP(edges, 1, np.pi / 180, 100, minLineLength=50,
maxLineGap=10)

for line in lines:
    x1, y1, x2, y2 = line[0]
    cv2.line(image, (x1, y1), (x2, y2), (0, 255, 0), 2)

plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title("Result of Hough Transform")
plt.axis('off')
```



## Output

### Input image and grayscale image

## Input Image



## Grayscale Image



## Canny Edge detector output

## Canny Edge Detector

[Collapse Output](#)



Display the result of Hough transform

## Result of Hough Transform



### Result:

Thus the code to perform Edge detection using Hough Transform was written and executed successfully.