

```

# EXP:07 Edge-Linking-using-Hough-Transform
# NAME: ABISHEK PV
# REF NO: 21222230003

import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

# Step 1: Upload the image in Colab
uploaded = files.upload() # Upload your image file here
image_path = list(uploaded.keys())[0]

# Step 2: Load the image using imread() from cv2 module
image = cv2.imread(image_path)

# Step 3: Convert the image to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Input image and grayscale image
plt.figure(figsize=(10,8))
plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) # Convert image to RGB for displaying
plt.title("Input Image")
plt.axis('off')
(np.float64(-0.5), np.float64(412.5), np.float64(530.5), np.float64(-0.5))

plt.subplot(1, 2, 2)
plt.imshow(gray_image, cmap='gray')
plt.title("Grayscale Image")
plt.axis('off')
(np.float64(-0.5), np.float64(412.5), np.float64(530.5), np.float64(-0.5))

plt.show()

# Step 4: Using Canny operator from cv2, detect the edges of the image
edges = cv2.Canny(gray_image, 50, 150) # Canny edge detection with threshold values 50 and 150

# Canny Edge Detector output
plt.imshow(edges, cmap='gray')
plt.title("Canny Edge Detector")
plt.axis('off')
(np.float64(-0.5), np.float64(412.5), np.float64(530.5), np.float64(-0.5))
plt.show()

# Step 5: Using the HoughLinesP(), detect line coordinates for every point in the image
# The parameters of HoughLinesP are: image, resolution, threshold, minLineLength, maxLineGap
lines = cv2.HoughLinesP(edges, 1, np.pi / 180, 100, minLineLength=50, maxLineGap=10)

# Step 6: Using a for loop, draw the lines on the original image using the detected coordinates
for line in lines:
    x1, y1, x2, y2 = line[0] # Unpacking the line coordinates
    cv2.line(image, (x1, y1), (x2, y2), (0, 255, 0), 2) # Draw green lines with thickness of 2

# Display the result of Hough Transform (Image with lines)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) # Image with lines drawn
plt.title("Result of Hough Transform")
plt.axis('off')
(np.float64(-0.5), np.float64(412.5), np.float64(530.5), np.float64(-0.5))
plt.show()

```

Choose Files WIN\_20251...\_27\_Pro.jpg

WIN\_20251129\_07\_05\_27\_Pro.jpg(image/jpeg) - 117184 bytes, last modified: 29/11/2025 - 100% done

Saving WIN\_20251129\_07\_05\_27\_Pro.jpg to WIN\_20251129\_07\_05\_27\_Pro.jpg

Input Image



Grayscale Image



Canny Edge Detector



Result of Hough Transform

