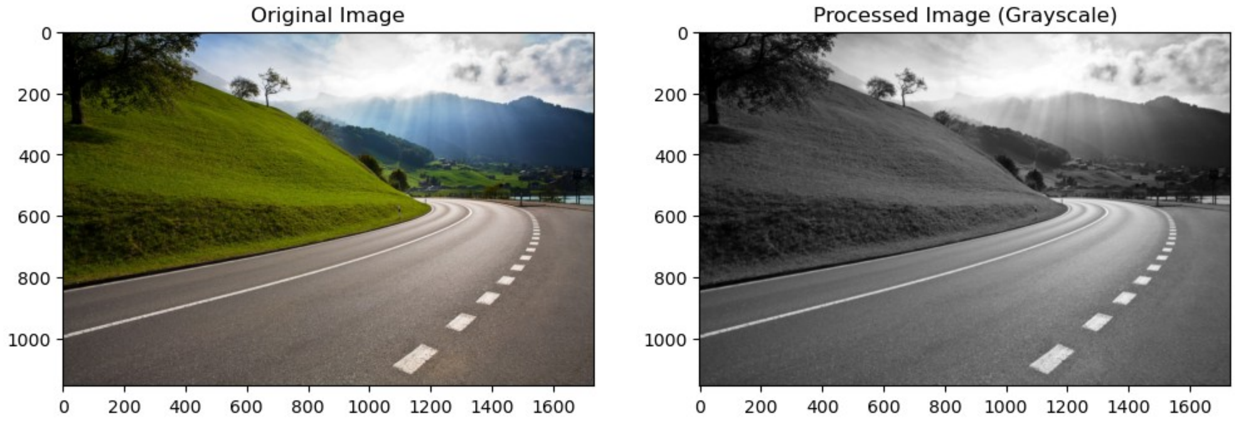
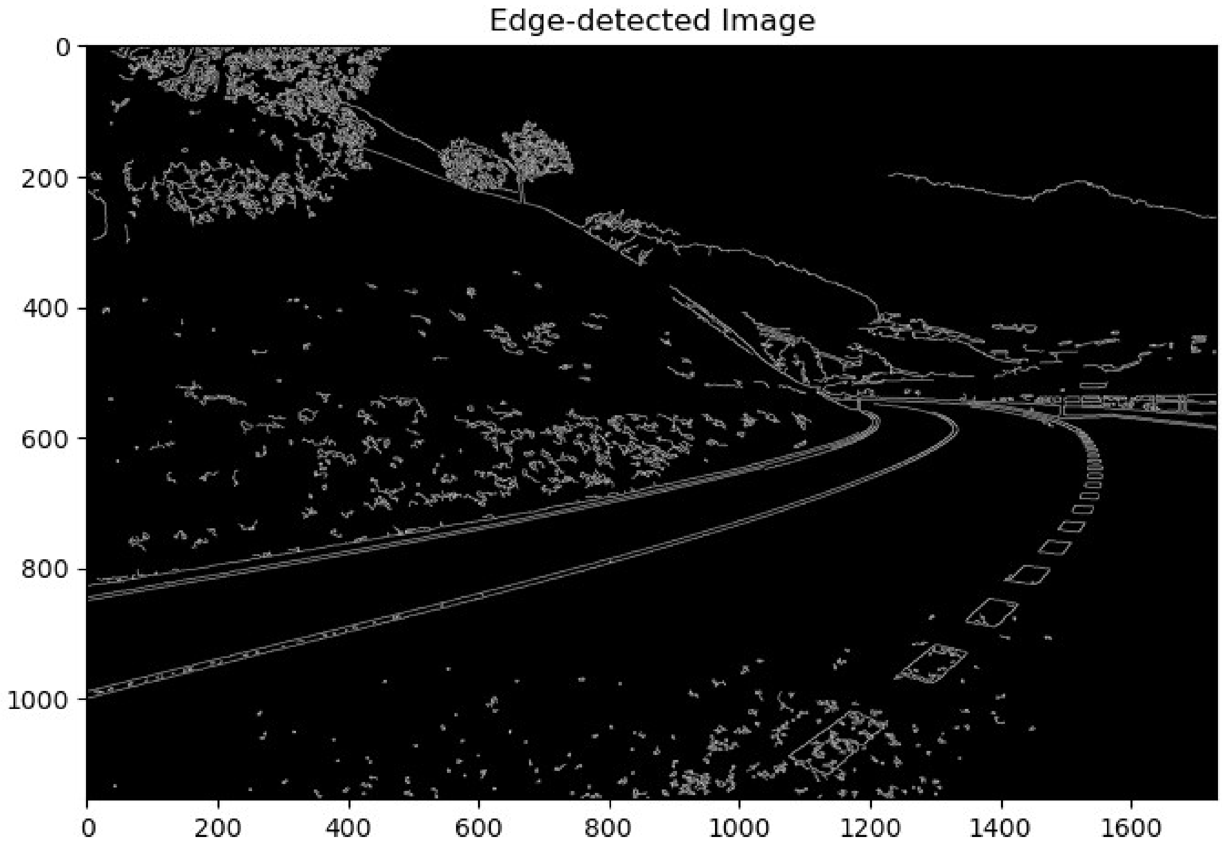
```
!pip install numpy opencv-python matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load image
image_path = 'road_image.jpg'  # Replace with your image path
image = cv2.imread(image_path)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  # Convert to RGB
(OpenCV uses BGR by default)

# Convert to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)

# Apply Gaussian blur
kernel_size = 5
blur_gray = cv2.GaussianBlur(gray, (kernel_size, kernel_size), 0)

# Display the original and processed image
plt.figure(figsize=(12, 6))
plt.subplot(121)
plt.imshow(image)
plt.title('Original Image')

plt.subplot(122)
plt.imshow(blur_gray, cmap='gray')
plt.title('Processed Image (Grayscale)')
plt.show()

# Define parameters for Canny edge detection
low_threshold = 50
high_threshold = 150
edges = cv2.Canny(blur_gray, low_threshold, high_threshold)

# Display the edge-detected image
plt.figure(figsize=(8, 6))
plt.imshow(edges, cmap='Greys_r')
plt.title('Edge-detected Image')
plt.show()

# Define a region of interest polygon (triangle for simplicity)
mask = np.zeros_like(edges)
ignore_mask_color = 255

# Define vertices of the triangular ROI
imshape = image.shape
vertices = np.array([[(1000000, imshape[0]), (imshape[1] // 2,
imshape[0] // 2 + 50), (imshape[1] - 100, imshape[0])]],
dtype=np.int32)
cv2.fillPoly(mask, vertices, ignore_mask_color)
```

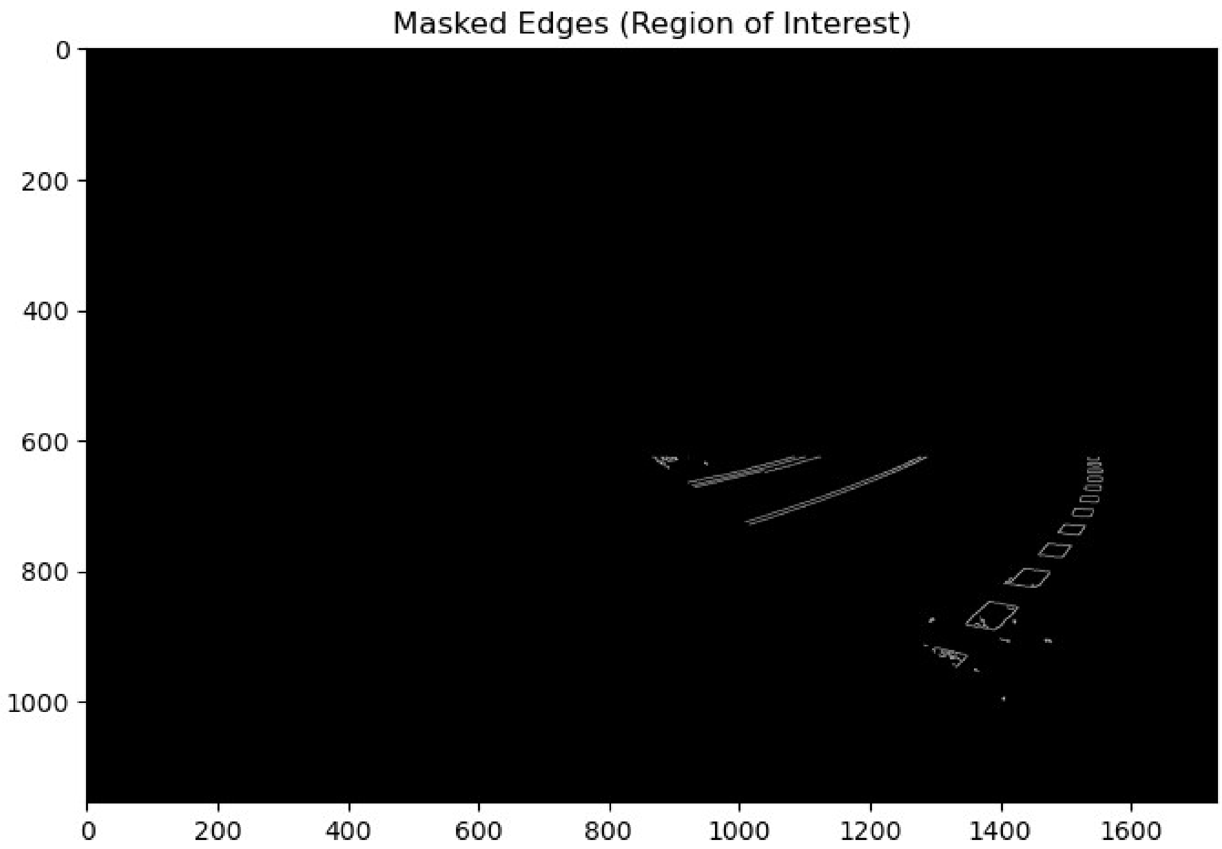
```
masked_edges = cv2.bitwise_and(edges, mask)

# Display the masked edges
plt.figure(figsize=(8, 6))
plt.imshow(masked_edges, cmap='Greys_r')
plt.title('Masked Edges (Region of Interest)')
plt.show()

# Define Hough transform parameters
rho = 1  # distance resolution in pixels of the Hough grid
theta = np.pi/180  # angular resolution in radians of the Hough grid
threshold = 15  # minimum number of votes (intersections in Hough grid
cell)
min_line_length = 40  # minimum number of pixels making up a line
max_line_gap = 20  # maximum gap in pixels between connectable line
segments

# Run Hough transform on edge-detected image
lines = cv2.HoughLinesP(masked_edges, rho, theta, threshold,
np.array([]),
                        min_line_length, max_line_gap)

# Create a blank image to draw lines on
line_image = np.zeros((masked_edges.shape[0], masked_edges.shape[1],
3), dtype=np.uint8)

# Iterate over detected lines and draw them on the blank image
for line in lines:
    for x1, y1, x2, y2 in line:
        cv2.line(line_image, (x1, y1), (x2, y2), (255, 0, 0), 5)

# Overlay lines image on original image
lines_edges = cv2.addWeighted(image, 0.8, line_image, 1, 0)

# Display the final image with detected lanes
plt.figure(figsize=(12, 6))
plt.imshow(lines_edges)
plt.title('Lane Lines Detected')
plt.show()
```

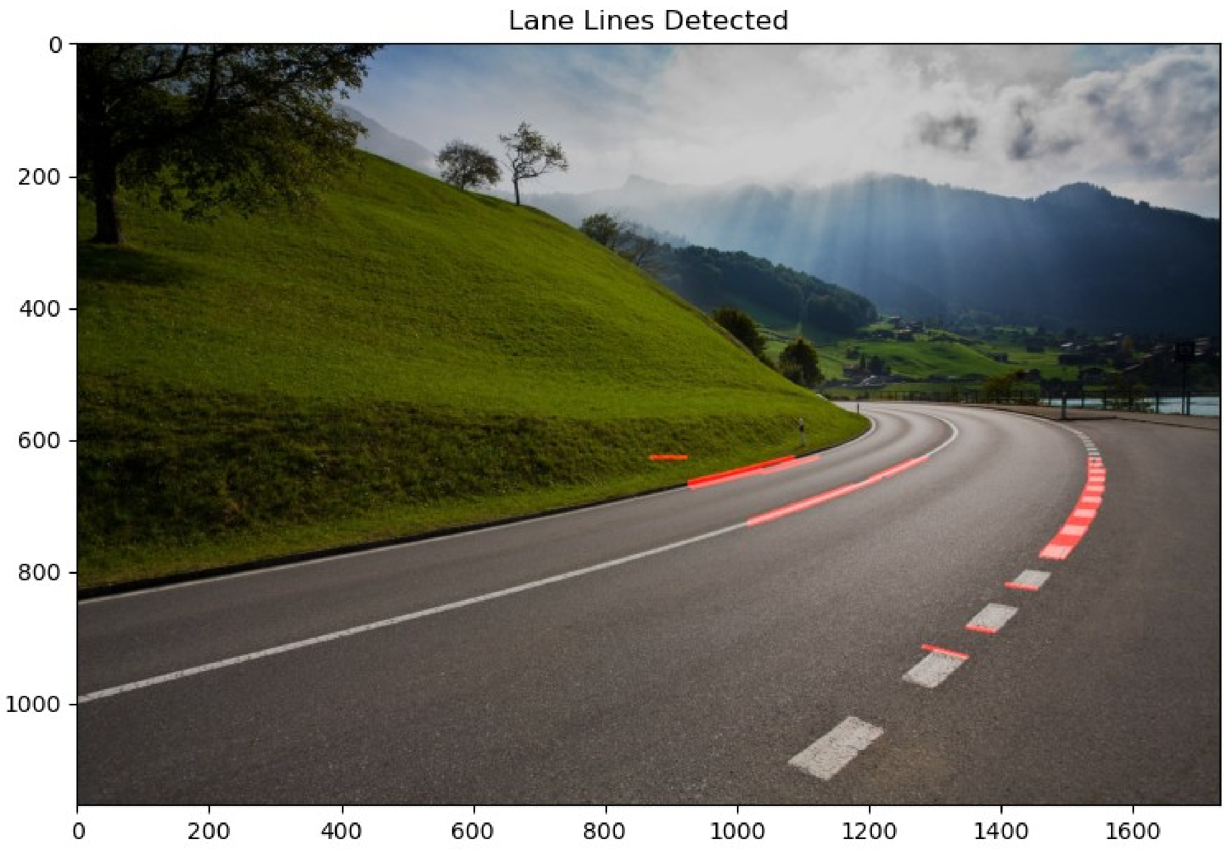
```
Requirement already satisfied: numpy in c:\users\windows\downloads\
mani\anaconda\lib\site-packages (1.26.4)
Requirement already satisfied: opencv-python in c:\users\windows\
appdata\roaming\python\python311\site-packages (4.8.0.74)
Requirement already satisfied: matplotlib in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (0.11.0)
```

```
Requirement already satisfied: fonttools>=4.22.0 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\windows\
downloads\mani\anaconda\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
windows\downloads\mani\anaconda\lib\site-packages (from matplotlib)
(2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\windows\downloads\
mani\anaconda\lib\site-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)
```




Edge-detected Image
0
200
400
600
800
1000
0
200
400
600
800
1000
1200
1400
1600


Masked Edges (Region of Interest)
0
200
400
600
800
1000
0
200
400
600
800
1000
1200
1400
1600


Lane Lines Detected
0
200
400
600
800
1000
0
200
400
600
800
1000
1200
1400
1600