

LANE DETECTION

```
import cv2

import numpy as np

def lane_detection(image):

    # Get image dimensions

    height, width = image.shape[:2]

    # Convert the image to grayscale

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur to reduce noise

    blur = cv2.GaussianBlur(gray, (5, 5), 0)

    # Use Canny edge detection

    edges = cv2.Canny(blur, 50, 150)

    # Define a region of interest

    polygons = np.array([

        [(0, height), (600, height // 2), (width - 200, height // 2), (width, height)]

    ])

    mask = np.zeros_like(edges)

    cv2.fillPoly(mask, polygons, 255)

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

```
# Detect lines using Hough Transform

lines = cv2.HoughLinesP(cropped_edges, 1, np.pi / 180, threshold=100, minLineLength=100,
maxLineGap=10)

# Draw the lines on the original image

line_image = np.zeros_like(image)

if lines is not None:

    for line in lines:

        x1, y1, x2, y2 = line[0]

        cv2.line(line_image, (x1, y1), (x2, y2), (0, 255, 0), 3)

# Combine the original image with the line image

combined = cv2.addWeighted(image, 0.8, line_image, 1, 0)

return combined


# Load an image

image = cv2.imread(r"C:\Users\admin\Desktop\road.jpg")

lane_detected_image = lane_detection(image)


# Show the result

cv2.imshow("Lane Detection", lane_detected_image)

cv2.waitKey(0)

cv2.destroyAllWindows()
```