Software Subsystem

Q2: Lane Marking Extraction and Segmentation

REPORT

APPROACH-

I have used OpenCV and Numpy for processing . The basic steps included:

1. **Color Space Transformation:** Converted from BGR (OpenCV default) to HSV color space for better color-based segmentation
2. **Color Thresholding:** Isolated white and yellow lane markings based on their HSV color ranges
3. **Noise Reduction:** Applied Gaussian blur to reduce noise
4. **Edge Detection:** Used Canny edge detection to identify lane boundaries
5. **Region of Interest (ROI) Selection:** made focused analysis on the relevant portion of the image
6. **Line Detection:** Applied Hough Line Transform to identify straight lines
7. **Mask Generation:** Created separate masks for white and yellow lane markings
8. **Morphological Operations:** Cleaned up the masks to remove noise and fill gaps
9. **Visualization:** Combined the masks with the original image for clear visualization

Detailed Implementation and their reasons:

a. **The HSV (Hue, Saturation, Value) color space was chosen because it separates color information (hue) from intensity (value) and saturation). This is particularly important for lane detection where**

shadows and lighting conditions can significantly impact RGB values.

b. Two separate masks were created:

- **White Lane Mask:** Used high value (brightness) and low saturation parameters to capture white lanes.
- **Yellow Lane Mask:** Used specific hue range around yellow combined with appropriate saturation and value ranges.

I have chosen the threshold values by surfing through net and selecting the one which is typically found in real road conditions.

c. After applying Gaussian blur to reduce noise, Canny edge detection was used to identify strong edges in the image. The region of interest was defined as the bottom half of the image to focus on the road area and reduce computation on irrelevant parts of the image.
d. To improve the quality of the masks, morphological operations was applied to:

   Fill small gaps in detected lane markings

   Remove isolated pixels (noise)

   Connect nearby line segments that belong to the same lane marking

e. This approach avoids the use of machine learning or deep learning models, following the constraint of using traditional image processing techniques.

#As a beginner the approach i have used might have some limitations:

- HSV thresholds may need adjustment for different lighting conditions
- The ROI is manually defined and might not be optimal for all road scenarios

- Curved lane markings might not be perfectly captured by Hough Line Transform

  However the implemented solution successfully extracts and highlights white and yellow lane markings using purely traditional image processing techniques.

Submission Files:

- `highlighted.png` (Highlighted lanes on the original image)
- `white_mask.png` (White lane binary mask)
- `yellow_mask.png` (Yellow lane binary mask)
- `LaneDetection.py` (Python implementation)
- `report_LaneDetection.pdf` (This report)