COL780 Assignment - 1

Vidushi Maheshwari (2021CS10083)

February 18, 2025

Contents

Introduction

Image preprocessing is an essential step in computer vision, enhancing image quality by reducing noise, increasing contrast, and emphasizing critical features. This report details the implementation of a lane boundary detection algorithm using image processing techniques. The assignment consists of two tasks:

1. Detecting and overlaying lane boundaries on input images.
2. Evaluating the quality of detected lines in grass images using intersection-based analysis.

Task 1

§1 Objective

Detect lane boundaries in the given images using edge detection and line/curve fitting methods. Overlay the detected lanes on the original images for visualization.

§2 Methodology

§2.1 Preprocessing

Color-based Segmentation

- Into grassy surfaces and red/brown dirt pathways
- Extracted brown regions from the image
- If the percentage of brown pixels > threshold, call `red_image(img)` else, call `grass_image(img)`

Dirt Pathways `red_image(img)`

- The brown mask is subtracted from the grayscale version of the image to suppress brown regions while preserving lane markings.
- A Gaussian blur is applied to remove noise and smooth the image:
- The Sobel operator is applied in the x-direction to detect vertical lane markings.
- A gradient mask is created by thresholding the Sobel gradient to highlight significant edges and the brown mask is subtracted to remove unwanted regions.

Grassy lanes `green_image(img)`

- White pixels are extracted using an intensity threshold across all three color channels.
- Non-white pixels are set to 0, leaving only the lane markings.

Gaussian Blurring

To remove noise due to grass and fallen leaves.

§2.2 Edge Detection

Performed using the Canny edge technique. It detects strong edges in the preprocessed image, highlighting areas where lane markings are present. The Canny method works by:

- Computing gradients in the x and y directions using Sobel filters.
- Calculating the magnitude of the gradient to identify the intensity of the edge.
- Thresholding the gradient magnitude to classify strong and weak edges.

§2.3 Hough transform for straight lane detection

Straight boundaries are detected using the Hough transform method.

- The image is transformed into a polar coordinate space where each point is represented as a line with parameters ρ (distance from origin) and θ (angle).
- For each edge pixel in the image, its corresponding line equation in polar coordinates is computed.
- An accumulator matrix is created to count the votes for potential lines.
- Lines that receive votes above a certain threshold are selected, and the corresponding polar coordinates are converted back to Cartesian coordinates.
- Lines shorter than a predefined minimum length are filtered out to remove noise.

§2.4 Polynomial fitting for curved lanes

Curved boundaries are detected using the Polynomial fitting method.

- Edge points are divided into left and right lane based on their relative position to others.
- Applied polynomial fitting of degree 2 to the edge points separately for left and right lane boundaries.
- The polynomial curves are then evaluated for different y-coordinates to obtain their corresponding position and visualised on the image.

§2.5 Filtering the Output

- **Non-ROI lines**: To improve the accuracy of the lane detection, lines that do not fit within the Region of Interest (ROI) are filtered. Lines that are completely restricted to upper portion of image are more likely to be background noises and hence, filtered out.
- **Removing edge clusters**: The lines that are too proximal or intersect with each other at an angle less than the threshold are treated as duplicate. This removes the less significant edges and refines the detected lanes boundary.

§2.6 NOTE: Grass images

- In some images (like image 39), white mask with high threshold (around 230) fails to capture the difference between grass and white lane properly due to lack of high contrast, noisy texture of grass and lighting conditions.
- Such images might go without having the edges detected and end up with 0 lines at the end of process.
- Such cases are checked at the end of lane detection and if some image has 0 edges, it is again processed with a lower threshold (around 200).
- This improved the end result for multiple images including image 37 to 40.

§3 Results

§3.1 Preprocessing

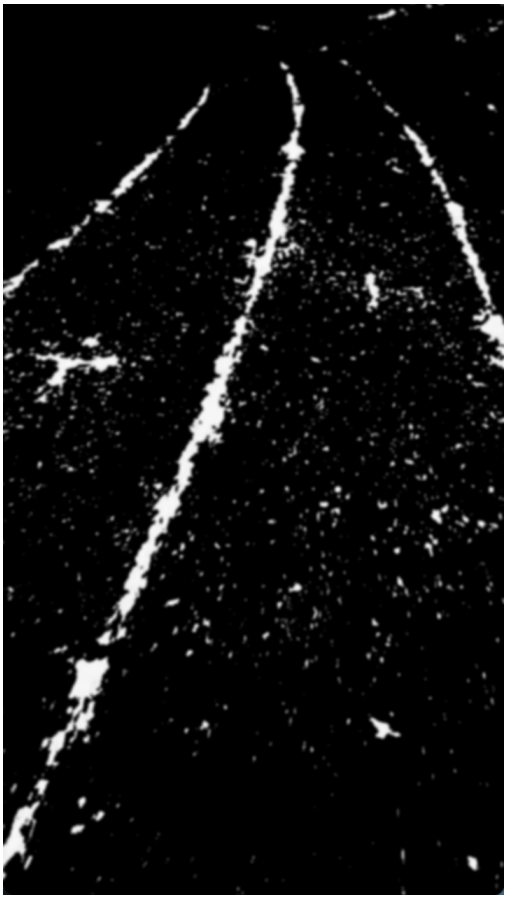

Figure 1: Image 27

Figure 2: Image 45

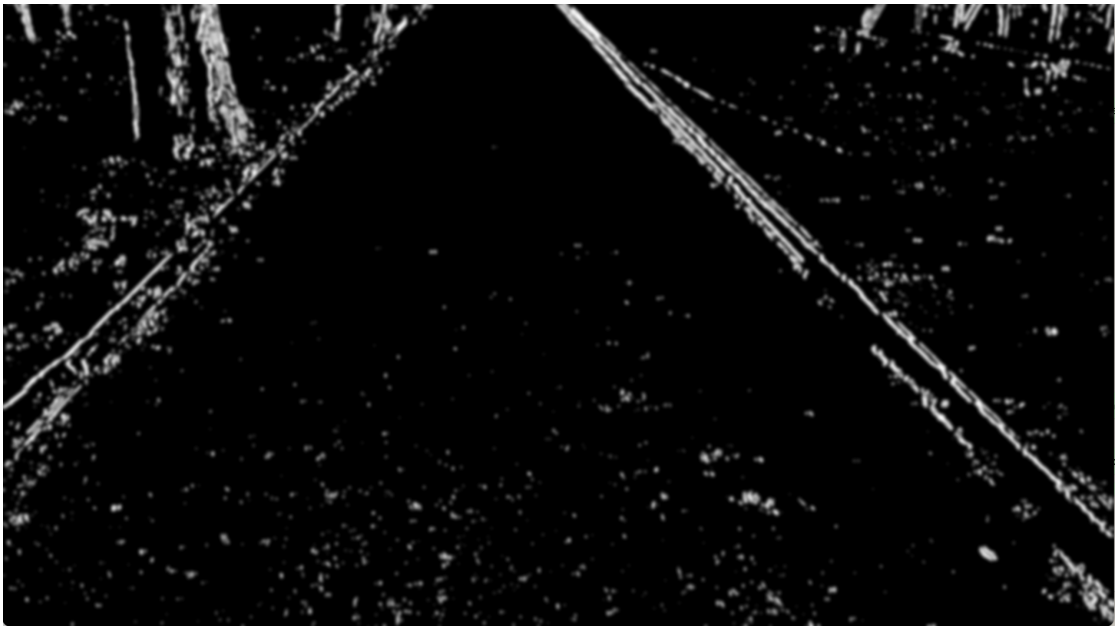

Figure 3: Image 42

§3.2 Edge Detection

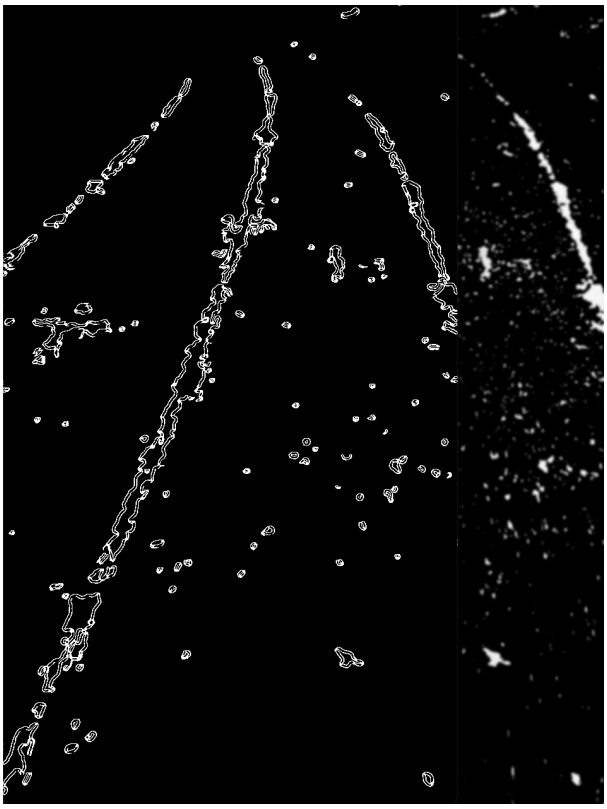

Figure 4: Image 27

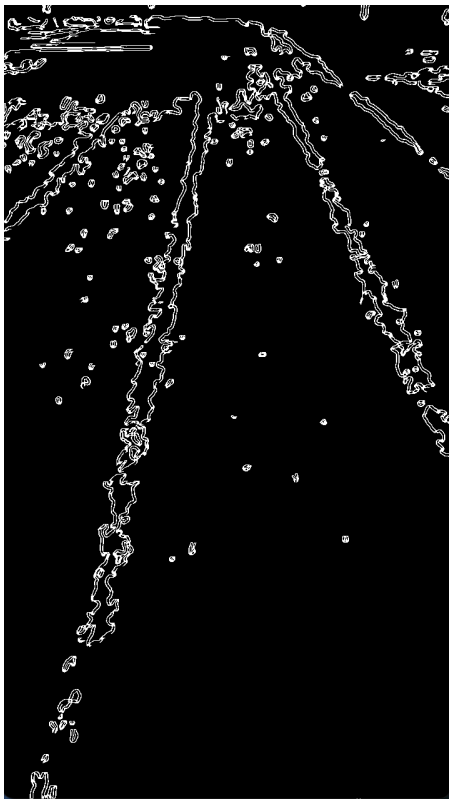

Figure 5: Image 45

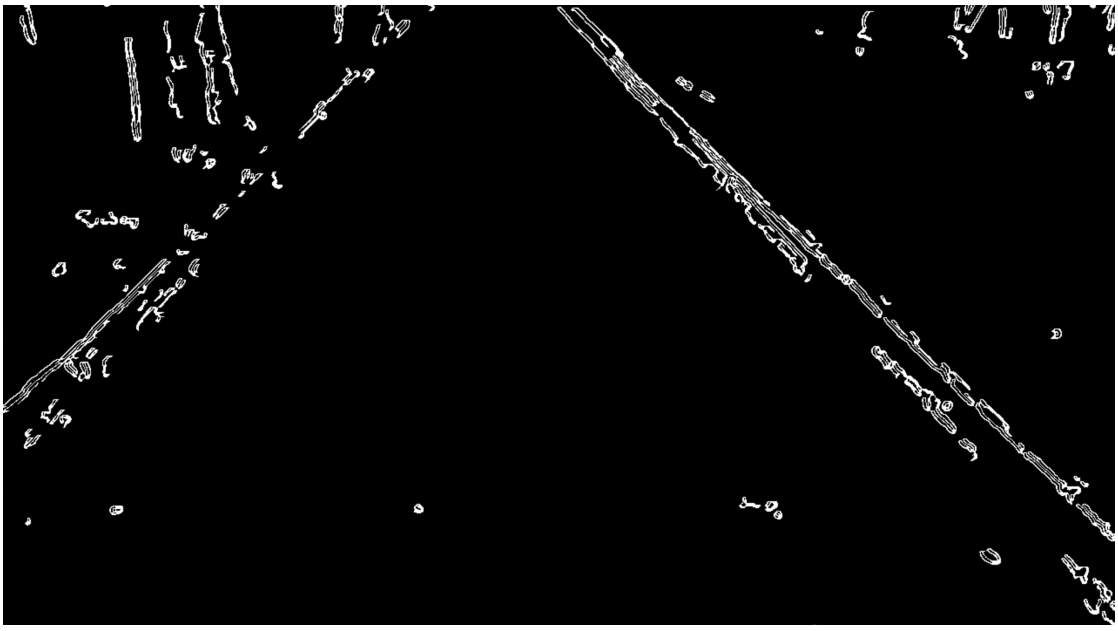

Figure 6: Image 42

§3.3 Final Image

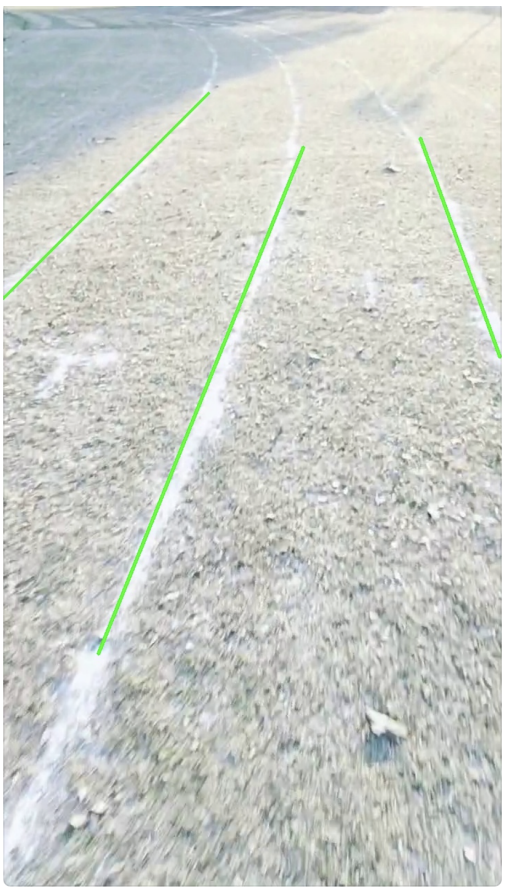

Figure 7: Image 27

Figure 8: Image 45

Figure 9: Image 42

Task 2

§4 Objective

This task focuses on analyzing the quality of detected lines in grass images to assess lane detection performance.

§5 Methodology

- Pre-processing using `grass_image()` function.
- Edge detection using Canny, line detection using Hough's Transform and filtering the obtained output as done in Task 1
- For calculating the quality metric, intersections between each pair of lines is computed. Then, the intersection coordinates are used to determine the centroid. The sum of distance of each point from the centroid is used as a metric to assess the quality of line detection.

§6 Results

Filename	Line-Fit Score
image_3.jpg	2235.54
image_4.jpg	0.00
image_6.jpg	3844.47
image_11.jpg	559.83
image_15.jpg	1788.28
image_16.jpg	0.00
image_17.jpg	1320.52
image_19.jpg	0.00
image_20.jpg	0.00
image_25.jpg	0.00
image_26.jpg	0.00
image_27.jpg	181.10
image_32.jpg	137.60
image_34.jpg	0.00
image_35.jpg	0.00
image_37.jpg	0.00
image_38.jpg	0.00
image_39.jpg	1776.63
image_40.jpg	0.00
image_41.jpg	0.00
image_43.jpg	1716.42
image_45.jpg	0.00
image_46.jpg	0.00

Table 1: Line-Fit Scores for Detected Lane Boundaries

NOTE: For lines 19 and 20, the score is 0 because of only single line being clearly detected in the image. For others, the 0 score is because of clear lane detected with 2 lines only.

Conclusion

The lane detection method could be improved by follwoing measures

- Robust Edge Detection: Improve edge detection by experimenting with adaptive thresholding or combining multiple edge detection methods for more accurate boundary identification, especially under varying lighting conditions.
- Dynamic Parameter Selection: Implement adaptive parameter tuning for Hough Transform and polynomial fitting to enhance detection performance across different road scenes and conditions.
- Better Line Filtering: Refine line filtering logic to exclude outliers and improve the quality of detected lanes, especially in grass images for intersection analysis.
- Efficiency: More efficient and improvised implementations can be explored, as methods like Hough Transform were a bottleneck to the time complexity. Optimizing the Hough Transform, exploring alternative line detection methods, or implementing parallel processing techniques could significantly enhance the efficiency and scalability of the system