ENHANCING TRAFFIC SIGN DETECTION FOR AUTONOMOUS VEHICLES

Presented By
Ramazan Bozkurt 200254031
Abdulkadir Sözgen 210254031

INTRODUCTION

Objective:

- To detect and highlight traffic signs from images under challenging conditions (e.g., low contrast, noisy backgrounds).

Problem Statement:

- Traffic signs can be hard to detect in low-contrast or noisy environments.
- Autonomous systems require accurate and reliable detection for safe navigation.

METHODOLOGY OVERVIEW

1. Image Preprocessing:

- Convert input image to grayscale.
- Enhance contrast using CLAHE (Contrast Limited Adaptive Histogram Equalization).

2. Noise Reduction:

- Apply Gaussian Blur to reduce random noise.

3. Edge Detection:

- Use Canny Edge Detection to identify potential traffic sign contours.

METHODOLOGY OVERVIEW CONTINUING

4. Contour Analysis:

- Analyze shapes and filter based on size to detect traffic sign candidates.

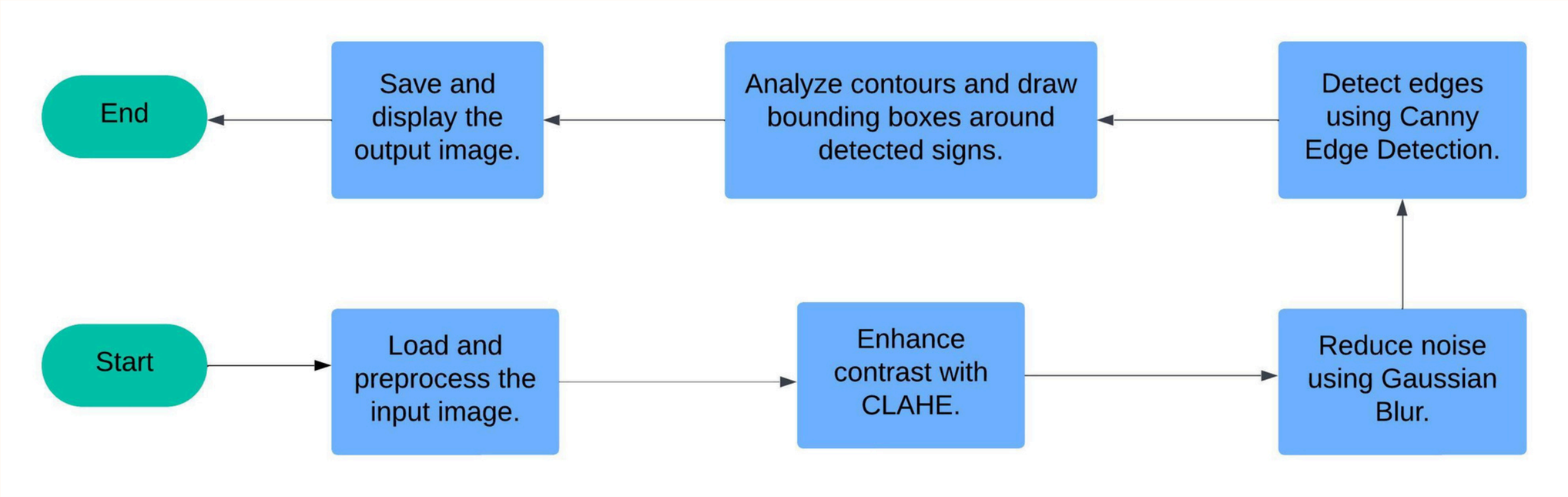
5. Output Generation:

- Highlight detected signs on the enhanced image.

TOOLS AND TECHNOLOGIES

- Programming Language:
  - Python
- Libraries:
  - OpenCV
- Techniques Used:
  - CLAHE for contrast enhancement.
  - Gaussian Blur for noise reduction.
  - Canny Edge Detection for edge detection.

WORKFLOW


Start
Load and preprocess the input image.
Enhance contrast with CLAHE.
Reduce noise using Gaussian Blur.
Detect edges using Canny Edge Detection.
Analyze contours and draw bounding boxes around detected signs.
Save and display the output image.
End

RESULTS

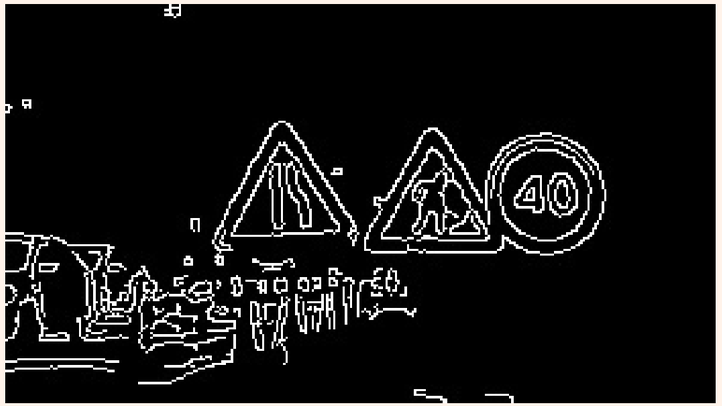
Input Image

Enhanced Image

Final Output

Blurred Image

Edges Detected



RESULTS CONTINUING

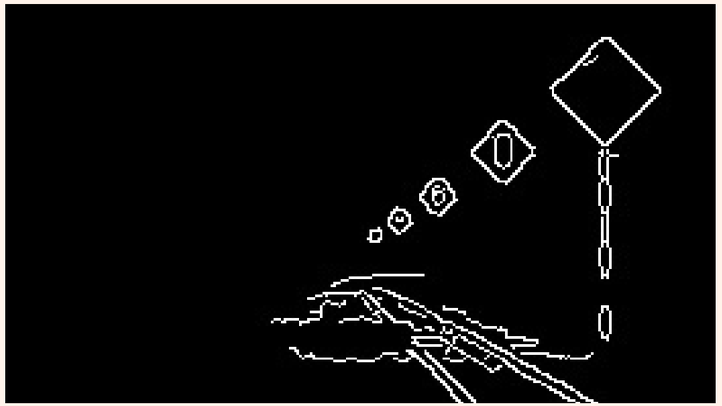
Input Image

Enhanced Image

Final Output

Blurred Image

Edges Detected



RESULTS CONTINUING

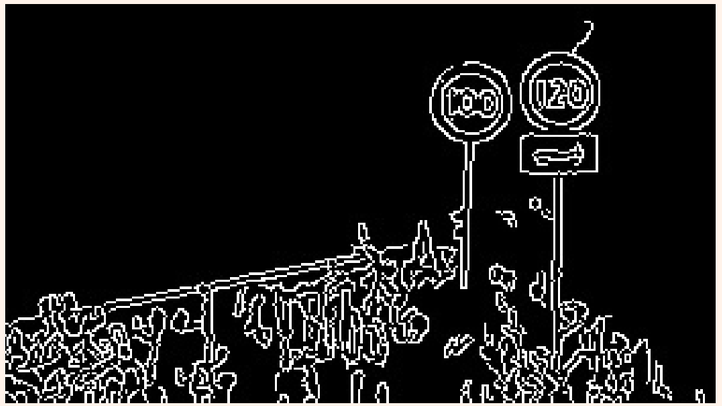
Input Image

Enhanced Image

Final Output

Blurred Image

Edges Detected

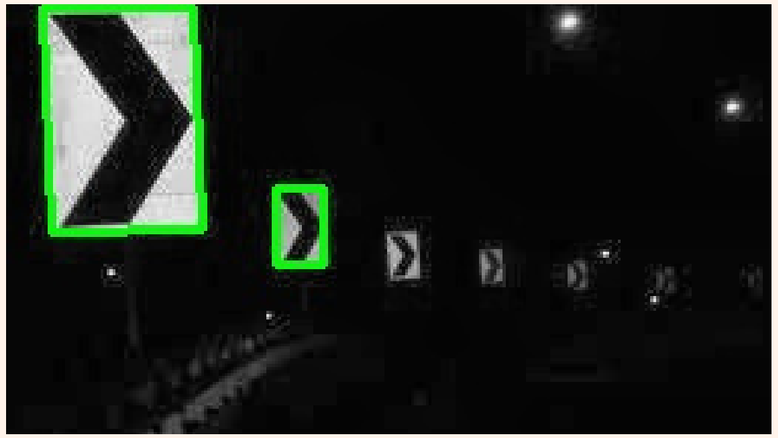


RESULTS CONTINUING

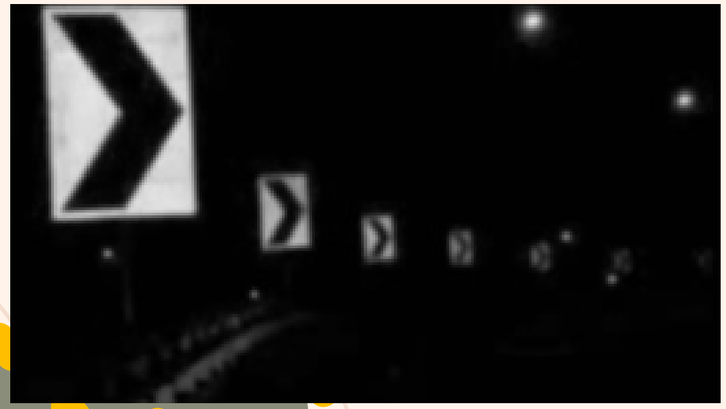
Input Image

Enhanced Image

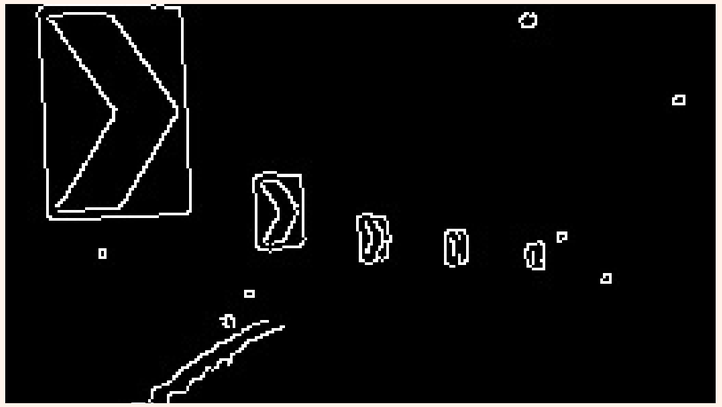
Final Output



Blurred Image



Edges Detected



THANK YOU