Image Processing and Computer Vision Laboratory

CSE 4128

Project Title: **Stair Counter**

Supervised By:

Dr. Sk. Md. Masudul Ahsan
Professor
Department of CSE, KUET

Dipannita Biswas
Lecturer
Department of CSE, KUET

Submitted By:

Md. Rakibul Hasan Adnan
Roll: 1907106
Year: 4th
Term: 1st
Date of Submission: 02-09-2024

Outline

- Introduction
- Application
- Key Steps
- Methodology Explanation
- Result Analysis
- Conclusion
- References

Introduction

- Stair counter project involves developing a python program that automatically detects and counts stairs in images.

- This project uses image processing techniques and custom algorithms to analyze images for stair structures.

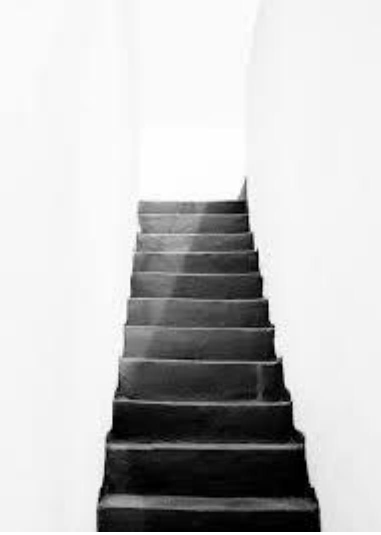


Figure 1 : Stair Image

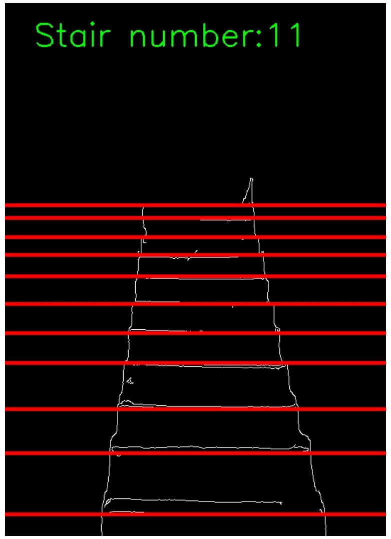


Figure 2 : Stair counted

Application

- Architectural Analysis
- Safety Assessments
- Autonomous Navigation

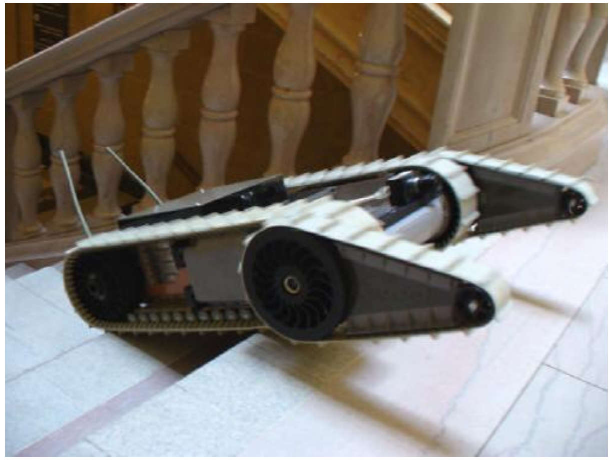


Figure 3 : Autonomous Robot climbing Stairs

Key Steps

1. Load and Preprocess Image
2. Edge Detection
3. Line Detection
4. Count Stairs

Methodology Explanation

- **Image Preprocessing**
  1. Load the Image and resize if necessary to standardize dimensions
  2. Convert the image to grayscale , simplifying it for further processing

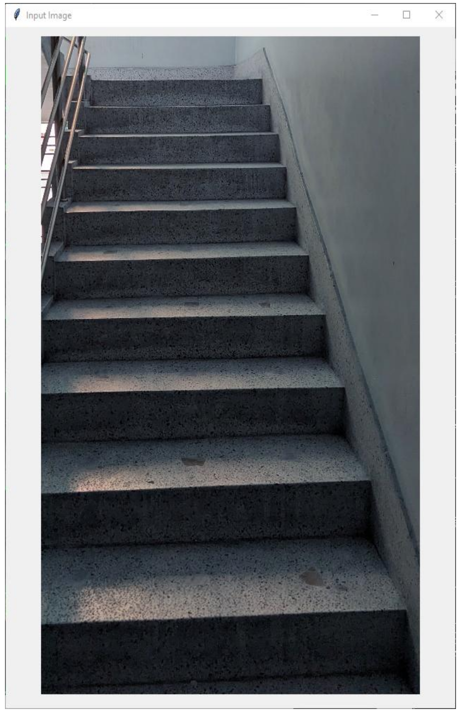


Figure 4: Input Image

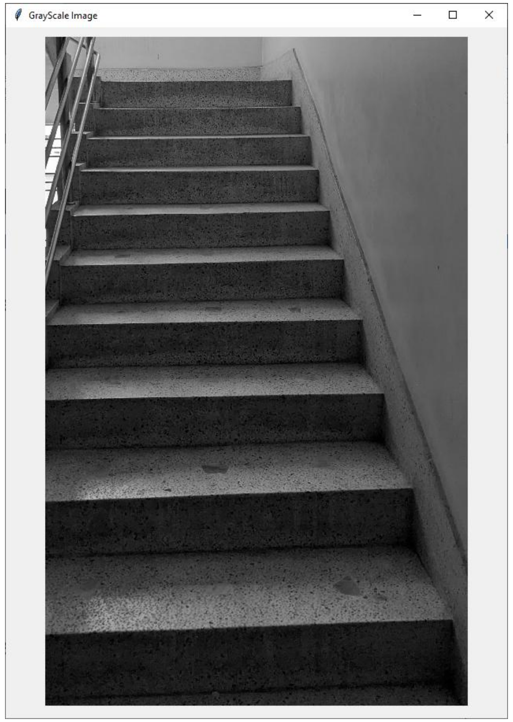


Figure 5: Image Converted to Grayscale

Methodology Explanation continued..

- **Edge Detection**
  1. Apply a custom Canny Edge Detection Algorithm
     - Gaussian Blur
     - Gradient Calculation
     - Non-Maximum Suppression
     - Double Threshold & Hysteresiss

Figure 6 : Input Image

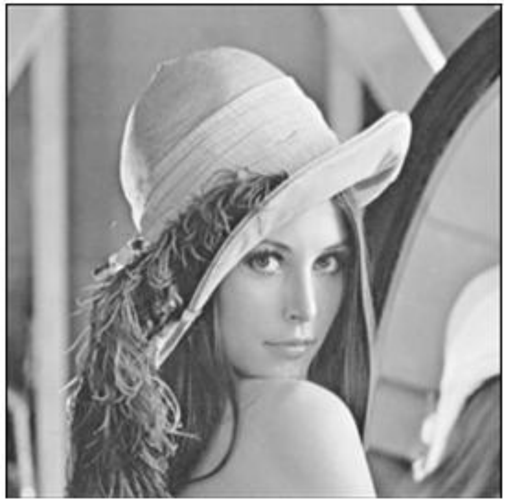


Figure 7 : Gaussian Blur Output

Methodology Explanation continued..

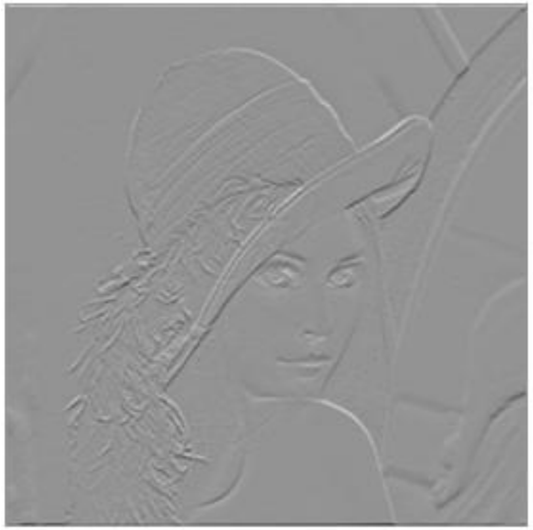


Figure 8: X derivative

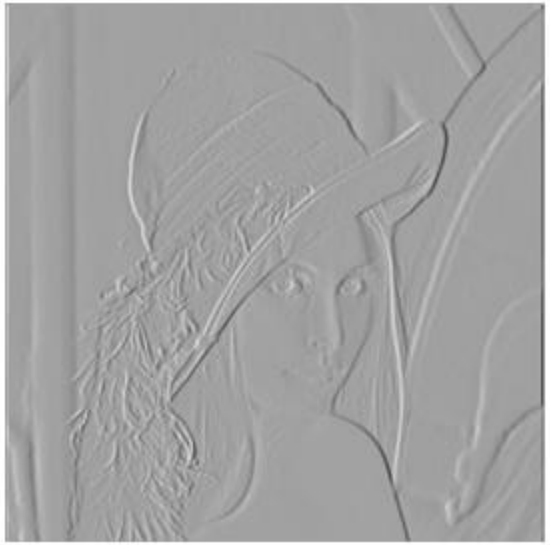


Figure 9 : y derivative

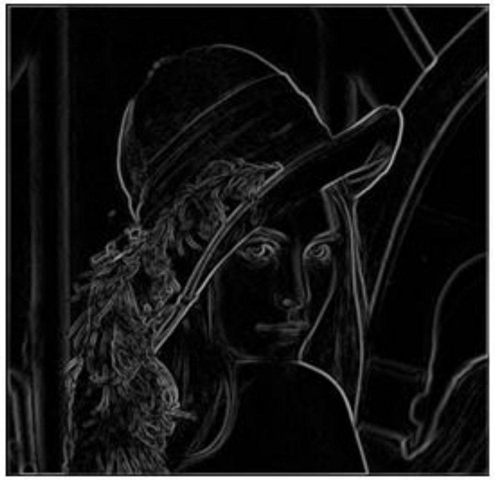


Figure 10 : Gradient Magnitude

Methodology Explanation continued..

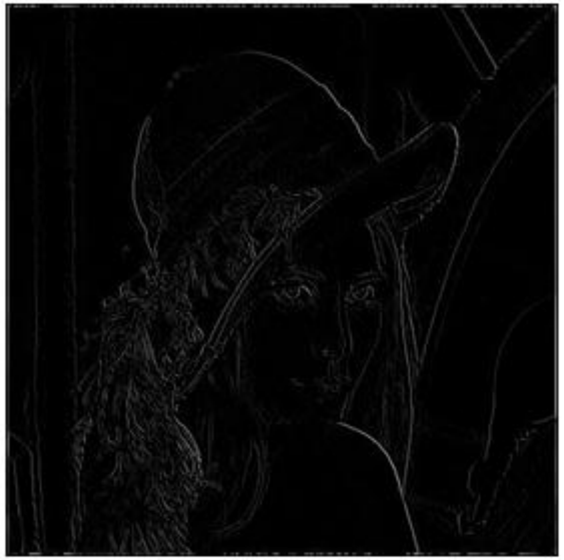


Figure 11: NMS suppression

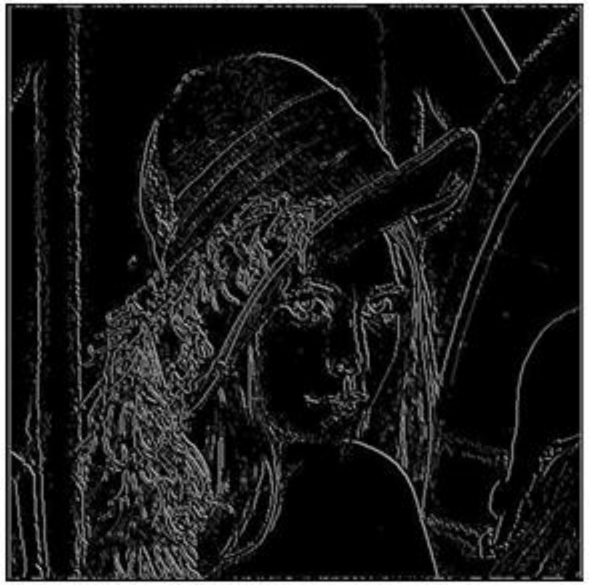


Figure 12 : Double Thresholding

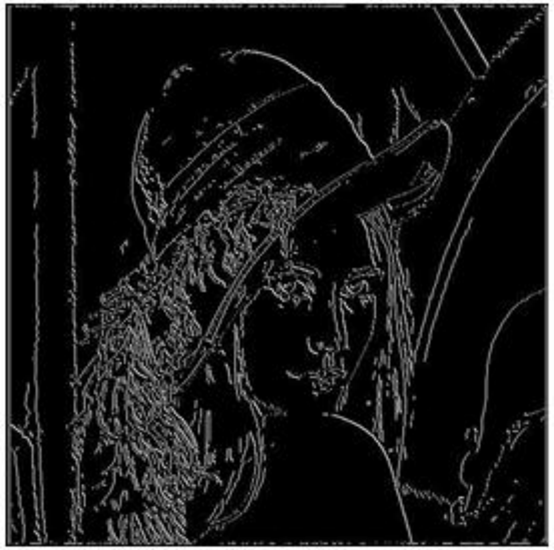


Figure 13 : Hysteresis

Methodology Explanation continued..

- **Line Detection**
  1. Hough Line Transform to detect straight lines from the edge detected image.
  2. Filter lines based on their orientation and distance to identify potential stair edges

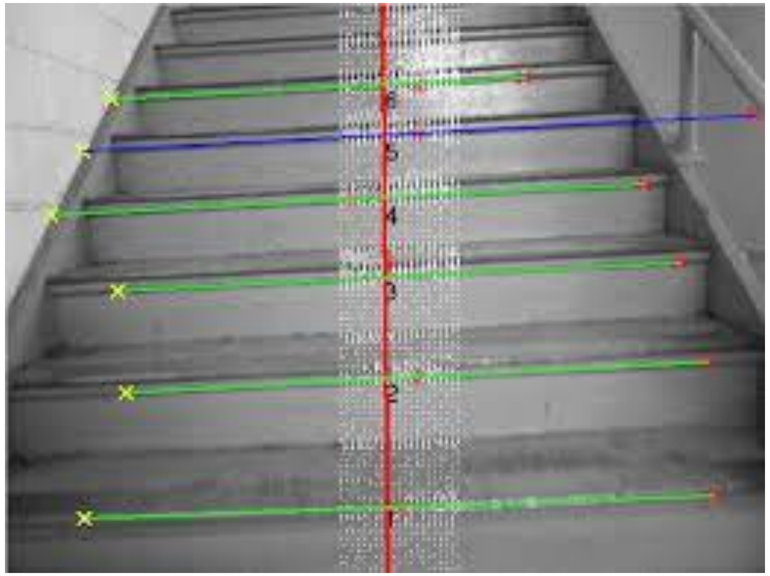
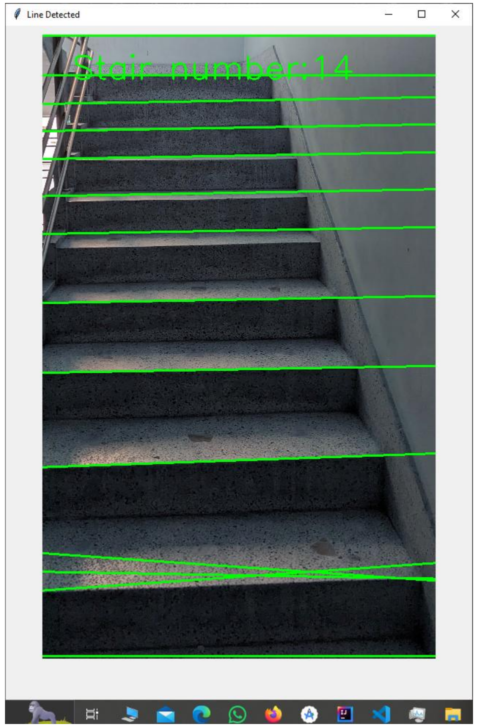


Figure 14 : Detection of lines from a stair image

Methodology Explanation continued..

- **Counting Stairs**
  1. Analyze the detected lines to determine the number of stairs:
     - Count lines that meet criteria for stair edges
     - Ensure lines are parallel and consistently spaced to avoid false positives



Stair number:14

Figure 15 : Stair Number Counted

Result

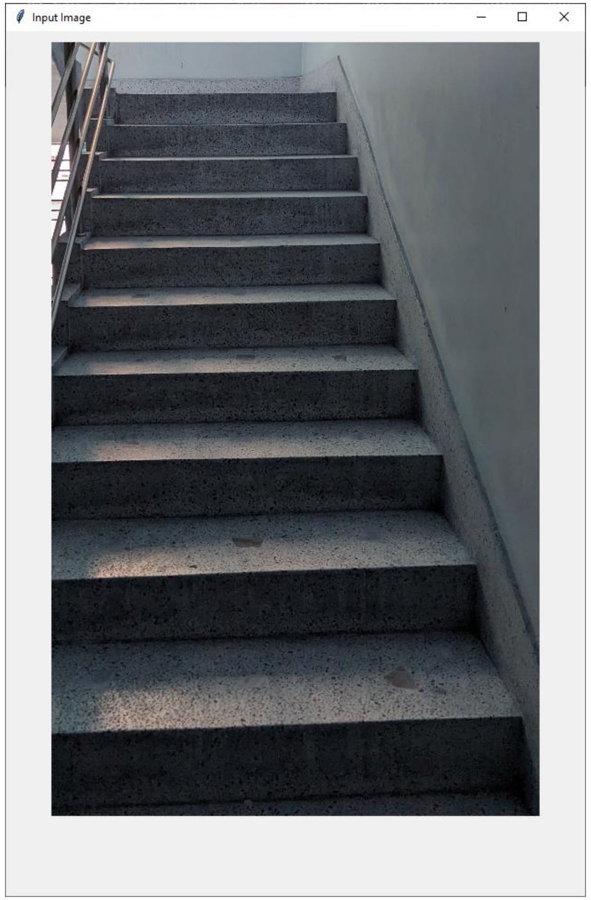


Figure 16: Input Image

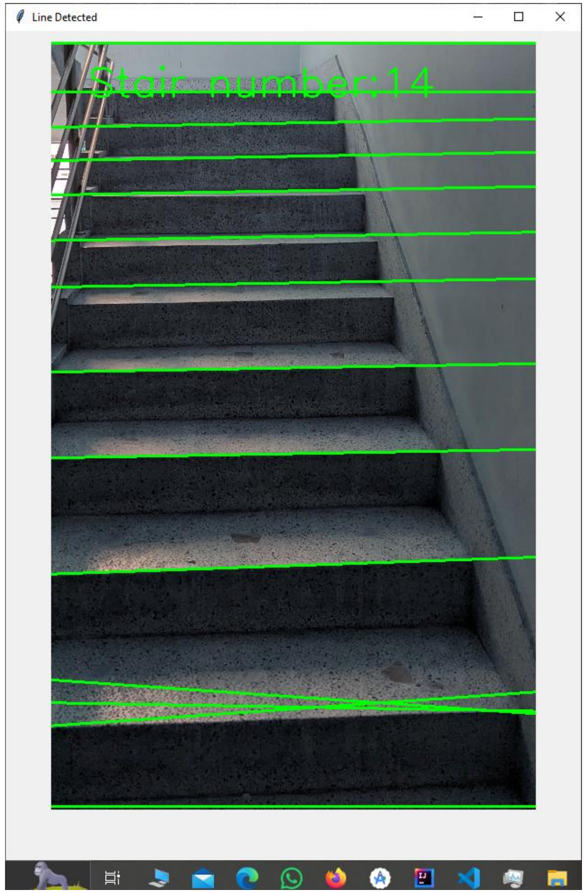


Figure 17: Result

Limitations

- The algorithm may have difficulty with images containing complex staircases or unusual angles.
- The presence of clutter or occlusion in images may result in missed of false stair detections.
- Further refinement and integration with machine learning techniques could improve accuracy

Conclusion

- The Stair Counter project demonstrates the feasibility of using computer vision techniques for stair detection.
- The project achieves its goal, with the algorithm detecting stairs accurately in most cases.
- Future work could involve enhancing the algorithm with machine learning to improve robustness and adaptability

References

- Canny Edge Detection – https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

- Hough Line Transform - https://medium.com/@alb.formaggio/implementing-the-hough-transform-from-scratch-09a56ba7316b