

Title: Sobel Edge Detection in Python

Author: Martin Brettschneider

Institution: Radboud University

Course: Programming I

Date: 15.10.2025

Contact: martin.brettschneider@ru.nl

ResearchGate: https://www.researchgate.net/profile/Martin-Brettschneider-2?ev=hdr_xprf

This project implements a simple edge detection program using the Sobel operator, a classic image processing technique used to highlight areas of rapid intensity change, i.e., edges.

The user provides an image (in .jpg or .png format), chooses a threshold value, and the program detects edges, visualizes them, and saves the result to a text file.

What it does:

1. Reads and converts an image into grayscale.
2. Applies horizontal and vertical Sobel kernels to detect changes in intensity along both directions.
3. Combines both gradients to compute the edge magnitude.
4. Applies a threshold (chosen by the user) to produce a binary edge map.
5. Displays the resulting edges visually.
6. Saves the binary edge map to a text file.
7. Calculates and displays basic statistics (mean, min, max) for the detected edges.

How to use it:

Run the program from the terminal:

```
python main.py path/to/your/image.jpg
```

Then follow the prompts:

- Enter a threshold value between 0 and 1.
- View the resulting edge-detected image.
- Enter a name to save the output file (saved inside the output/ folder).

Example Uses of Each Project Requirement

Requirement	Example in Project
Read/write from files	<code>read_file()</code> reads an image file; <code>save_to_file()</code> writes the edge map to a .txt file in the output/ folder.
User input and output	<code>get_user_threshold()</code> and <code>input("Enter output filename...")</code> gather input from the user; <code>print_usage()</code> and <code>print()</code> provide output feedback.
Use of functions	Every major task (reading, processing, thresholding, saving) is encapsulated in its own function (<code>sobel_operator()</code> , <code>threshold_edges()</code> , etc.).

Loops and conditionals	The nested for loops in <code>sobel_operator()</code> perform the convolution; if conditions validate inputs in <code>check_args()</code> and user threshold in <code>get_user_threshold()</code> .
Lists and indexing	The <code>analyze_edges()</code> function stores statistics in a list of tuples, demonstrates indexing (<code>stats[0]</code>), and manipulation (<code>stats.pop()</code>).
Modules/libraries	Uses <code>numpy</code> for matrix operations, <code>matplotlib</code> for reading and displaying images, and <code>os/sys</code> for file and command-line handling.
Exception handling	The <code>main()</code> function uses a try-except block to catch and handle runtime errors gracefully.
Docstrings and comments	Every function includes a detailed docstring describing its purpose, inputs, and outputs.

Acknowledgements and References

- Conceptual references:
 - Fisher R, Perkins S, Walker A, Wolfart E (1996) Hypermedia image processing reference. John Wiley & Sons Ltd, England, pp 183–202
- Code references:
 - The implementation of Sobel convolution and edge magnitude calculation was written entirely by Martin Brettschneider (me), inspired by standard textbook methods.

Authorship Statement

I declare that the work submitted here is from my authorship only. I haven't used any generative AI to help with any code/text included in my work. I have given credit for the help I had conceptualizing my project. My work respects the university and course code of conduct