

# Parallel Image Processing (Edge Detection)

Deniz Can Calkin



# Introduction to Image Processing and Edge Detection

- Image Processing: The use of algorithms to perform operations on images for enhancement or extraction of information.
- Edge Detection: A fundamental image processing task that identifies significant transitions in intensity in an image.
- Sobel Filter: A widely used edge detection algorithm that computes the gradient of image intensity.





# Project Overview

## **Objective:**

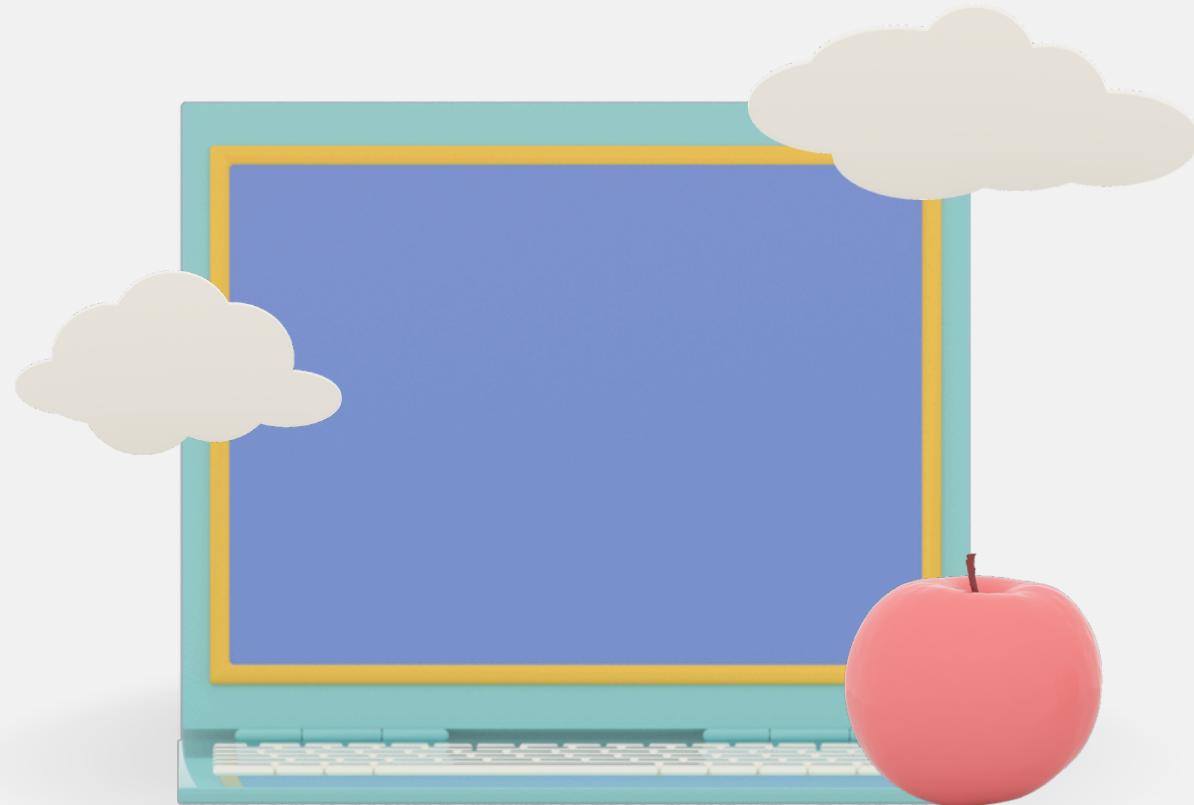
Implement a parallelized version of the Sobel filter for edge detection using OpenCV and OpenMP.

## **Goals:**

Leverage parallel computing to enhance performance.

Compare results visually to ensure the correctness of the parallel implementation.

# Tools and Libraries



## OpenCV:

- A robust library for computer vision and image processing tasks.
- Provides functions for reading, displaying, and processing images.

## OpenMP:

- An API for parallel programming in C/C++.
- Simplifies the implementation of multi-threading to improve performance.

# Image Preprocessing

## **Loading the Image:**

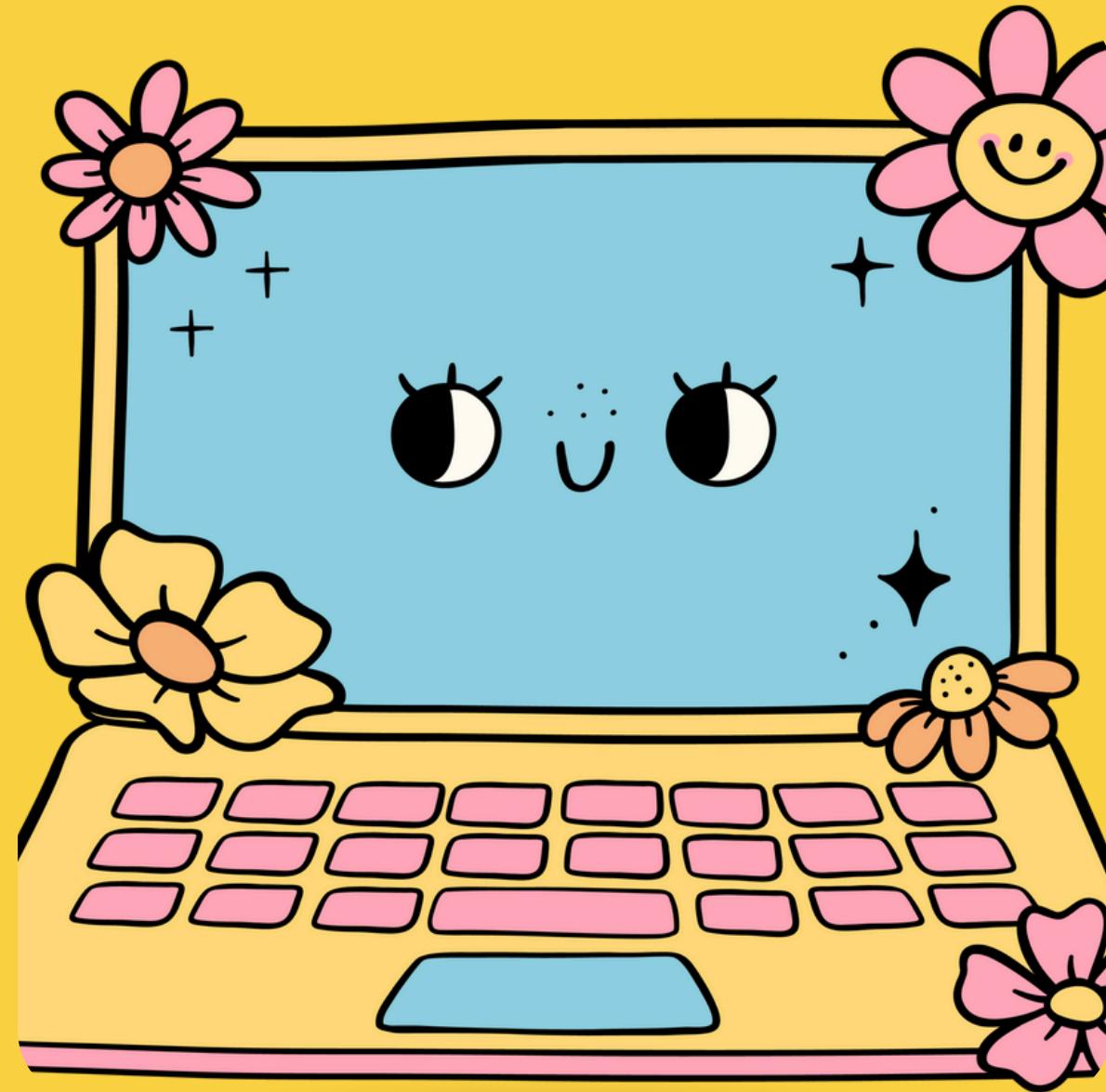
- Utilizes OpenCV's imread function to read the image from a file.
- Validates image loading to prevent errors during processing.

## **Grayscale Conversion:**

- Converts the original image to grayscale to simplify processing and reduce computational complexity.

## **Gaussian Blur:**

- Applies a Gaussian filter to smooth the image and reduce noise.
- Helps in reducing false edge detection in the subsequent Sobel operation.



# Parallel Sobel Filter Implementation



## **Objective:**

Apply the Sobel filter in parallel using OpenMP to enhance computational efficiency.

## **Gradient Calculation:**

Computes horizontal ( $G_x$ ) and vertical ( $G_y$ ) gradients using Sobel operators.

## **Magnitude Calculation:**

Computes the magnitude of the gradient vector to detect edges.

## **Parallel Execution:**

Employs OpenMP to divide the computation across multiple threads.

Dynamic scheduling ensures efficient workload distribution.



# Image Display and Comparison

## **Displaying Images:**

Utilizes OpenCV's imshow function to display the original and processed images.

Resizes images to fixed dimensions for consistent viewing.

## **Comparison:**

Visual inspection of the processed image to verify the accuracy of edge detection.

Ensures that parallelization does not affect the correctness of the output.

# Performance Benefits

## **Parallelization Advantages:**

Significantly reduces the computation time by utilizing multiple CPU cores.

Makes real-time image processing feasible for large images or video streams.

## **Scalability:**

The implementation can be scaled to exploit more cores as available, enhancing performance on multi-core systems.



# Conclusion

- Successfully implemented a parallel Sobel filter for edge detection.
- Demonstrated the benefits of using OpenMP for computational efficiency.



THANK YOU  
FOR LISTENING