Self Introduction and Image Processing Demonstration

Hari Varshan Dharmendra Mohan Prabu
Arizona State University
hdharmen@asu.edu

Abstract

This document introduces myself and demonstrates the use of basic image processing techniques in Python. I present an original photograph and its processed version and describe the steps used to enhance the image. I also provide background on my academic journey, motivations for enrolling in CSE 507, and future career objectives.

1. Original and Processed Image

Figure 1 shows my original photograph, and Figure 2 shows the processed version created using sharpening, Sobel edge detection, and background blurring.

Figure 1. Original photo.

Figure 2. Processed output: Sobel edge map after thresholding (0.1). Dilation was invoked with k=1 (no effective thickening).

2. Image Processing Steps (As seen in Code)

My goal was to extract the outlines of the subject from my original photo. Below is a step-by-step explanation of the operations I performed in my code. Let $I \in [0,1]^{H\times W\times 3}$ denote the RGB image.

(1) Resize & Horizontal Flip. I resize to (700×500) using bilinear interpolation and apply a horizontal flip (I used `flip='h'`) so that it has a consistent and manageable size for processing. Flipping was an added touch to mirror the photo left-to-right. These operations only adjust the image geometry for convenience.

(2) Unsharp Masking (Sharpen). Raw images often

have soft or blurred edges. To make edges and details stand out, I applied a technique called *unsharp masking*. First, a blurred (smoothed) version of the image is created using a Gaussian blur filter, which averages nearby pixels and removes fine detail. Then, the blurred image is subtracted from the original image to extract just the fine details (the differences). Finally, those extracted details are added back to the original image with a scaling factor ($\alpha = 2.0$ in my code). Gaussian blur $G_\sigma * I$ (kernel size 5, σ=1) :

$$I_{\text{sharp}} \;=\; I \;+\; \alpha\,\big(I - G_\sigma * I\big),$$

with α=2.0. This boosts edges and fine detail.

(3) Grayscale for Gradients. Color images have three channels (red, green, blue). For edge detection, we only need intensity, not color. I convert the sharpened image to grayscale by taking a weighted sum of the three color channels:

$$I_{\text{gray}} \;=\; 0.299\,R + 0.587\,G + 0.114\,B.$$

The result is a single-channel image where bright areas represent higher intensity.

(4) Sobel Gradient Magnitude. Edges are places where the image intensity changes sharply, like object boundaries or outlines. To detect them, I used the Sobel operator, which is a simple mathematical filter. It is computed via 3×3 Sobel kernels over the grayscale image: one measures horizontal changes (G_x), the other measures vertical changes (G_y). The amount of change at each pixel is calculated as:

$$G_x = I_{\text{gray}} * K_x, \quad G_y = I_{\text{gray}} * K_y, \quad M = \sqrt{G_x^2 + G_y^2 + \varepsilon}.$$

The magnitude M is called the gradient magnitude, which is normalized to [0,1]. Intuitively, this gives us a "strength of edge" map: high values where there are strong edges, and low values where the image is smooth.

(5) Normalize and Threshold. The gradient values are scaled (normalized) so they all fall between 0 and 1 for consistency. Then, I apply a threshold: any pixel with an edge strength above 0.1 is considered an edge (set to white), and the rest are considered non-edges (set to black). This turns the smooth gradient map into a binary image with clear black-and-white edges.

(6) "Dilation" (edge cleanup). In image processing, dilation expands or thickens white regions. Although my code calls a dilation function, I used a $k = 1$ (1x1 window), which effectively does nothing. In future work, using $k = 3$ or $k = 5$ could help connect broken edges and make outlines more continuous.

(7) Save. The binary map E is saved as an 8-bit PNG (values scaled from $[0, 1]$ to $[0, 255]$). No edge overlay on color and no background blur are performed in this version, so the processed image is strictly the edge map.

By following these steps, the original photo is transformed into a sketch-like image showing only its prominent structural lines. This pipeline uses basic and interpretable operations, making it a good starting point for learning about edge detection and preparing for more advanced tasks like neural network-based image analysis.

3. Background

I am currently pursuing a Ph.D. in Computer Science at Arizona State University with a primary focus on applying medical imaging, text, and sequencing data to Deep leanng models. My skills include extended reality development (VR/AR), machine learning, and software engineering, including cloud computing and full-stack development. I have prior experience working with multi-omic data and am familiar with the theoretical concepts behind image analysis such as convolution, filtering, and segmentation, and how these combined with deep learning models can be applied to medicine and data science. However, I have not had much hands-on experience directly processing images or training deep neural networks, which is why I am taking this course.

4. Motivation for Taking CSE 507

As part of my Ph.D. research, I am exploring how multimodal data can improve prediction and prognosis accuracy for diseases such as lung adenocarcinoma (LUAD) and Alzheimer's disease (AD). I am particularly interested in understanding the impact of combining health records, sequencing, and imaging data on predictive models for these diseases. To gain deeper insight, I enrolled in this class to strengthen my foundations in image processing and analysis. With the expectations set clearly on the first day, I look forward to immersing myself in the opportunities this course provides, applying its lessons to my research, and transferring the knowledge gained here directly into my Ph.D. work.

5. Career Objectives

My long-term goal is to develop AI-driven tools that integrate multimodal biomedical imaging with clinical data to improve early disease detection and personalized healthcare. I want to contribute to the field of science and technology in this area by combining advances in machine learning with medical knowledge to create solutions that have real-world impact for patients and clinicians.

Code Repository

The full code used for image processing in this assignment is available Here