Prime Image Studio - Ultimate Image Enhancement
Team_9

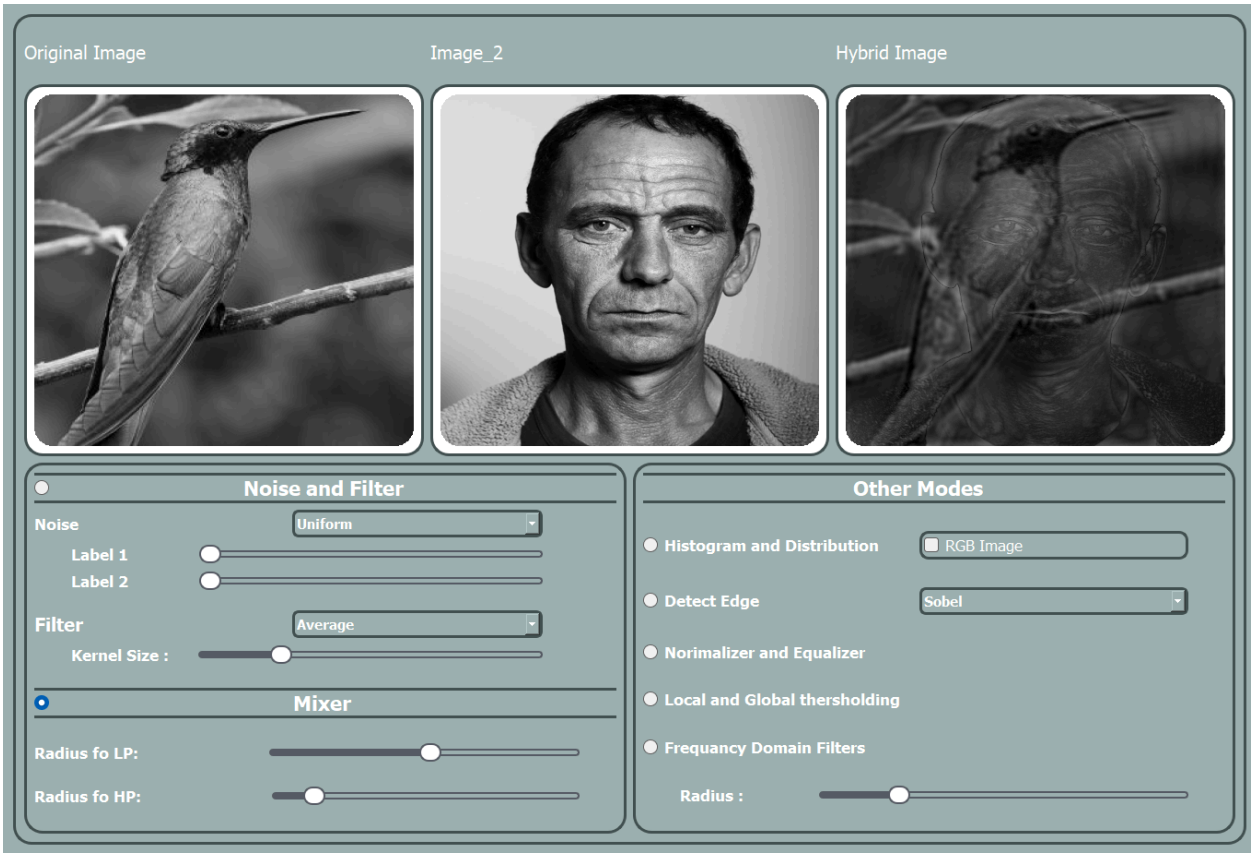
Supervised by: Dr. Ahmed Badawy
Submitted to: Eng. Yara Wael & Eng. Omar Muhammed

Name	Sec & BN
Muahmmed Salah	2 & 56
Marcilino Adel	
Nada Khaled	
Shaimaa kamel	1 & 32

Introduction

This project is an image processing application developed using PyQt5 in Python and C++. The application provides various functionalities for image manipulation, including adding noise, filtering, edge detection, histogram analysis, equalization, normalization, thresholding, frequency domain filtering, and creating hybrid images.

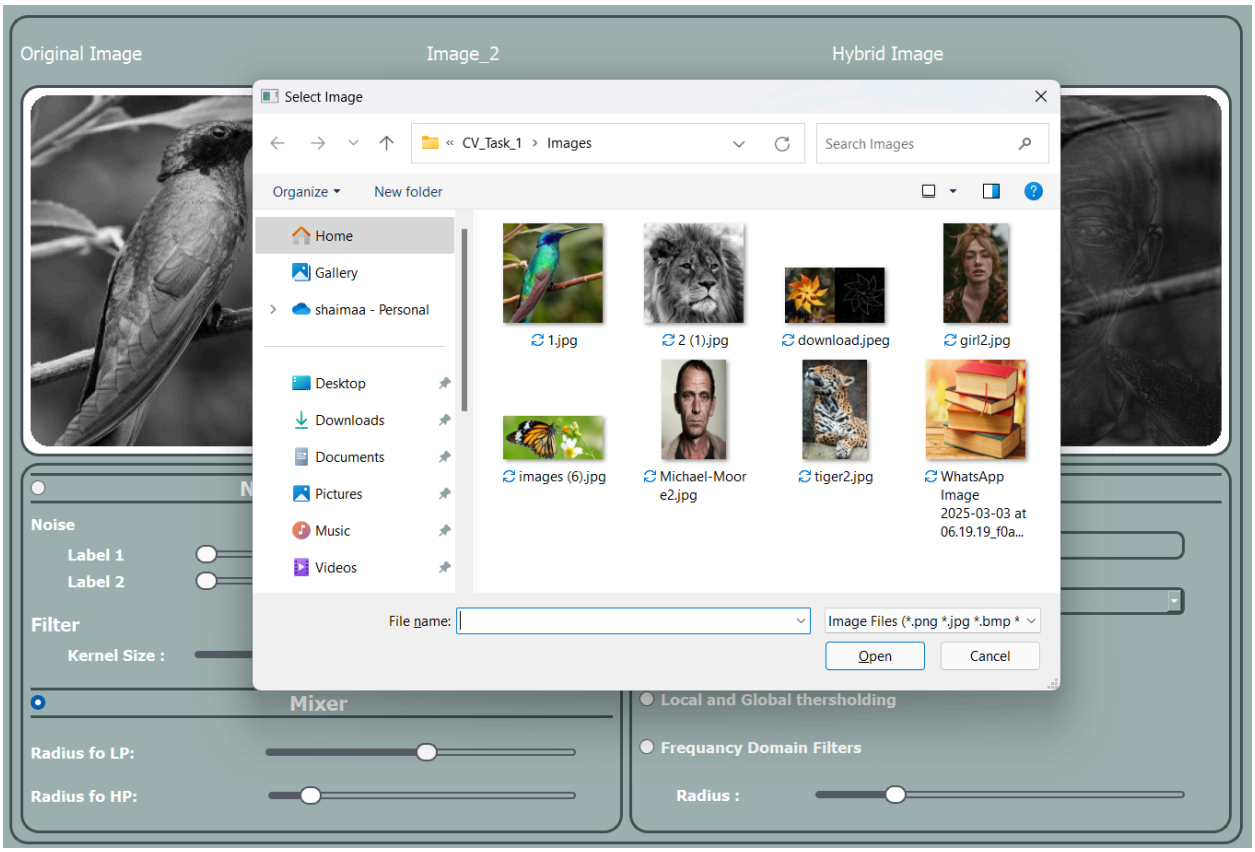
UI :



Features:

1.Load Image :

We have a class called Image_Widget, designed to read the image using OpenCV, resize it, and finally display it within the application.
The image is uploaded when the user double-clicks on the widget.



2.Modes:

2.1 .Noise and Filter Mode:

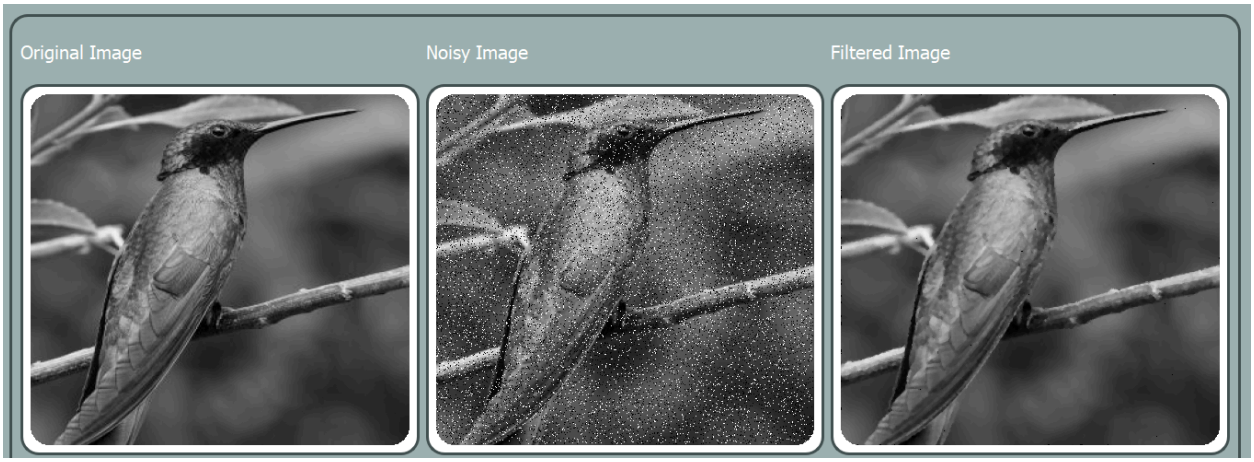
Digital images often suffer from noise due to environmental factors, transmission errors, or hardware limitations. Common types of noise and their input parameters include:

- Uniform Noise: Intensity variations are evenly distributed, controlled by min and max values.
- Gaussian Noise: Follows a bell-shaped distribution, defined by mean and variance parameters.
- Salt & Pepper Noise: Characterized by random bright (salt) and dark (pepper) pixels, controlled by a percentage parameter.

To reduce noise while preserving important details, different filtering techniques are applied, with kernel size as a key parameter:

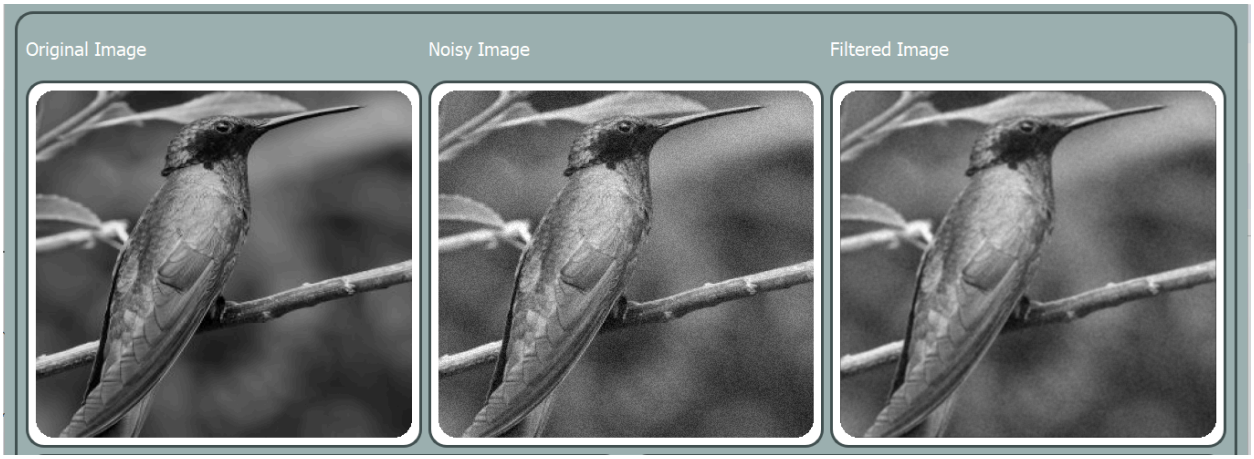
- Average Filter: Replaces each pixel with the average intensity of its neighborhood, smoothing the image (kernel size: 3×3, 5×5, etc.).
- Gaussian Filter: Applies a weighted average using a Gaussian distribution, where a larger kernel increases smoothing.
- Median Filter: Replaces each pixel with the median of its neighbors, effectively removing impulse noise, with kernel size affecting noise reduction efficiency.

Salt & Pepper Noise and Median filter

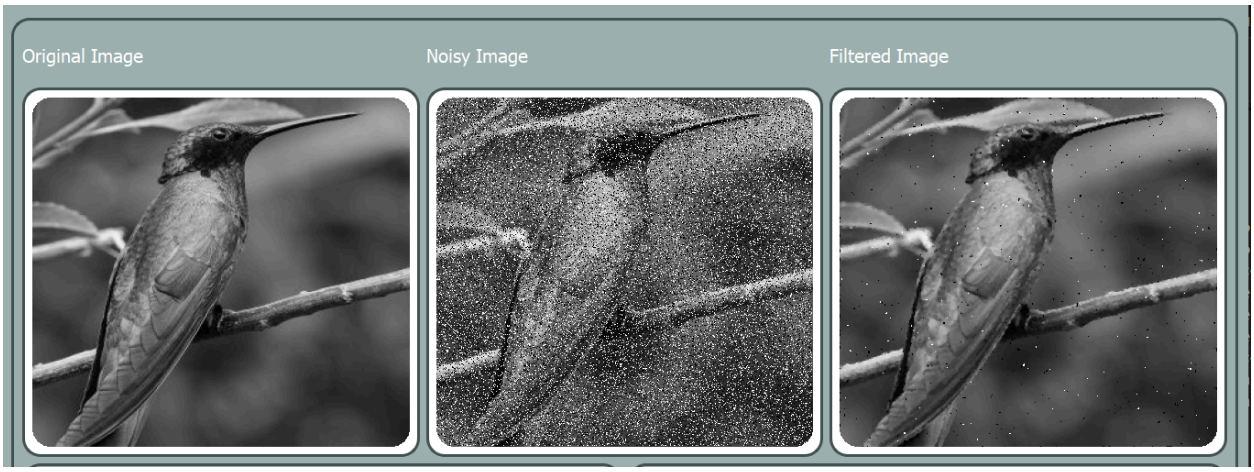


Gaussian Noise and Gaussian filter

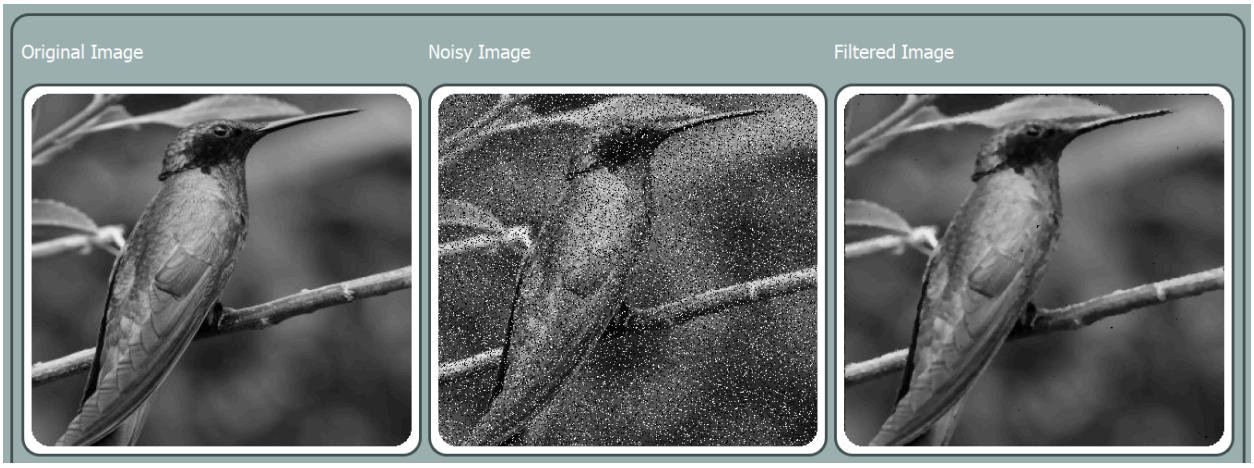
Uniform Noise and Average filter



Median filter & Ksize=3

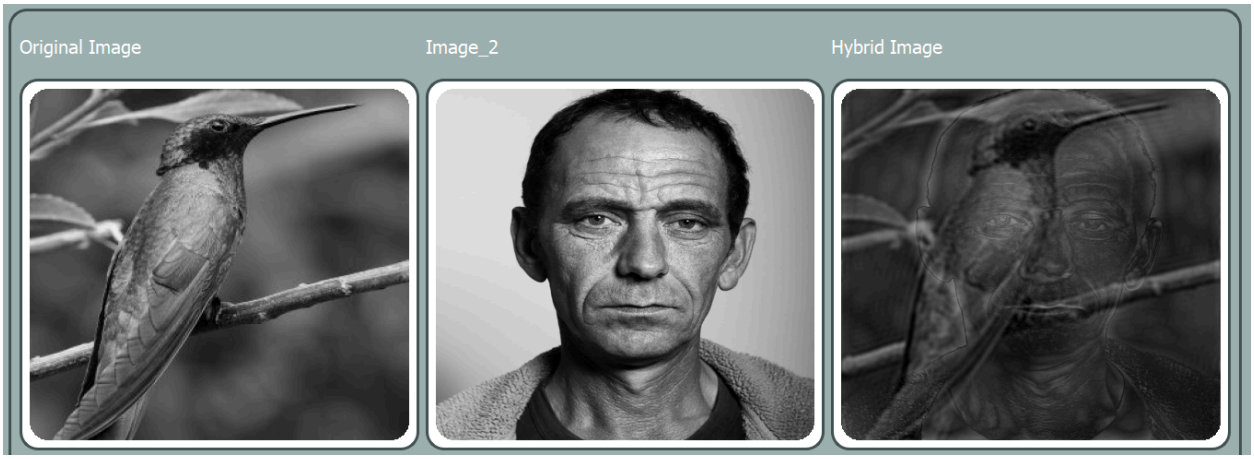


Median filter & Ksize=5



2.2 Mixer Mode :

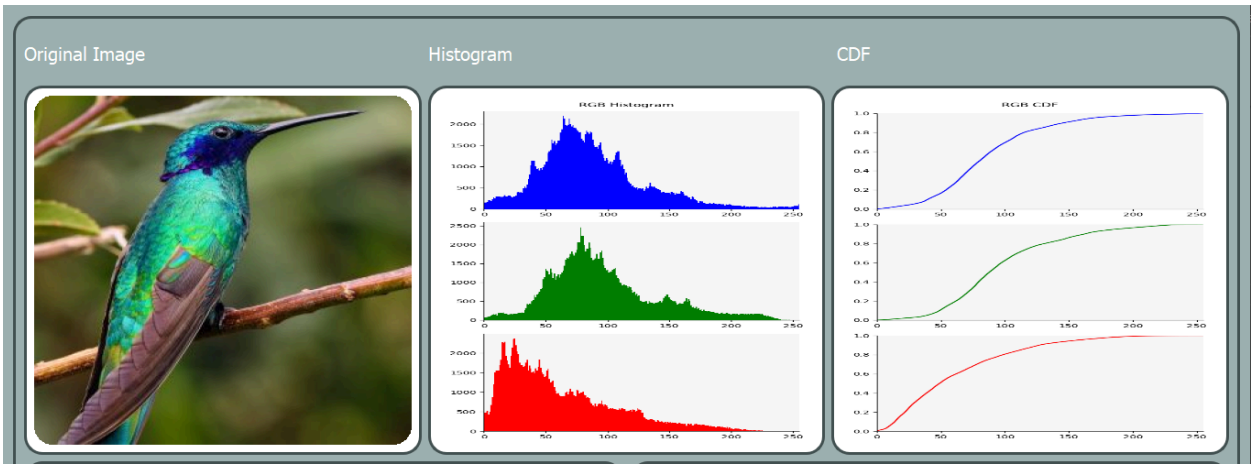
Hybrid images blend the low-frequency content of one image with the high-frequency content of another. A low-pass filter (mask radius controlled) extracts low frequencies from one image, while a high-pass filter extracts high frequencies from another. The filtered components are combined.

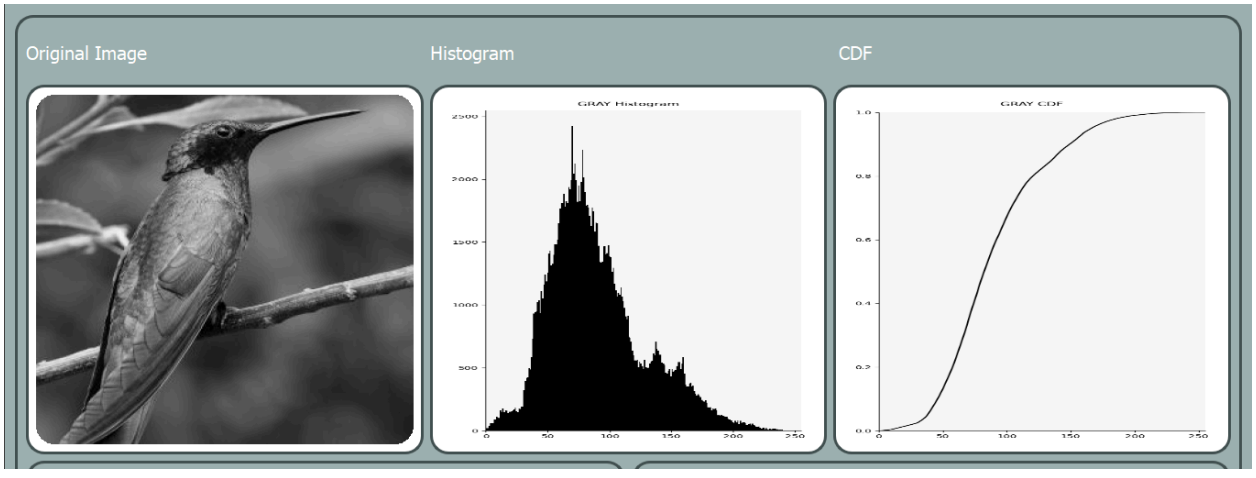


2.3 Histogram and CDF Mode :

An image histogram graphically represents pixel intensity distribution, dividing intensities into bins that count pixel frequency. Distribution curves can be overlaid to show intensity spread, aiding in image enhancement by revealing brightness, contrast, and tonal balance.

- In grayscale mode, the histogram covers intensity values from 0 (black) to 255 (white).
- In RGB mode, the histogram consists of three separate histograms — one for each color channel: Red, Green, and Blue.





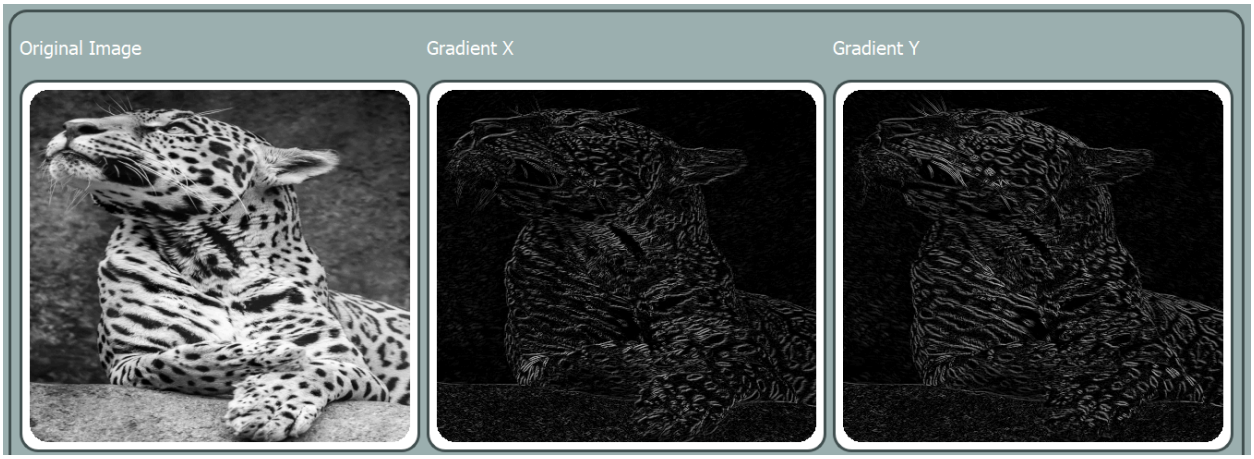
2.4 Detect Edge Mode :

2.4.1 Sobel Edge Detection:

Is a common technique for detecting edges in digital images. It works by convolving the image with 3x3 kernels to compute the gradient magnitude (rate of intensity change) and gradient direction (steepest ascent). Applying a threshold to the gradient magnitude effectively identifies edges.

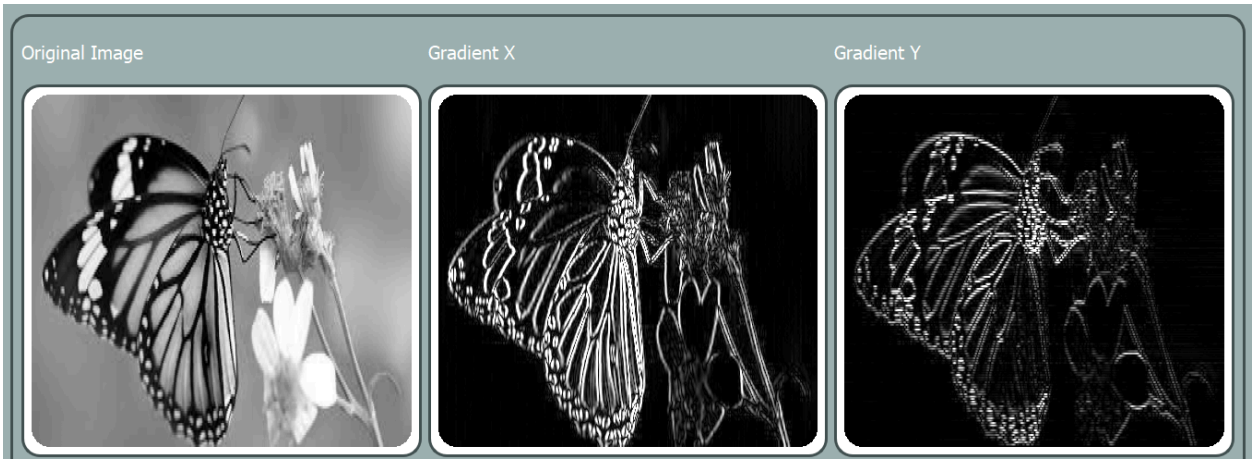
2.4.2 Roberts Edge Detection:

Is a simple and efficient technique that uses 2x2 kernels to approximate gradients in horizontal and vertical directions. The gradient magnitude is calculated by combining the absolute gradients. While fast, it often provides less accurate results than more advanced methods.



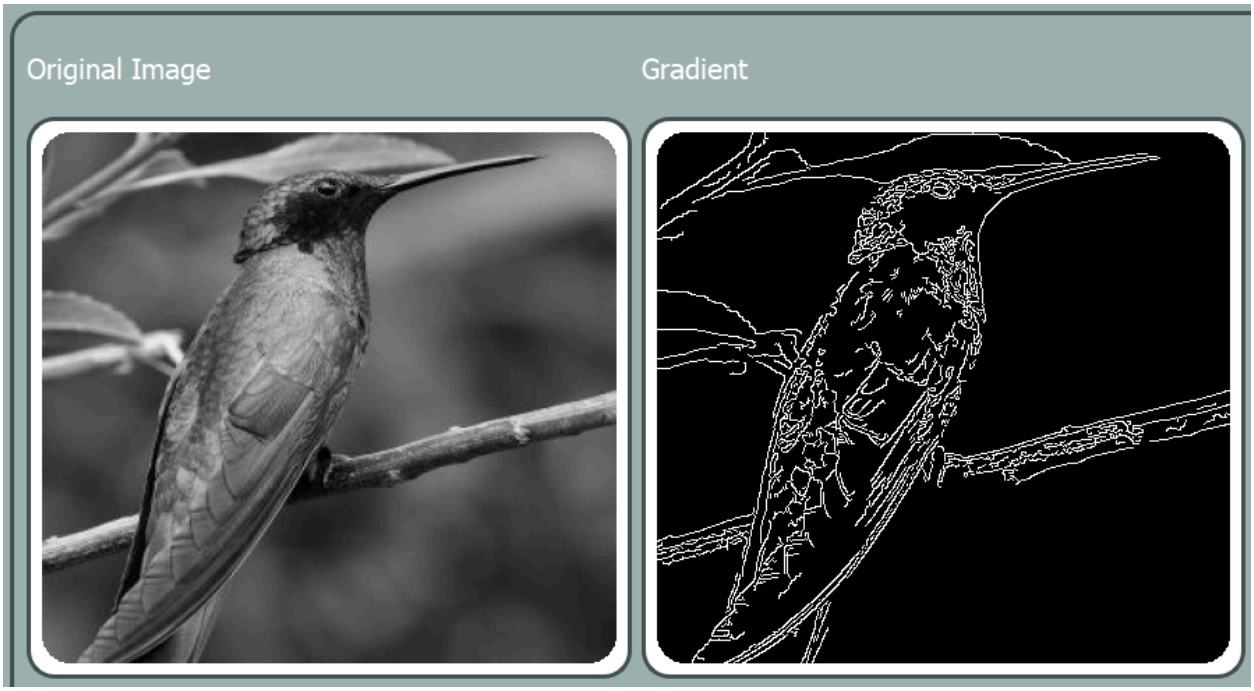
2.4.3 Prewitt Edge Detection:

Like Sobel, it uses 3x3 kernels to compute gradient magnitude and direction. However, Prewitt kernels are better at detecting diagonal edges, making it more robust to diagonal features than Sobel.



2.4.4 Canny Edge Detection:

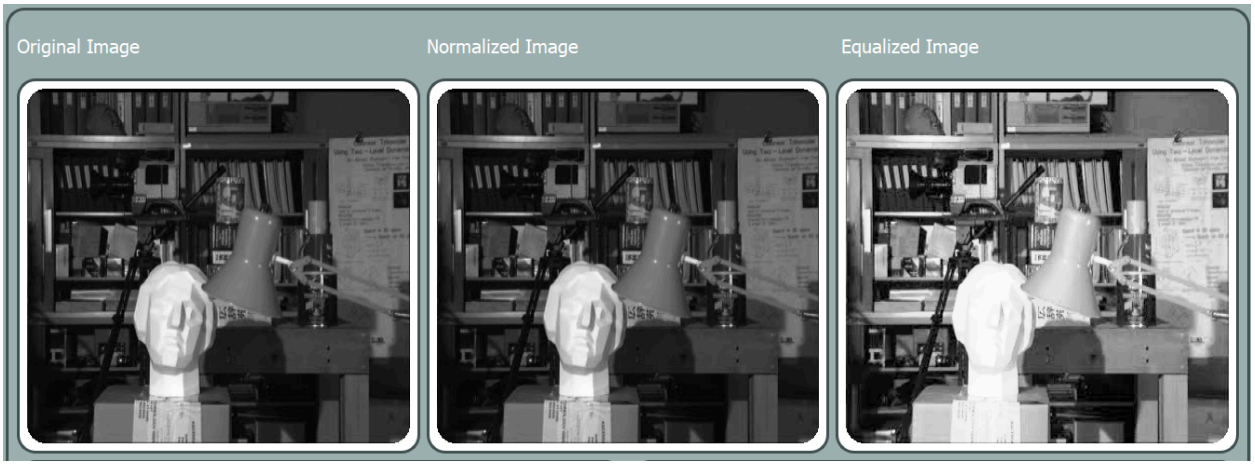
is a multi-stage process designed for accurate edge detection with minimal noise and false positives. It applies Gaussian blurring, gradient calculation (using Sobel or Prewitt), non-maximum suppression to thin edges, and hysteresis thresholding to link edges. It's widely used for its high accuracy and reliability

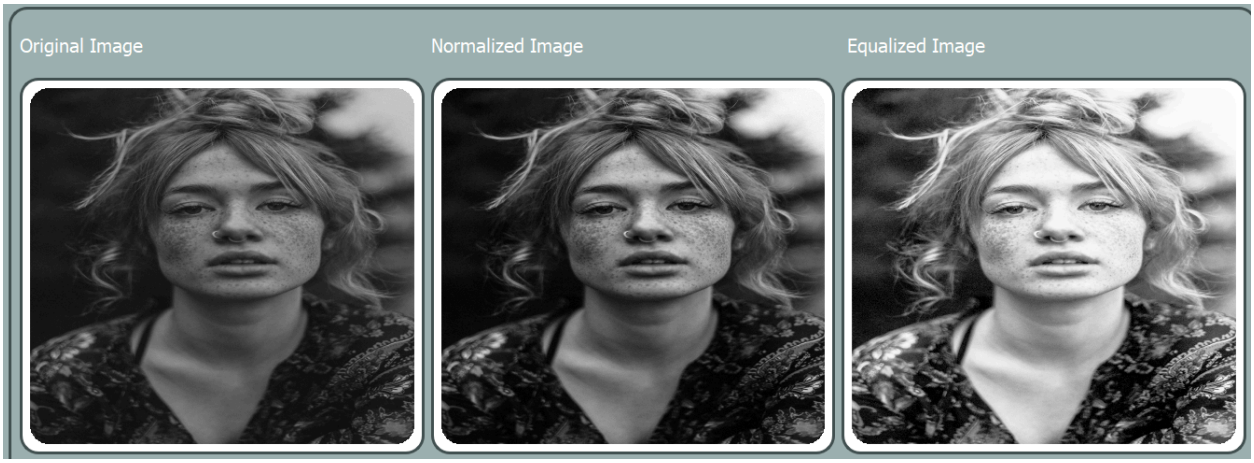


2.3 Normalization and Equalization Mode:

Normalization scales image pixel values to a specified range, such as [0, 255] for 8-bit images, to enhance visualization and analysis. It works by finding the minimum and maximum pixel intensities, then scaling all pixels proportionally to fit within the target range, producing a normalized image with adjusted intensities.

Histogram equalization enhances image contrast by redistributing pixel intensities to produce a more uniform histogram. The process involves computing the image histogram, calculating its cumulative distribution function (CDF), normalizing the CDF to the target range (e.g., [0, 255]), and mapping original pixel values to their new equalized values, resulting in an image with improved contrast.

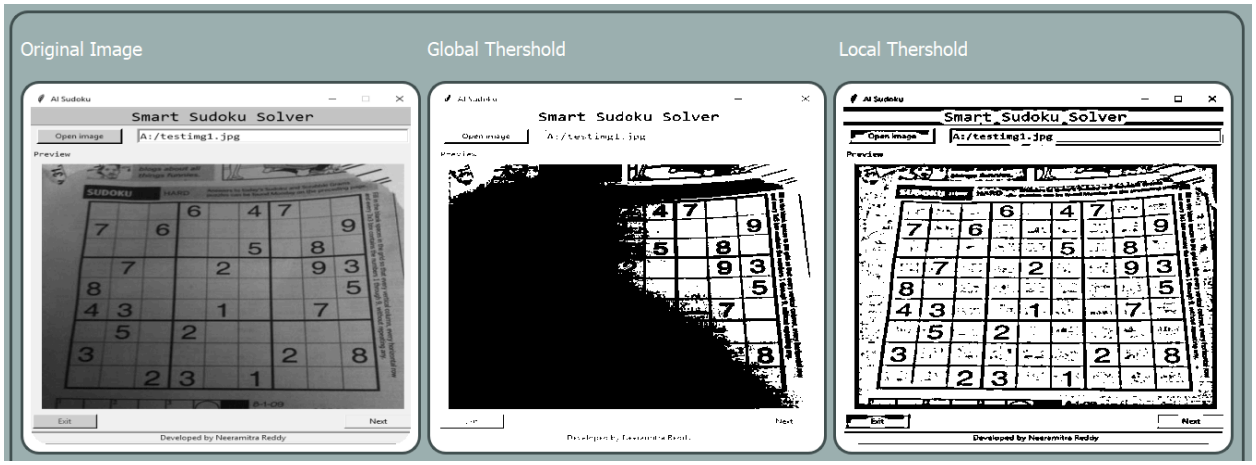


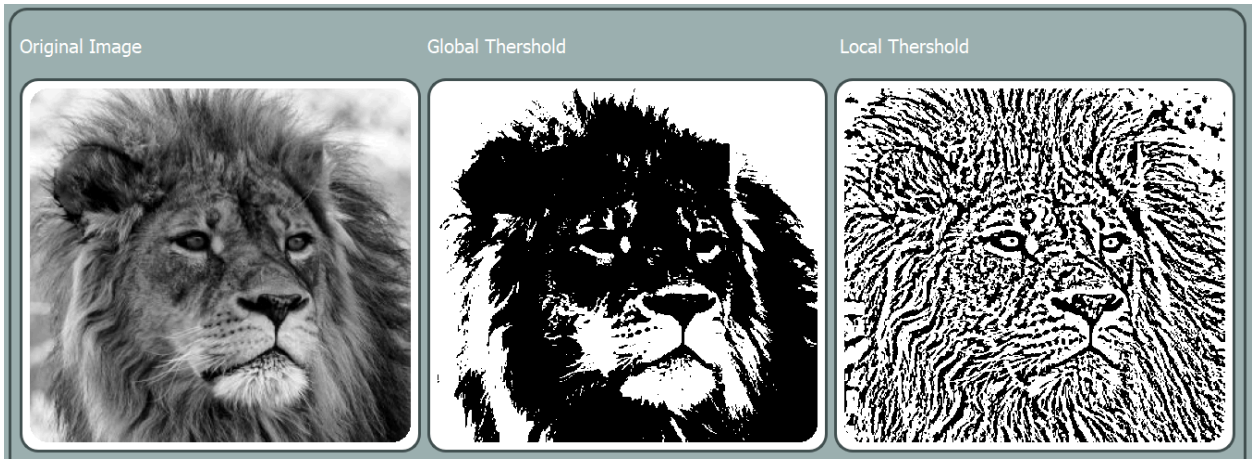


2.4 Local and Global Thresholding Mode :

Thresholding segments an image into foreground and background based on pixel intensity.

- Global Thresholding: Applies a single threshold to the whole image. Pixels above the threshold are foreground, and those below are background. Simple and efficient, but less effective with uneven lighting.
- Local (Adaptive) Thresholding: Calculates a separate threshold for each region of the image, based on local intensity statistics (e.g., mean). This handles uneven illumination better, making it more suitable for complex images.





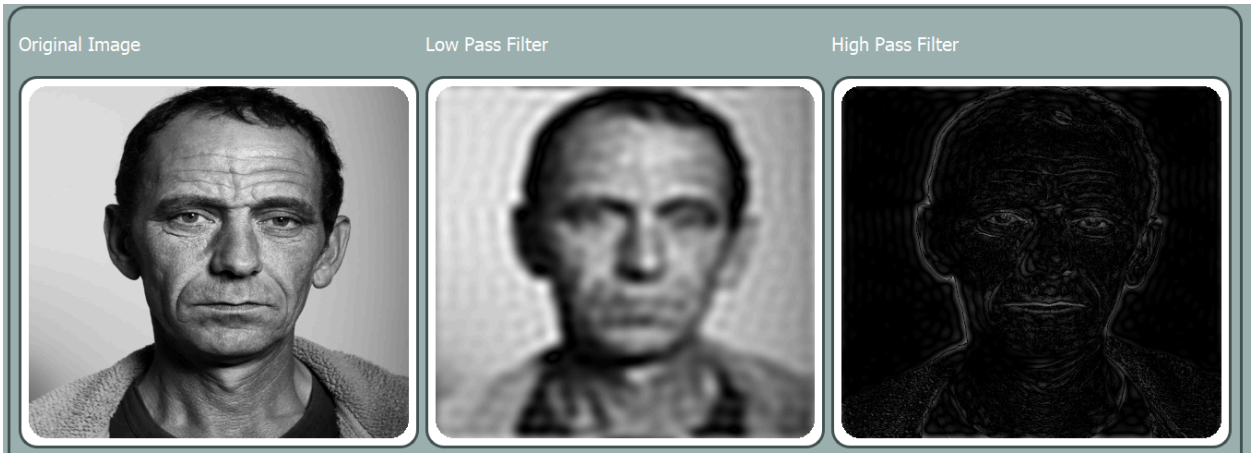
2.5 Frequency Domain Filters:

Images are transformed to the frequency domain using the Discrete Fourier Transform (DFT). Filtering is applied using circular masks:

- Low-pass filter: Passes low frequencies within a defined radius, removing fine details (smoothing).
- High-pass filter: Passes high frequencies outside a defined radius, enhancing edges.

After masking the frequency spectrum, the Inverse DFT reconstructs the filtered image.

Radius of Mask =20



Radius of Mask =60