Image Editor

J.S.V. Samhitha (22116040), Muskan Gupta (22116057), Nancy Chouksey (22116059)
Department of Electronics and Communication Engineering
Indian Institute of Technology Roorkee
Digital Image Processing (ECN-316)

Abstract: ***This project presents an advanced image editor developed using OpenCV in Python. It includes essential features like watermarking, cropping, and converting images to black and white. Additionally, it implements histogram equalization for contrast enhancement, noise removal, blurring, sharpening (using LPF and HPF), and edge detection, offering comprehensive image processing functionalities.***

Image Editing Techniques:

A. Conversion to Grayscale:

Grayscale conversion is the process of transforming a colour image into shades of gray, where each pixel represents only the intensity (brightness) of the image, without colour information. In this process, the colour channels (Red, Green, and Blue) are combined into a single channel that represents the brightness.

Key Features:

- Separates the BGR channels of the image.
- Uses the below mentioned matrix to compute the grayscale intensity (Y)

$$\begin{bmatrix} Y \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256} \begin{bmatrix} 65.738 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

here, Y is Luminance, C_B is Chrominance-Blue and C_R is Chrominance-Red. The component Y is used for grayscale conversion because it represents the perceived brightness of the image.

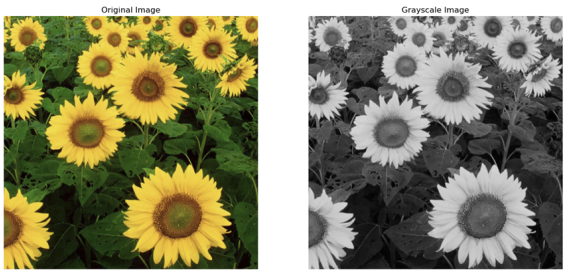

Fig. 1: Conversion to Grayscale

B. Contrast Enhancement:

An image's histogram reflects its appearance: dark images cluster on low intensities, light ones on high. Low-contrast images have narrow, centered histograms, appearing dull, while high-contrast images span a wide range, offering rich detail.

Key Features:

- Enhances contrast by evenly distributing intensity levels, improving dynamic range and visual quality.
- Uses cumulative distribution function to adjust pixel values.

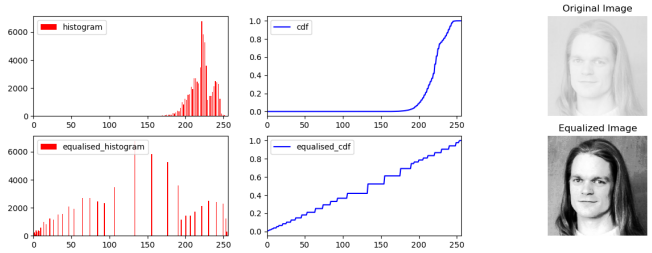

Fig. 2: Contrast Enhancement

C. Adjusting HSV of an Image:

The HSV (Hue, Saturation, Value) colour space is a conical representation of colours, commonly used in image processing. It separates image intensity (brightness) from colour information, making it particularly useful for tasks like colour adjustment, segmentation, and filtering.

Key Features:

- Adjust Hue, Saturation, and Brightness in real-time using trackbars.
- Instant feedback on image adjustments.

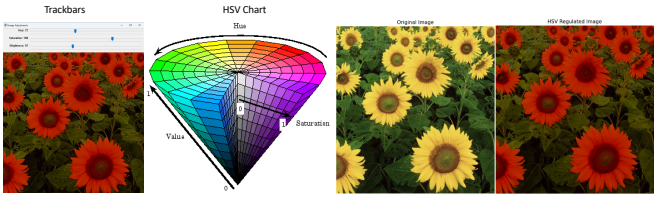

Fig. 3: Adjusting HSV

D. Cropping:

Cropping in image processing refers to the technique of removing unwanted outer parts of an image to focus on a specific region of interest (ROI). This operation allows for better framing, improved aesthetics, and can help isolate features that are more relevant for analysis or processing.

Key Features:

- The user selects two points on the image using mouse clicks to define the cropping area.
- A callback function captures the coordinates of mouse clicks.
- The crop area is determined based on the coordinates provided by the user.
- This approach can be used for preprocessing images in tasks like annotation, ROI extraction, or segmentation.

Fig. 4: Cropping Image

E. Watermarking:

Watermarking is the process of embedding information, typically in the form of text, logos, or patterns, into an image in such a way that it is imperceptible or minimally visible to the viewer, yet detectable by certain algorithms.

Fig. 5: Location of watermark

Fig. 6: Output of Watermarking

Key features:

- Users can click on the image to select a region where the watermark will be placed.
- The watermark automatically aligns based on the click location (corners or center), as shown in Fig. 5.
- Text font, size, colour, and transparency can be modified.
- The watermark is blended with the original image for a semi-transparent effect.

F. Scaling and changing Aspect Ratio:

Scaling and aspect ratio are fundamental concepts in image processing, particularly in resizing and transforming images. They are used to adjust the size of an image while maintaining its visual integrity and proportionality.

$$\text{Aspect ratio} = \text{width}/\text{height}$$

Key features:

- Resizes the image while maintaining aspect ratio by specifying width or height.
- Allows resizing to a custom width and height, changing the aspect ratio.

G. Noise Removal:

In image processing, noise removal is a crucial task that involves reducing or eliminating unwanted random variations in pixel values caused by external factors like sensor errors, transmission errors, or environmental conditions. Different filters are used to remove noise while preserving important image details.

Key Features:

- Multiple filters (Mean, Median, Max, Min, Alpha Trimmed) are applied to reduce noise.
- Adjustable kernel size for flexible filtering.

TABLE I: Filters

Arithmetic Mean Filter	$\hat{f}(x,y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} g(r,c)$
Geometric Mean Filter	$\hat{f}(x,y) = \left[\prod_{(r,c)\in S_{xy}} g(r,c) \right]^{\frac{1}{mn}}$
Median Filter	$\hat{f}(x,y) = \underset{(r,c)\in S_{xy}}{\text{median}} \{g(r,c)\}$
Min Filter	$\hat{f}(x,y) = \min_{(r,c)\in S_{xy}} \{g(r,c)\}$
Max Filter	$\hat{f}(x,y) = \max_{(r,c)\in S_{xy}} \{g(r,c)\}$
Alpha-Trimmed Mean Filter	$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(r,c)\in S_{xy}} g_R(r,c)$

where, S_{xy} represent the set of coordinates in a rectangular subimage window. The value of the restored image $\hat{f}$ at point (x,y) is computed using one of the filter mentioned above using the pixels in the region defined by S_{xy}.

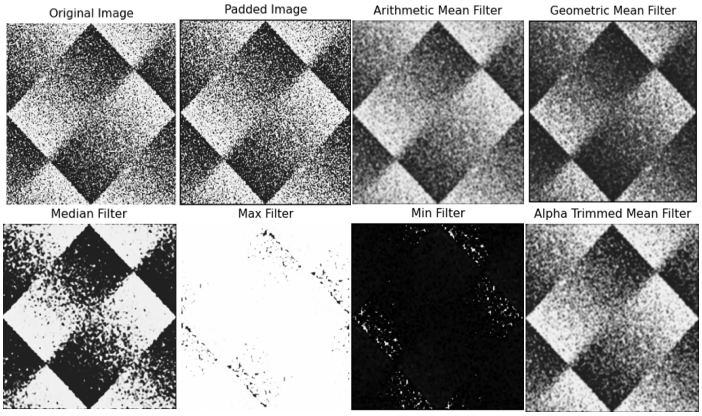

Fig. 7: Kernel Size = 13, Alpha = 0.1(For Alpha Trimmed Mean Filter)

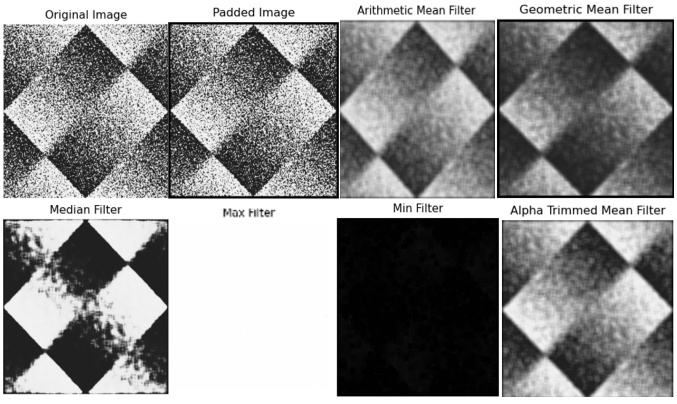

Fig. 8: Kernel Size = 29, Alpha = 0.1(For Alpha Trimmed Mean Filter)

- Median filter effectively removes salt-and-pepper noise, preserving edges with small kernels but causing smoothing with large kernels.
- Arithmetic mean filter reduces general noise but introduces blurring, especially with larger kernels.
- Geometric mean filter reduces noise while slightly preserving finer details, though it still blurs edges at large kernel sizes.
- Max filter emphasizes bright regions, reducing dark noise but distorting details, especially with larger kernels.

- Min filter highlights dark regions, reducing bright noise but losing detail and distorting the image as the kernel size increases.
- Alpha-trimmed mean filter balances noise removal and detail retention, effectively handling various noise types with proper kernel size and alpha values.

Smaller kernels (Fig. 7) offer moderate noise reduction while preserving edges and details.
Larger kernels (Fig. 8) aggressively reduce noise but over-smooth the image, leading to loss of fine details and edge sharpness.

H. Blurring and Sharpening:

Filters play a vital role in enhancing image quality by reducing noise and preserving critical details.
Blurring is a technique to reduce image noise and smooth details by averaging pixel values, often used to create a softer appearance.
Sharpening is a technique to enhance image clarity by emphasizing edges and fine details, making features more distinct.

Key Features:

- Includes filters such as Box, Gaussian, Laplacian, Max, Min, Median, and Unsharp Mask for achieving various effects.
- Allows control over the intensity of blurring and sharpening through adjustable kernel size.

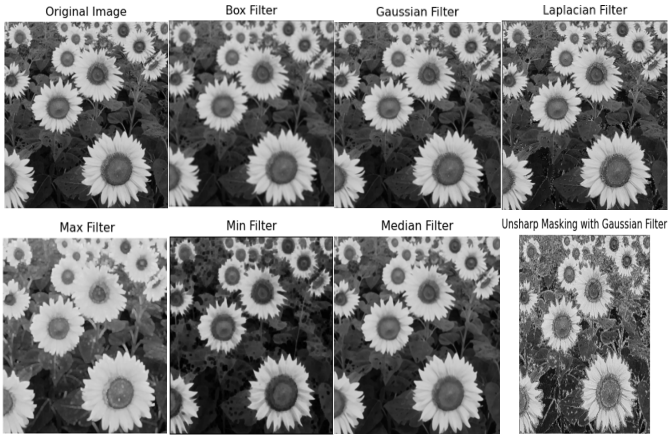

Fig. 9: Blurring and Sharpening

Fig. 9 shows blurring and sharpening with Kernel Size = 3 for Laplacian Filter, Kernel Size = 7 for Other filters and k = 10 for Unsharp Masking.

I. Edge Detection:

Edge detection in the code involves enhancing high-frequency components by subtracting a smoothed version of the image from the original. The Box Filter highlights edges by removing low-frequency details through local averaging. The Gaussian Filter further refines this process, leveraging a weighted smoothing kernel to preserve spatial relevance. Together, these techniques emphasize transitions in intensity, effectively detecting and highlighting edges in the image.

Key Features:

- Converts the image to grayscale and pads it to handle edge cases during filtering.
- Applies box filtering and Gaussian filtering to enhance edges by subtracting the filtered response from the original image.
- Uses iterative convolution with sliding kernels to detect edges with pixel-level precision.
- Provides adjustable parameters like kernel size and Gaussian variance, with visual comparison of results for analysis.

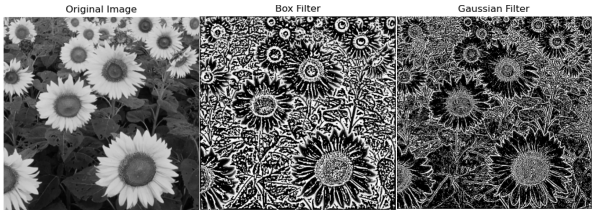

Fig. 10: Kernel Size = 13

Fig. 11: Kernel Size = 3

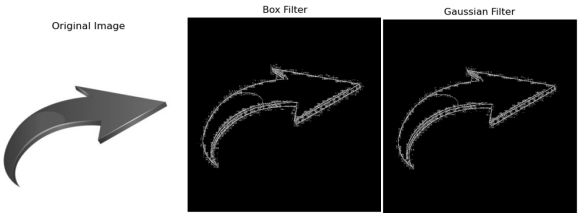

Fig. 12: Kernel Size = 3

Conclusion:

This image editor project successfully implemented key image processing techniques including grayscale conversion, HSV transformation, blurring, sharpening, contrast enhancement, watermarking, edge detection, scaling, noise removal, and cropping. These methods allowed for effective image manipulation, improving visual quality, highlighting key features, and providing flexibility in resizing and focusing on specific areas. The project demonstrates the practical application of image processing tools for enhancing and modifying images to meet diverse needs.

Reference:

R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2018.

GitHub Link:

https://github.com/Muskan05Gupta/Image-Editor-using-DIP-Fundamentals