Haramaya University

School of Graduate Studies

Department of Computer Science

Image Filtering: Spatial Filtering

Computer Vision and Image Processing

Abrahim Abdo

College: Computing and Informatics

Department: Computer Science

Program: Computer Vision and Image Processing

Major Advisor: Gadisa . (PhD)

27 February 2022

Haramaya, Oromia, Ethiopia

1. Introduction

Vision allows humans to perceive and understand the world surrounding them, while computer vision aims to duplicate the effect of human vision by electronically perceiving and understanding an image [1]. For human the most important parts of the human visual system are the eyes and the brain in particular, the part of the brain that is associated with vision is the visual cortex. The eye is the device that captures light that has bounced off nearby objects. While computer use electronically perceiving and understanding an image usually by camera to capture visual representation of a real-life object (a person or any other object) in a two-dimensional form is called an image [2]. An image is nothing but a collection of pixels in different color spaces. Image capture can be performed by a simple camera (or pair of cameras, to give stereoscopic vision), which converts light signals from a scene to electrical signals, much as the human visual system does. Having obtained these light signals, which are simply a set of 1s and 0s.

Image representation and image analysis tasks start form Image understanding**.** Image understanding by a machine can be seen as an attempt to find a relation between input image(s) and previously established models of the observed world. Transition from the input image(s) to the model reduces the information contained in the image to relevant information for the application domain. This process is usually divided into several steps and several levels representing the image are used. The bottom layer contains raw image data and the higher levels interpret the data. Computer vision designs these intermediate representations and algorithms serving to establish and maintain relations between entities within and between layers. This hierarchy of image representation and related algorithms is frequently categorized in an even simpler way low-level image processing and high-level image understanding [1].

Low-level processing methods usually use very little knowledge about the content of images. In the case of the computer knowing image content, it is usually provided by high-level algorithms or directly by a human who understands the problem domain. Low level methods may include image compression, pre-processing methods for noise filtering, edge extraction, and image sharpening [1]. If the image is to be processed using a computer it will be digitized first, after which it may be represented by a rectangular matrix with elements corresponding to the brightness at appropriate image locations. More probably, it will be presented in color, implying (usually) three channels: red, green and blue. Very often, such a data set will High-level processing is based on knowledge, goals, and plans of how to achieve those goals, and artificial intelligence methods are widely applicable [1].

High-level computer vision tries to imitate human cognition and the ability to make decisions according to the information contained in the image. In the example described, high-level knowledge would be related to the 'shape' of a cow and the subtle interrelationships between the different parts of that shape, and their (inter)dynamics. High-level vision begins with some form of formal model of the world, and then the 'reality' perceived in the form of digitized images is compared to the model. A match is attempted, and when differences emerge, partial matches (or subgoals) are sought that overcome them; the computer switches to low-level image processing to find information needed to update the model. [1].

Computer vision is expected to solve very complex tasks, the goal being to obtain similar results to those provided by biological systems. To illustrate the complexity of these tasks, consider Figure 1 in which a particular image representation is presented. Machine vision generally consists of the following five steps or operations (Fig. 1). First image acquisition operations to convert images into digital form [3].

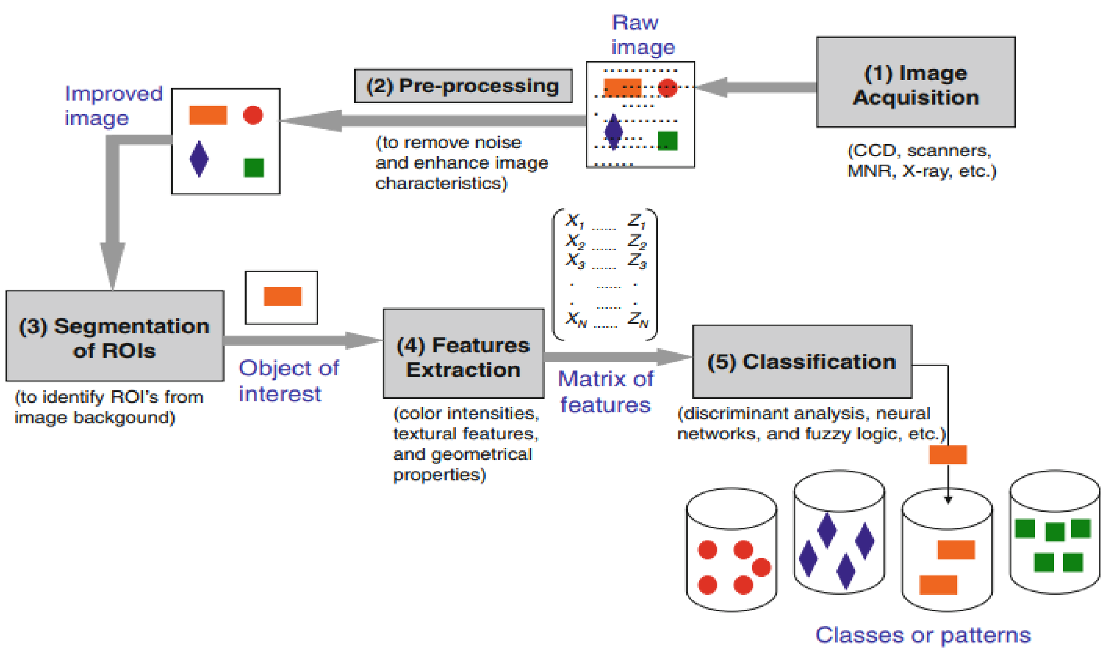


Figure 1: general image preprocessor [3].

1.1Statement of Problem

Why is computer vision difficult?

This philosophical question provides some insight into the complex landscape of computer vision [1].

Loss of information in 3D → 2D is a phenomenon which occurs in typical image capture devices such as a camera or an eye.

Interpretation of image(s) is a problem humans solve unwittingly that is the principal tool of computer vision. When a human tries to understand an image then previous knowledge and experience is brought to the current observation.

Interpretation: image data -→ model

The (logical) model means some specific world in which the observed objects make sense. Considering observations (images) as an instance of formal expressions, semantics studies relations between expressions and their meanings. The interpretation of image(s) in computer vision can be understood as an instance of semantics. Practically, if the image understanding algorithms know into which particular domain the observed world is constrained, then automatic analysis can be used for complicated problems.

Noise is inherently present in each measurement in the real world. Its existence calls for mathematical tools which are able to cope with uncertainty; an example is probability theory. Of course, more complex tools make the image analysis much more complicated compared to standard (deterministic) methods.

Too much data. Images are big, and video increasingly the subject of vision applications correspondingly bigger. Technical advances make processor and memory requirements much less of a problem than they once were, and much can be achieved with consumer level products. Nevertheless, efficiency in problem solutions is still important and many applications remain short of real-time performance.

Brightness measured in images is given by complicated image formation physics. The radiance

(≈ brightness, image intensity) depends on the irradiance (light source type, intensity and position), the observer's position, the surface local geometry, and the surface reflectance properties.

2. Proposed solution

Image quality

An image might be degraded during capture, transmission, or processing, and measures of image quality can be used to assess the degree of degradation. The quality required naturally depends on the purpose for which an image is used. Methods for assessing image quality can be divided into two categories: subjective and objective. Subjective methods are often used in television technology, where the ultimate criterion is the perception of a selected group of professional and lay viewers. They appraise an image according to a list of criteria and give appropriate marks [1].

Digital image processing is a part of signal processing which uses computer algorithms to perform image processing on digital images. It has numerous applications in different studies and researches of science and technology. The fundamental steps in Digital Image processing are image acquisition, image enhancement, image analysis, image reconstruction, image restoration, image compression, image segmentation, image recognition, and visualization of image. The main sources of noise in digital image processing come under image acquisition and image transmission. Image Enhancement basically improves the visual quality of the image by providing clear images for human observer and for machine in automatic image processing techniques. Digital image processing has fundamental classes depending on their operations [4]:

Noise is usually described by its probabilistic characteristics. Idealized noise, called white noise is often used. White noise has a constant power spectrum, meaning that all noise frequencies are present and have the same intensity. For example, the intensity of white noise does not decrease with increasing frequency as is typical in real-world signals. White noise is frequently employed to model the worst approximation of degradation, the advantage being that its use simplifies calculations. A special case of white noise is **Gaussian noise**. A random variable with a **Gaussian** (normal) distribution has its probability density function given by the Gaussian curve.

Quantization noise occurs when insufficient quantization levels are used, for example, 50 levels for a monochromatic image. In this case false contours appear. **Impulse noise** means that an image is corrupted with individual noisy pixels whose brightness differs significantly from that of the neighborhood. The term salt-and pepper noise is used to describe saturated impulsive noise an image corrupted with white and/or black pixels is an example. Salt-and-pepper noise can corrupt binary images [1].

In this project we went to look over the method used to overcome all this mentioned above problem. Local pre-processing methods are divided into two groups according to the goal of the processing. **Smoothing** aims to suppress noise or other small fluctuations in the image; it is equivalent to the suppression of high frequencies in the Fourier transform domain. Unfortunately, smoothing also blurs all sharp edges that bear important information about the image.

Gradient operators are based on local derivatives of the image function. Derivatives are bigger at locations of the image where the image function undergoes rapid changes, and the aim of gradient operators is to indicate such locations in the image. Gradient operators have a similar effect to suppressing low frequencies in the Fourier transform domain. Noise is often high frequency in nature; unfortunately, if a gradient operator is applied to an image, the noise level increases simultaneously. Clearly, smoothing and gradient operators have conflicting aims. Some pre-processing algorithms solve this problem and permit smoothing and edge enhancement simultaneously

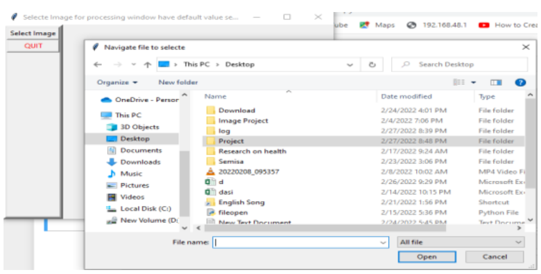


Figure 2:GUI- select image file

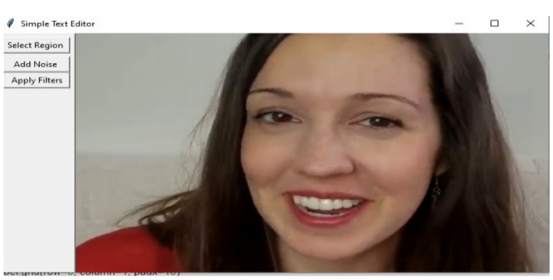


Figure 3:GUI- display image file

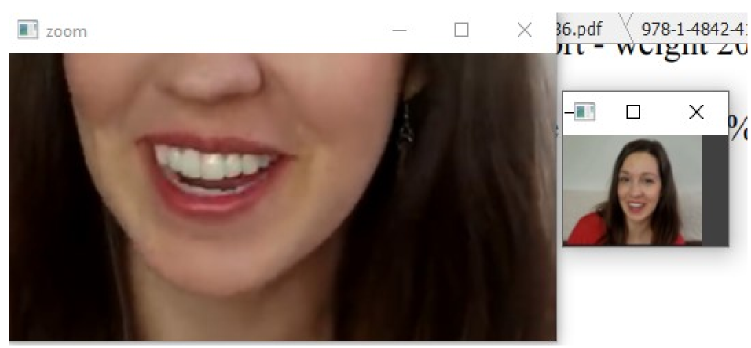


Figure 4:GUI- select image part from image

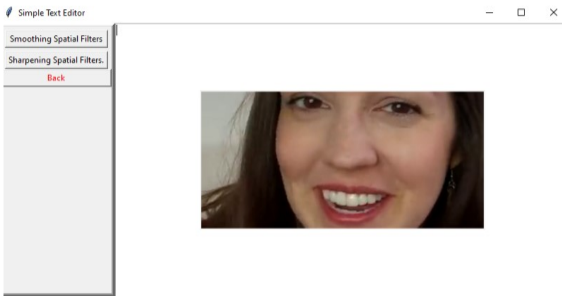
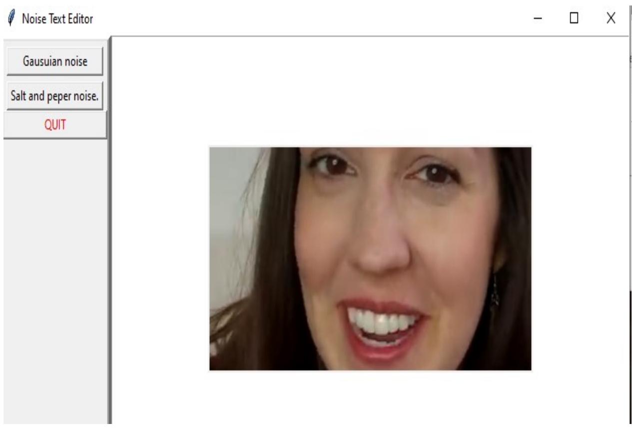


Figure 5:GUI- display selected image part



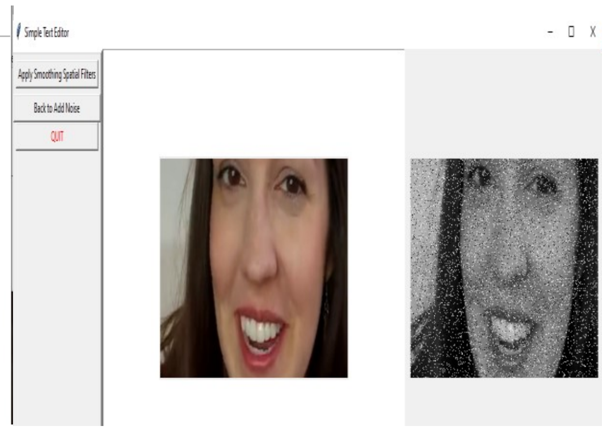


Figure 6:GUI- select noise type file

Figure 7:GUI- salt and paper noise applied and displayed

A. Image smoothing

Image smoothing uses redundancy in image data to suppress noise, usually by some form of averaging of brightness values in some neighborhood O. Smoothing poses the problem of blurring sharp edges, and so we shall consider smoothing methods, which are, edge preserving here, the average is computed only from points in the neighborhood, which have similar properties to the point being processed [5].

Smoothing Spatial Filters: contain Averaging filter,Median filter, Min Filter and Max filter.

a) Averaging, statistical principles of noise suppression

Assume that the noise value ν at each pixel is an independent random variable with zero mean and standard deviation σ. We might capture the same static scene under the same conditions n times. From each captured image a particular pixel value g_i, $i = 1, \dots, n$ is selected. An estimate of the correct value can be obtained as an average of these values, with corresponding noise values $\nu_1, \dots, \nu_n$

$$\frac{g_1 + \ldots + g_n}{n} + \frac{\nu_1 + \ldots + \nu_n}{n} \, . \tag{5.24}$$

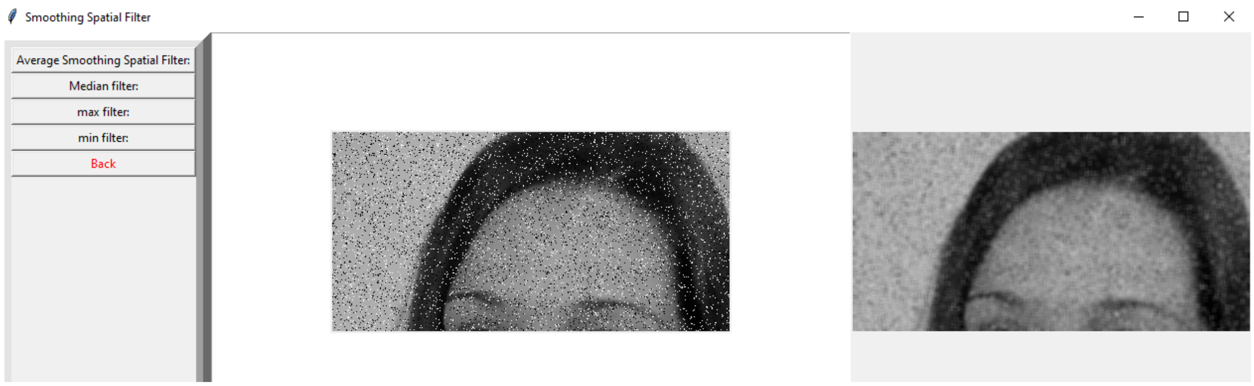


Figure 8:GUI- Average filter.

The second term here describes the noise, which is again a random value with zero mean and standard deviation $\sigma/\sqrt{n}$. Thus, if n images of the same scene are available, smoothing can be accomplished without blurring the image by

$$f(i,j) = \frac{1}{n} \sum_{k=1}^{n} g_k(i,j) \, . \tag{5.25}$$

b) **Median filtering**

In probability theory, the median divides the higher half of a probability distribution from the lower half. For a random variable x, the median M is the value for which the probability of the outcome $x < M$ is 0.5

Figure 9:GUI- median filter.

The median of a finite list of real numbers is simply found by ordering the list and selecting the middle member. Lists are often constructed to be odd in length to secure uniqueness

c) **Non-linear mean filter**

The non-linear mean filter is another generalization of averaging techniques it is defined by

$$f(m,n) = u^{-1}\left(\frac{\sum_{(i,j)\in\mathcal{O}} a(i,j)\, u(g(i,j))}{\sum_{(i,j)\in\mathcal{O}} a(i,j)}\right), \tag{5.32}$$

where f(m, n) is the result of the filtering, g(i, j) is the pixel in the input image, and O is a local neighborhood of the current pixel (m, n). The function u of one variable has an inverse function u-1; the a(i, j) are weight coefficients.

Figure 10:GUI- mean min image file

Figure 11:GUI- mean max image file

B. Image sharpening

Image sharpening has the objective of making edges steeper the sharpened image is intended to be observed by a human. The sharpened output image f is obtained from the input image g as [6]

$$f(i,j) = g(i,j) - C\,S(i,j)\,, \qquad (5.36)$$

a) Edge detectors

Edge detectors are a collection of very important local image pre-processing methods used to locate changes in the intensity function; edges are pixels where this function (brightness) changes abruptly. An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of that pixel. It is a vector variable with two components, magnitude and direction.

b) Unsharp masking

is often used in printing industry applications. A signal proportional to an unsharp (e.g., heavily blurred by a smoothing operator) image is subtracted from the original image. A digital image is discrete in nature and so equations (5.33) and (5.34), containing derivatives, must be approximated by differences. The first differences of the image g in the vertical direction (for fixed i) and in the horizontal direction (for fixed j) are given by

$$\begin{aligned} \Delta_i\, g(i,j) &= g(i,j) - g(i-n,j)\,, \\ \Delta_j\, g(i,j) &= g(i,j) - g(i,j-n)\,, \end{aligned} \qquad (5.37)$$

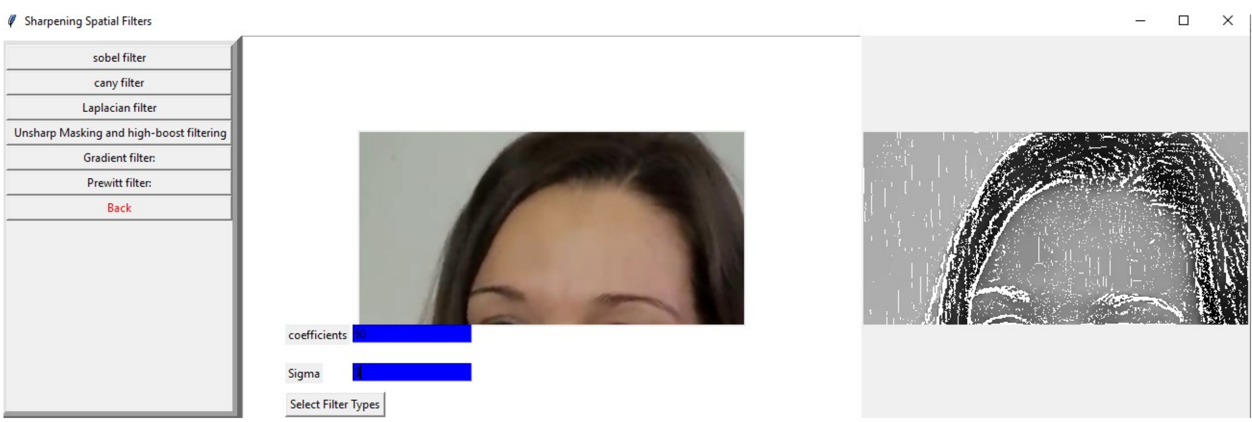


Figure 12:GUI- unsharpen mask image file

c) Laplace operator

The Laplace operator ∇*2* is a very popular operator approximating the second derivative which gives the edge magnitude only. The Laplacian, equation (5.35), is approximated in digital images by a convolution sum. A 3 × 3 mask h is often used; for 4-neighborhoods and 8-neighborhoods it is defined as.

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \qquad (5.41)$$

A Laplacian operator with stressed significance of the central pixel or its neighborhood is sometimes used. In this approximation it loses invariance to rotation

$$h = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}, \qquad h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}. \qquad (5.42)$$

The Laplacian operator has a disadvantage—it responds doubly to some edges in the image

Figure 13:GUI- Laplacian operator image file

d) Prewitt operator

The Prewitt operator, similarly to the Sobel, Kirsch, and some other operators, approximates the first derivative. The gradient is estimated in eight (for a 3 × 3 convolution mask) possible directions, and the convolution result of greatest magnitude indicates the gradient direction. Larger masks are possible. We present only the first three 3 × 3 masks for each operator; the others can be created by simple rotation

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad \ldots \qquad (5.43)$$

Figure14: The Prewitt operator.

e) Sobel operator

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \ldots \qquad (5.44)$$

The Sobel operator is often used as a simple detector of horizontality and verticality of edges, in which case only masks h1 and h3 are used. If the h1 response is y and the h3 response x, we might then derive edge strength (magnitude)

$$\sqrt{x^2 + y^2} \qquad \text{or} \qquad |x| + |y| \qquad (5.45)$$

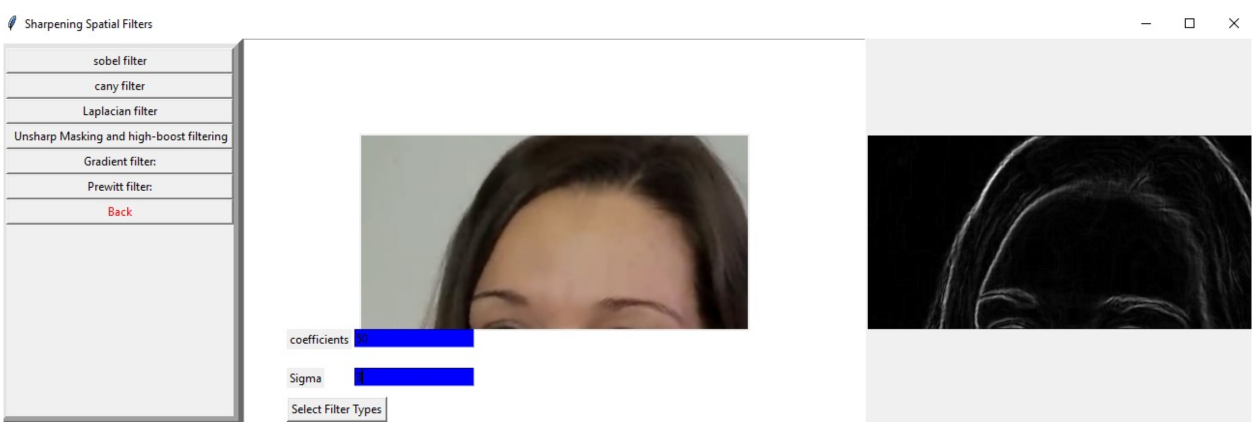


Figure 15: sobel operation.

f) Canny edge detection

Canny proposed an approach to edge detection that is optimal for step edges corrupted by white noise.

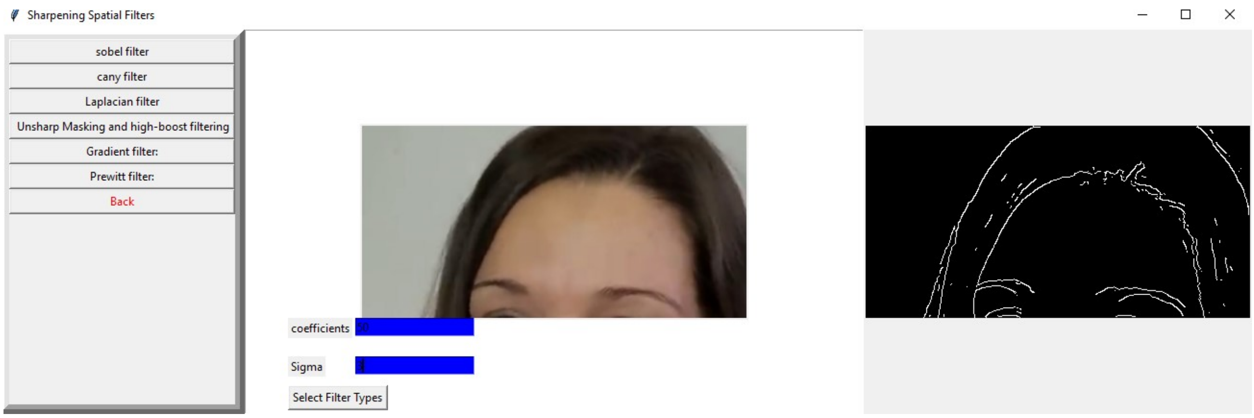


Figure 16:canny edge detection.

Gradient

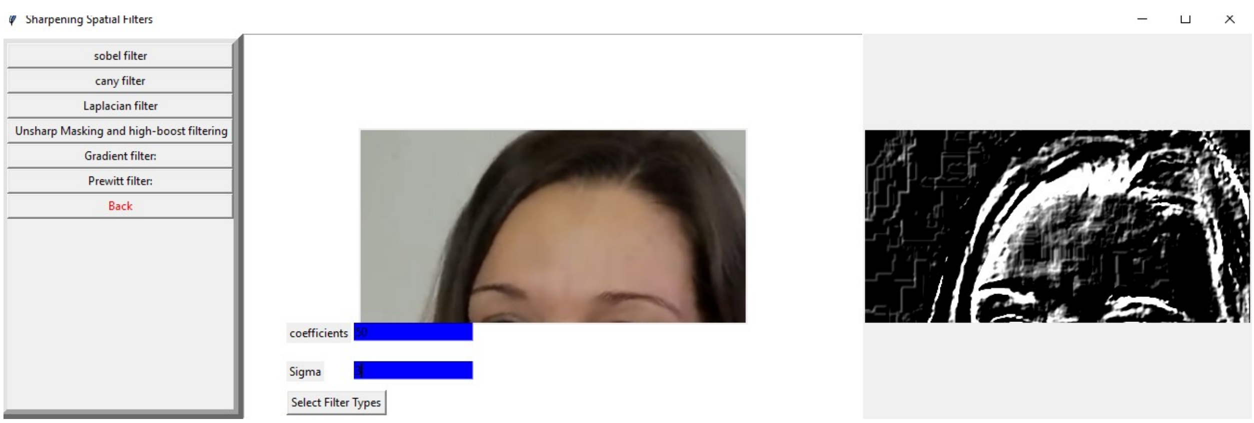


Figure 17: gradient calculator

3. References

[1] Milan Sonka, Vaclav Hlavac, and Roger Boyle, Image Processing, Analysis, and Machine Vision, Fourth Edition, Stamford,USA: Global Engineering, 2015.

[2] H. Singh, Practical Machine Learning and Image Processing, Allahabad, Uttar Pradesh, India: APress, 2019.

[3] Fernando Mendoza and Renfu Lu, "Basics of Image Analysis," Springer Science+Business Media, New York , January 2015.

[4] Swati Dewangan and Anup Kumar Sharma, "Image Smoothening and Sharpening using Frequency Domain Filtering Technique," *International Journal of Emerging Technologies in Engineering Research (IJETER),* vol. 5, no. 4, pp. 169-175, April (2017).

[5] O. O. S. C. V. Library, "Homepage: <https://opencv.org>," 1 2 2022. [Online]. Available: https://github.com/opencv/opencv. [Accessed 1 2 2022].

[6] a. J. L. Stéfan van der Walt, "scikit-image: Image processing in Python," 3 9 2021. [Online]. Available: [https://scikit-image.org/](https://scikit-image.org). [Accessed 5 1 2022].