Python based graphical user interface (GUI) for different image processing filters

Mahbubur Rahman
Ingram School of Engineering
Texas State University
San Marcos, Tx, USA
m_r1011@txstate.edu

***Abstract*—This paper confers some popular image processing techniques which is assisted by Python based graphical User Interface (GUI). An image can be chosen by the user and can apply different filters to the selected image. User can choose the range of applicable parameters which will be employed to the images.**

Keywords—image processing, Python, OpenCV, GUI.

I. INTRODUCTION

Human beings are predominantly visual creatures, and our computing environments reflect this [1] . We depend on our vision to get perception of the world around us. Image processing and filtering is an important part of computer vision. Spatial filtering is used in a broad spectrum of image processing applications, so a solid under-standing of filtering principles is important [2]. To apply different filters with different tunable parameters are time consuming and painstaking. If we equipped all the filters in a single window and can facilitate the user with different changeable working parameters, a lot of human effort can be minimized. In this paper, our proposed graphical user interface incorporates four major types of filtering which consists of eleven filters. Python is used as a primary programming language to construct the GUI. Opencv, tkinter and other libraries are used to build the graphical interface and filter application.

II. LITARATURE REVIEW

A systematic literature review of different image processing filtering techniques is described as follows:

A. Linear type

Linear filtering is concerned with finding suitable ways to modify the frequency content of an image [2]. In the spatial domain we do this via convolution filtering. In the frequency domain we do it with multiplicative filters. Four types of filtering have been incorporated in this section. Lowpass (also called Smoothing) spatial filters are used to reduce sharp transitions in intensity. box filters have limitations that make them poor choices in many applications. Gaussian low pass filters are useful when it is desired to reduce the effect of smoothing on edges. Gaussian kernels are the circularly symmetric kernels that are also separable. Gaussian kernels must be larger than box filters to achieve the same degree of blurring.

Filter type	Spatial kernel in terms of lowpass kernel, lp
Lowpass	lp x(,y)

A high pass filter tends to retain the high frequency information within an image while reducing the low frequency information. The kernel of the high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels. Sharpening of image is an often referred to as high pass filtering. A high pass filter transfer function is obtained by subtracting a lowpass function from 1. we obtain a high pass filter kernel in the spatial domain by subtracting a lowpass filter kernel from a unit impulse with the same center as the kernel. An image filtered with this kernel is the same as an image obtained by subtracting a low-pass-filtered image from the original image.

Filter type	Spatial kernel in terms of highpass kernel, lp
Highpass	hp(x,y)=d(x,y)−lp(x,y)

Band-pass filters attenuate signal frequencies outside of a range (band) of interest. In image analysis, they can be used to denoise images while at the same time reducing low-frequency artifacts such a uneven illumination. Band-pass filters can be used to find image features such as blobs and edges [3].

Filter type	Spatial kernel in terms of Bandpass kernel, lp
Bandpass	bp(x,y)=lp(x1,y1)- lp(x2,y2)

A band reject filter is useful when the general location of the noise in the frequency domain is known. A band reject filter blocks frequencies within the chosen range and lets frequencies outside of the range pass through [4].

Filter type	Spatial kernel in terms of Bandreject kernel, lp
Bandreject	br(x,y)= d(x,y)− bp(x,y)

B. Statistical type

Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the region encompassed by the filter. Smoothing is achieved by replacing the value of the center pixel with the value determined by the ranking result. The best-known filter in this category is the median filter. Median filters provide excellent noise reduction capabilities for certain types of random noise, with considerably less blurring than linear smoothing filters of similar size [2].

C. Derivative type

Derivative filters provide a quantitative measurement for the rate of change in pixel brightness information present in a digital image.

A Laplacian filter is an edge detector used to compute the second derivatives of an image, measuring the rate at which the first derivatives change. Laplacian filter kernels usually contain negative values in a cross pattern, centered within the array. The corners are either zero or positive values [5].

0	-1	0
-1	4	-1
0	-1	0

0	-1	0
-1	8	-1
0	-1	0

Fig. 1 Two commonly used Laplacian filter.

Laplacian filters are derivative filters used to find areas of rapid change (edges) in images. Since derivative filters are very sensitive to noise, it is common to smooth the image (e.g., using a Gaussian filter) before applying the Laplacian. This two-step process is call the Laplacian of Gaussian (LoG) operation [6].

The Prewitt operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector.

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

Fig. 2 Horizontal and vertical prewitt kernel.

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically, it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Fig. 3 Vertical and Horizontal Sobel kernel.

D. Histogram Equalization

Global histogram equalization (GHE) is a common technique used to enhance image contrast by utilizing the histogram information of the input image to create its transformation function. However, it usually fails to adapt with the local image brightness features. By contrast, Local histogram equalization (LHE) is suitable for handling the local feature; whereas it suffers from halo or blocking artifacts and requires high computation power[7].

Adaptive histogram equalization (ahe) is a contrast enhancement method designed to be broadly applicable and having demonstrated effectiveness. These algorithms include interpolated ahe, to speed up the method on general purpose computers; a version of interpolated ahe designed to run in a few seconds on feedback processors; a version of full ahe designed to run in under one second on custom VLSI hardware; weighted ahe, designed to improve the quality of the result by emphasizing pixels' contribution to the histogram in relation to their nearness to the result pixel; and clipped ahe, designed to overcome the problem of over enhancement of noise contrast. [8]

III. Procedure & Result

Firstly, we find the filter tuning parameters. To make the GUI more user friendly, Object oriented programming (OOP) is applied. For making the graphical window, tkinter library is used along with OpenCV, scikit-image, pillow, Numpy libraries. After importing all the necessary libraries, a new class called 'APP' is developed by python object-oriented programming. There are three attributes in this class which are 'window' and 'window_title'. All the necessary buttons are created by tkinter button and adorn the position of the button in the specific location. Scaling buttons are placed on the right of each filter in the window. Secondly, we created the method of the OOP and linked the method with the filter and scaling button.

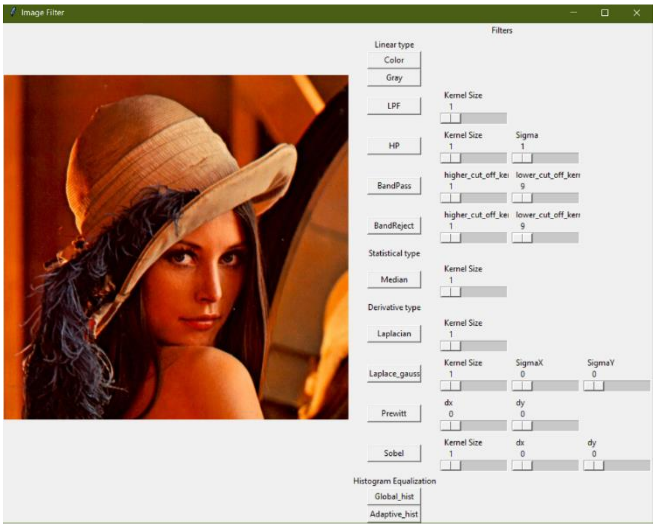

Fig. 4 Graphical User interface

After debugging the code, new window will pop up to take user defined image into the GUI.

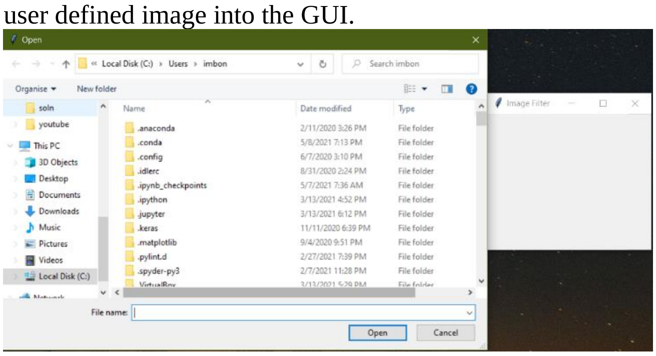

Fig. 5 Select input image.

In the canvas, all the filter button is placed in the thirteenth column. Scaling button of the respective filter is placed on the 14^{th} to 15^{th} column of the window. It can incorporate either color or grayscale image. If the input image is color image, we can easily convert the color image to grayscale image by simply clicking the button "Gray". Although, Opencv is being used to read the input image but color is corrected while showing the image on the canvas. The result image after applying each filter are shown below:

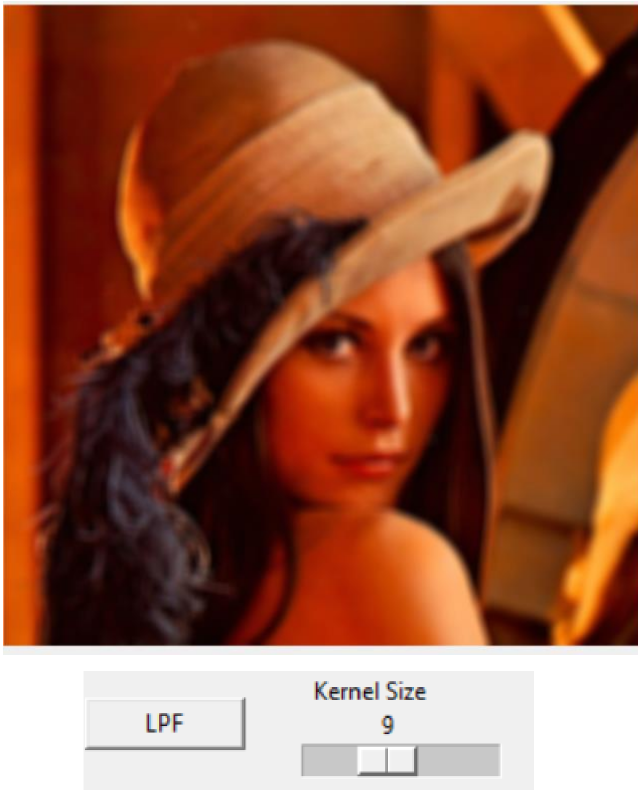

Fig. 6 Low Pass filter with kernel scaling

In the above LPF, gaussian blurring is used as lowpass filter. Kernel size can be changed. The blur effect increases with the increasing kernel size.

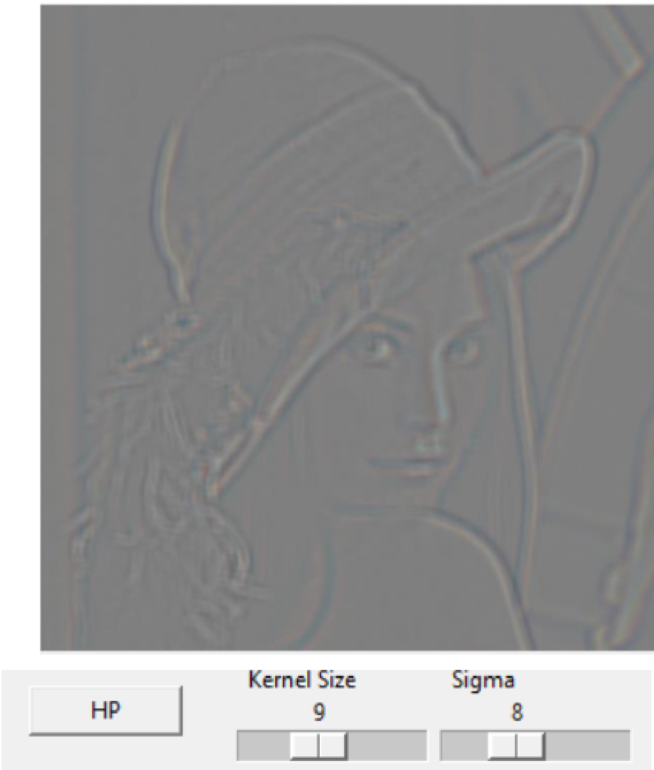

Fig. 7 High Pass filter with kernel & sigma scaling

Fig. 8 Band Pass filter with kernel scaling

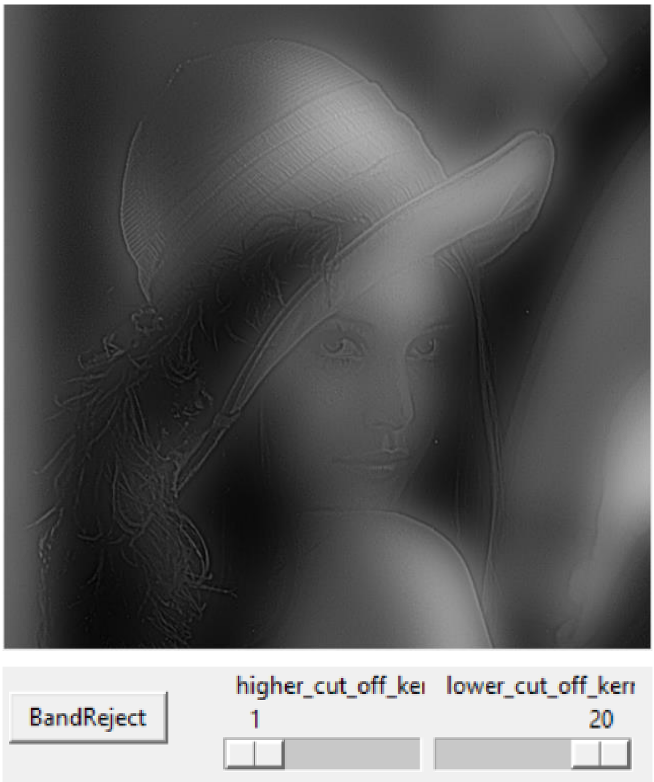

Fig. 9 Band reject filter with kernel scaling

Median filter is a nonlinear filter. Only median filter as statistical filter is considered here. Odd number of kernels is calculated in the method section. There are always odd kernel options in the scale button for median filter. Median filter can be applied on both color or grayscale image.

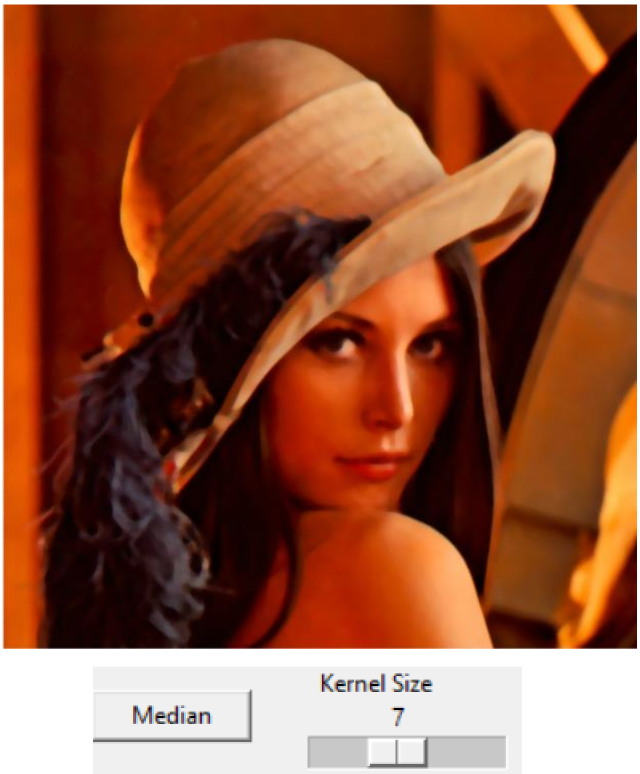

Fig. 10 Median filter with kernel scaling

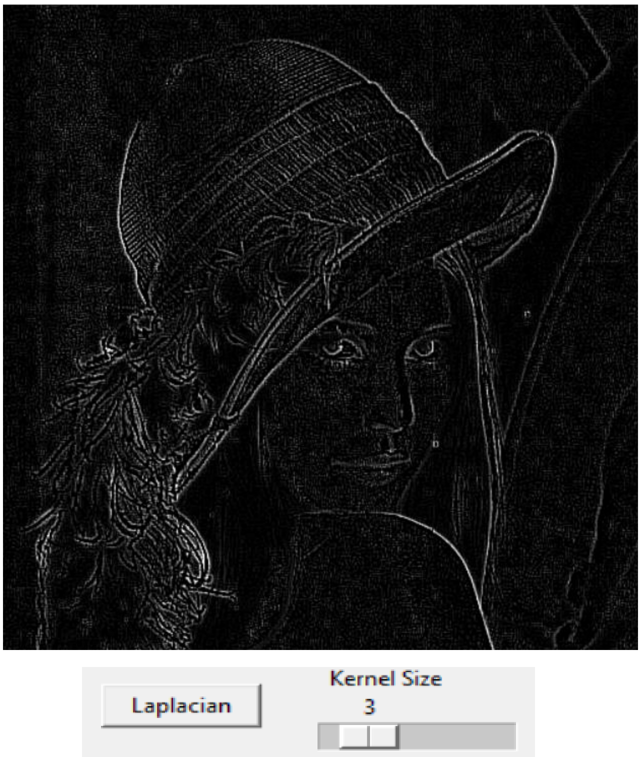

Fig. 11 Laplacian filter with kernel scaling

Laplacian filter is a derivative filter. It extracts the edges from the images. Laplacian gaussian is a gaussian blur applied derivative technique. Kernel size is applied for gaussian blurring, dx and dy implies horizontal and vertical edges. Horizontal and vertical edges can be applied individually or simultaneously.

Fig. 12 Laplacian gaussian filter with kernel scaling

Prewitt is a two-dimensional edge filter. one of kernel extracts horizontal edges and second kernel extracts vertical edges. If we add both simultaneously, we can get all the horizontal and vertical edges.

Fig. 13 Prewitt filter with vertical and horizontal kernel

Similar to the prewitt filter Sobel filter has similar kind of kernel characteristics but with different values. Like the prewitt kernel, Horizontal or vertical edge filter can be applied separately or instantaneously.

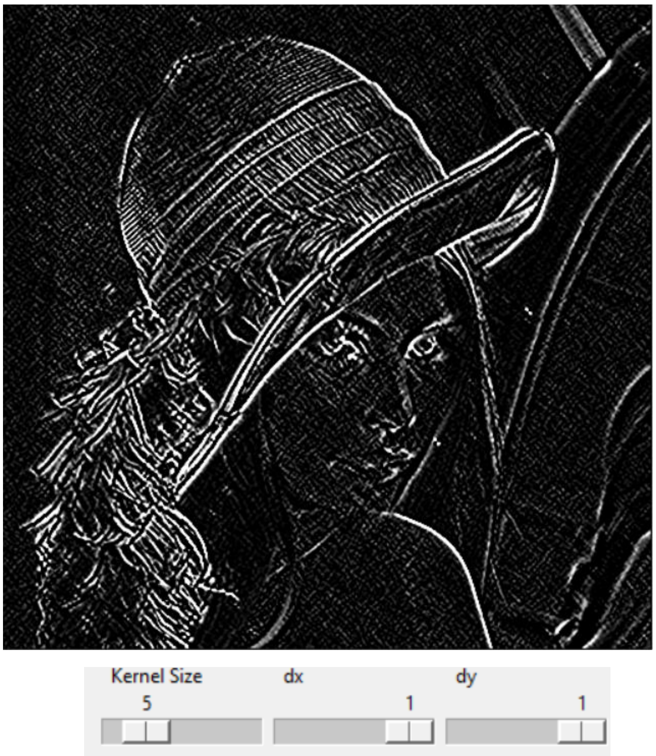

Fig. 14 Sobel filter with vertical and horizontal kernel

Global histogram and adaptive histogram equalization both are employed to the gray scale image. We can give either form of images. Model will transform automatically and give the equalized output.

Fig. 15 Global histogram equalization

Fig.16 Adaptive histogram equalization

IV. Conclusion

Python 3.6.13 is used as debugger. There are some limitations of this GUI. Most of the filters applied to grayscale image. Only hand moving scaling incorporate in GUI, in future integer digit input can be considered. To select new image, existing window need to close and run the python code again. After, running the code it allows user to select new image. There are also limitation of selecting image greater than 512×512. Some of the image can be out of canvas. A new condition and resize option can be incorporated to resolve this problem. Other than this small flaw, the GUI works fine with our desired calibration.

Reference

[1] A. Mcandrew, *A Computational Introduction to Digital Image Processing*, Second Edi. 2015.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Fourth Edi. .

[3] “Band-pass filtering by Difference of Gaussians — skimage v0.19.0.dev0 docs.” https://scikit-image.org/docs/dev/auto_examples/filters/plot_dog.html (accessed May 08, 2021).

[4] “BANDREJECT_FILTER.” https://www.l3harrisgeospatial.com/docs/bandreject_filter.html (accessed May 08, 2021).

[5] “Spatial Filters - Laplacian/Laplacian of Gaussian.” https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm (accessed May 08, 2021).

[6] “Laplacian of Gaussian Filter.” https://academic.mu.edu/phys/matthysd/web226/Lab02.htm (accessed May 08, 2021).

[7] S. C. Chien, F. C. Chang, K. L. Hua, I. Y. Chen, and Y. Y. Chen, “Contrast enhancement by using global and local histogram information jointly,” in *International Conference on Advanced Robotics and Intelligent Systems, ARIS*, Feb. 2018, vol. 2017-September, doi: 10.1109/ARIS.2017.8297188.

[8] S. M. Pizer *et al.*, “ADAPTIVE HISTOGRAM EQUALIZATION AND ITS VARIATIONS.,” *Comput. vision, Graph. image Process.*, vol. 39, no. 3, pp. 355–368, Sep. 1987, doi: 10.1016/S0734-189X(87)80186-X.