EE 610 Image Processing (July-Nov 2021) Assignment 1: Basic Image Editor

Nihar Mahesh Gupte
Electrical Department
IIT Bombay
Mumbai, India
nihar71298@gmail.com

***Abstract*—Image processing refers to a set of operations performed on an image in order to either enhance an image or extract certain meaningful features out the image. The Image is visualized in the form of a 2 Dimensional (Gray scale) or 3 Dimensional (Colour) array, followed by application of a certain transformation in either spatial domain or frequency domain, to achieve desired results. The current project achieves the mentioned functionality, in spatial domain with the help of Graphical User Interface (GUI). The GUI contains features such as loading an image, saving an image, applying transformations to the image in spatial domain from a list of predefined transformations that include logarithmic transformation, gamma correction, blurring as required, histogram equalization and sharpening the image with a mechanism to control the extent of sharpening. In this report, first we will look at the GUI design, followed by Image processing algorithms, and finally at some experimental results of these algorithms.**

***Index Terms*—Python, GUI, tkinter, Image Processing, Histogram Equalization, Laplacian Filter, Gamma correction, Logarithm transform, Image Sharpening, Convolution**

I. Introduction

Image processing has a variety of applications [2] in various fields like medical imaging like X-ray imaging and Magnetic Resonance Imaging (MRI), computer vision, defect detection, satellite imaging, agriculture for purpose like weed detection, Biometric verification etc. Image processing techniques are necessary in order to enhance certain required quality of an image, and suppress the unwanted distortions. For example, sharpening an image can cause the details of the image to be highlighted.Gaussian filter or any other low pass filter when applied on an image, reduces the aliasing effect. The project focuses on applying a chosen set of transformations from a given list to an image. The Graphical User Interface (GUI) allows the user to apply and observe these transformations on a given image by merely clicking on a set of provided buttons. The major focus here lies on development of GUI as well as developing algorithms to implement the transformations.

II. GUI design

The language used for the implementation of GUI in this project is Python, with the help of tkinter [1] library. The GUI developed for this project consists of three canvases [1]. The top canvas contains heading of the image, the left sided canvas consists of all the widgets necessary to apply transformations on the image, along with a button to load an image from any folder, and a button to store the final image.

[1]canvas is used in tkinter to display image as well as to hold various widgets like buttons, labels, entry, scale bar, etc.

Fig. 1. Screenshot of GUI.
Canvas background image https://wallpaperaccess.com/glossy-desktop

A. GUI features

As evident from the figure 1 , the image to be processed is displayed in the main canvas, whereas the buttons are displayed in left canvas. The reason why canvas is used here in place of frame is that background image can be added to the canvas in order to make the GUI more attractive.

- Top canvas: This canvas contains the heading for the image editor.
- Left canvas: This canvas is the heart of the entire GUI. All the widgets present in this canvas control the operations to be applied on the image.
- Open Image Button: This Button is used to open file browser in order to select the image.
- Labels and corresponding widgets:
  - Algorithm Label: This label points to a drop down menu as shown below. The drop down menu consists of a list of algorithms which can be applied in order to process a given image. The reason of using a drop down menu is that in order to add a new algorithm, rather than creating a new button every time, the name of algorithm can be easily added to the drop

down menu, thus reducing the number of buttons to be used.
- Gamma value label: This label points to an entry widget, used to take input from the user about the gamma value that needs to be applied.
- Smooth value label: This label points to a scale widget, this widget can take value from 1 to 25 which corresponds to the size of filter used for filtering the image. Since filter size needs to be of odd order, 1 corresponds to a filter of size 3x3, 2 corresponds to a filter of size 5x5 and so on.
- Sharpen value label points to a scale widget which can take value between 0 to 10, 10 being the highest possible sharpening of image. This value directly corresponds to the percentage of Laplacian filtered image to be added to the original image.
- Apply Filter button is used to apply the filter selected in the drop down menu to the currently displayed image in the canvas.
- Save, Undo last change and Undo all changes buttons are used to perform the respective operations of saving the currently displayed image in canvas.

III. Image processing operations

The following list of Image processing operations are carried out in this project.

1) Histogram Equalization
2) Logarithmic Transform
3) Sharpening Image
4) Blur Image
5) Gamma Correction
6) Sketcher

A. Histogram Equalization

Histogram equalization [5] technique is used in order to distribute the probability density function of the original image uniformly throughout the entire intensity range. It can happen that sometimes, the image can have a large number of pixel values concentrated between a small range of pixels. Histogram equalization technique takes such images, and redistributes their probability density function to make it uniform.

$$p_x(r) = p(x = r) = \frac{n_r}{n}, \quad 0 \leq i < L \tag{1}$$

$$cdf_x(r) = \sum_{j=0}^{i} p_x(x = r) \tag{2}$$

$$s = T_r = (L - 1)\ cdf_x(r) \tag{3}$$

The above equations are used for histogram equalization. Firstly, probability distribution function of r is calculated, which is followed by obtaining cummulative distribution function of r. Multiplying $cdf(r)$ with $(L - 1)$ gives the corresponding value of s. The value of s corresponds to the new value of pixel value in equalized image

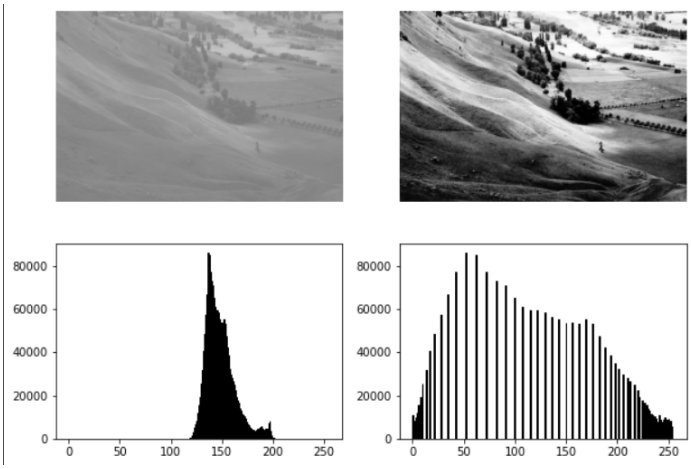

Fig. 2. Histogram Equalization

As shown in figure 2, the distribution of pixel values after application of histogram equalization on left side image produces a uniform distribution.

B. Logarithmic Transform

Logarithmic Transformation [3] is used to distribute the darker pixel values over a wider range. The simple equation for calculating log transform of an image pixel r is given by the following equation.

$$s = clog(1 + r) \tag{4}$$

c is a constant term needed to arrange the logarithm value between 0 and 255. The application of logarithmic transform on the image shown in 3 maps the dark intensity pixels over a wider range, as evident from the histogram of both the images.

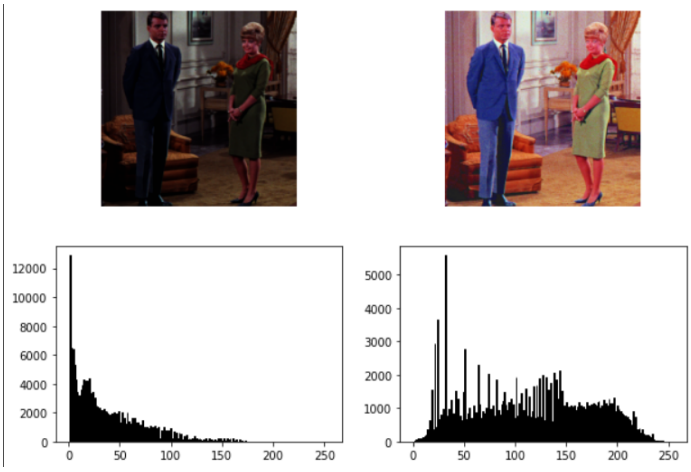

Fig. 3. Log transform

C. Sharpening Image

Sharpening [7] an Image is done using various edge detection filters, like Laplacian filter, Sobel filter, Robert filter, etc. These filters work on the principle of finding the gradient of the given image. If an edge exists, the gradient will be maximum. Sobel edge detection methodology consists of two filters, one to detect horizontal edge and one to detect vertical edge. The Sobel filters are mentioned below.

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The first matrix is used to detect edges in vertical direction and second filter is used to detect edges in horizontal direction. These filters are convolved with the image to produce two components namely $G(x)$ and $G(y)$.

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2} \tag{5}$$

The Sobel filter works on the principle of first derivative. The other filter, Laplacian filter is used in this project, in order to reduce computation due to the use of only one matrix. The Laplacian filter works on the principle of second derivative.

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{6}$$

In discrete form, the above equation can be written as follows.

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y) \tag{7}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y) \tag{8}$$

$$\begin{aligned} \nabla^2 f(x,y) =& f(x+1,y) + f(x-1,y) - 4f(x,y) \\ &+ f(x,y+1) + f(x,y-1) \end{aligned} \tag{9}$$

The following operators are used in order to process an image using Laplacian filter. Since Laplacian filter is isotropic, the image in convolved with only one of the given filters in order to produce the edge detected image. Let $f(x,y)$ be the intensity of a pixel whose coordinate in the image is given by (x,y)

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The image obtained after convolving through the Laplacian gives out edge detected image. The Laplacian of the image is added with the original image to obtain sharpened image.

$$g(x,y) = f(x,y) + c\nabla^2 f(x,y) \tag{10}$$

Here, c is a constant that controls the amount of edge detected image to be added to the original image. $g(x,y)$ is the obtained pixel intensity value for the given pixel $f(x,y)$. The image in 5 shows the effect of sharpening using $c = 1$. The result is enhanced and sharpened edges.

Fig. 4. Sharpening $c = 10$

Fig. 5. Blurring using averaging filter of size 15x15

D. Blur Image

Blurring [4] an Image is carried out by use of averaging filter or Gaussian filter. In this project, averaging filter is used in order to find the average the pixel and its neighbourhood. The amount of blurring depends upon the size of filter, more the filter size, more the pixels which will be averaged and thus more will be the blurring. Averaging filter of size N is an NxN matrix having each element as 1, and all pixels are divided by the total filter size.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Gaussian filter [4] is an example of non linear smoothing filter whereas averaging filter is an example of linear smoothing filter. Gaussian filter is given by the following equation.

$$h(x,y) = e^{-(x^2+y^2)/2\sigma^2} \tag{11}$$

In the above equation x and y correspond to the distance of a given pixel from the central pixel. The central pixel will have (x,y) as $(0,0)$. For a general 3x3 size, the Gaussian filter is given by the following matrix.

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

The matrix is divided by 16 since the sum of all elements add up to 16. 5 shows blurred image using an averaging filter of size 15x15.

E. Gamma Correction

Gamma correction [6] for a pixel value r is given by the following equation.

$$s = cr^{\gamma} \tag{12}$$

The value of γ can be anything greater than 0. For γ less than 1, the darker pixels are distributed over a wider range of dark pixels. For γ greater than 1, the brighter pixels are confined to a narrower region. This is clearly evident from figure 8. A comparison between histograms for all the gamma corrected images in figure 6 is shown in figure 8.

Fig. 6. Gamma correction with $\gamma = 0.25, 1.5, 5$

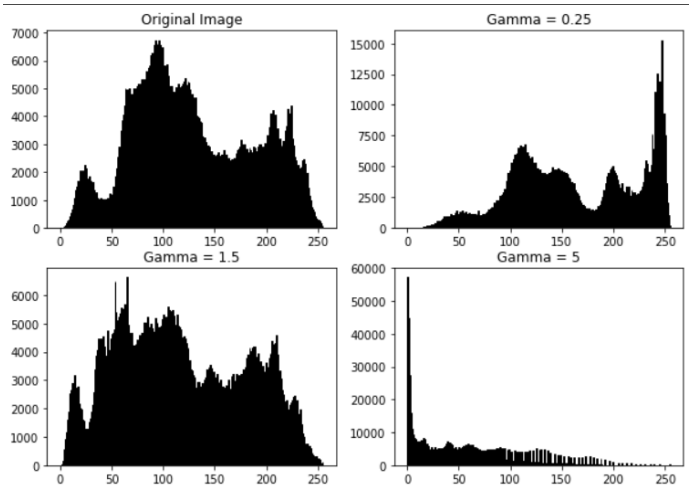

Fig. 7. Histograms for images in figure 6 for $\gamma = 0.25, 1.5, 5$

F. Sketch

For this operation, three processes have been used. First operation is sharpening the image using the sharpening algorithm, followed by edge detection using Sobel filter. Now the output of Sobel filter is inverted and binarized with a threshold of 40 to obtain the image shown below.

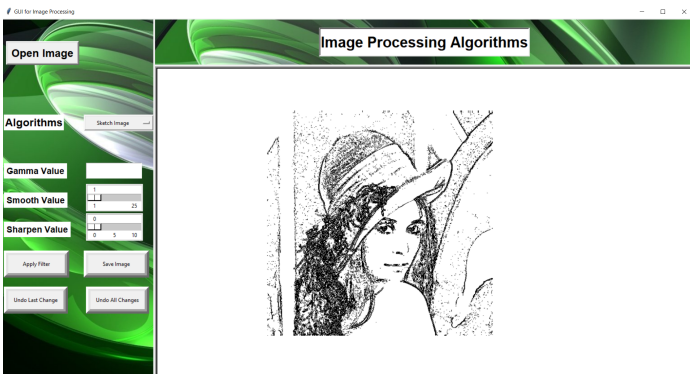

Fig. 8. Image after binarization

IV. Experiments and Results

The experiments on different images have been shown in section sec2. The following diagram represents image enhancement of an image in figure 9, first of all we apply image sharpening technique to make the edges more clear as shown in figure 10. Since the histogram of the image has more pixels in a narrower range, we apply histogram equalization. This is shown in figure 11. Finally, as shown in figure 12, gamma correction with a value of 2 is applied to finally get an enhanced version of the image.

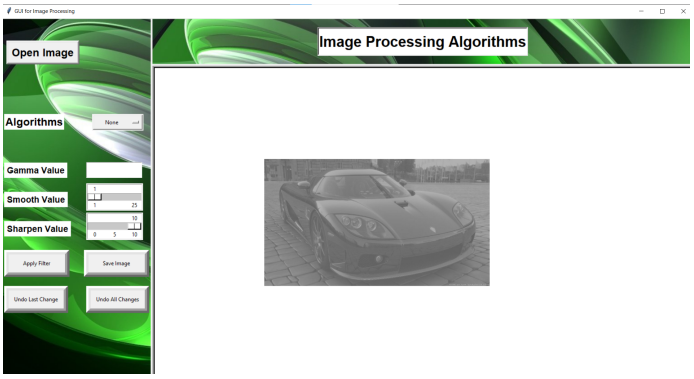

Fig. 9. Original image

V. Conclusion and Discussion

The project mentioned here has been implemented in order to provide a GUI to the user to apply various transformations on any image. Various challenges such as making sure that correct file (image file and not any other file) is loaded by providing an error message box, providing undo and undo buttons have been implemented to ensure a smooth user experience. The major challenge to overcome in this project was to implement the algorithms in such a way that the time required for image processing is not higher. This was carried out by the use of vectorization in python, where image was broken down in NxN matrices, stacked over each other to

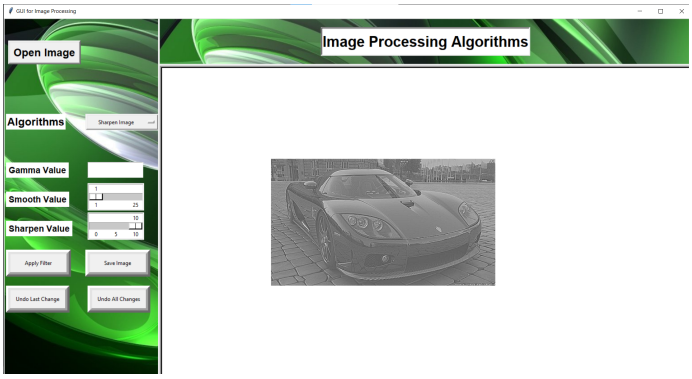

Fig. 10. Sharpened image

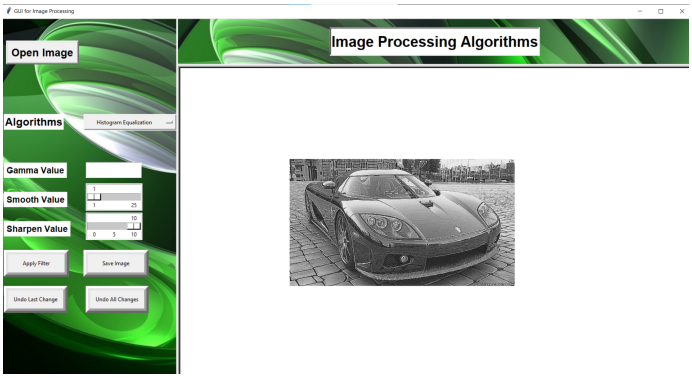

Fig. 11. Histogram equalization

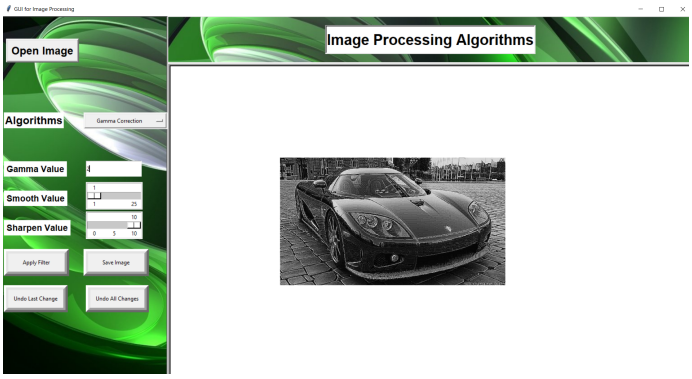

Fig. 12. Gamma correction with $\gamma = 2$

create a giant PxNxN matrix, where P = total pixels in the image, followed by convolution with another giant filter matrix of dimension PxNxN.This lead to a significant improvement in time but for large sized filters, the problem of memory shortage pertains. To solve this, a second convolution function is defined which convolutes the image with a given filter of size N by dynamically creating NxN matrix with a given pixel at the centre, and remaining elements of the matrix correspond to the neighbourhood pixels. Due to the use of drop down list in the GUI, it becomes easier to accommodate new algorithms for transformations to be applied on the image. If time permitted, it could have been of interest to implement frequency domain filtering for images as well.

REFERENCES

[1] Graphical user interfaces with tk. Available at https://docs.python.org/3/library/tk.html.

[2] B.Sridhar. Applications of digital image processing in real time world. *International journal of scientific and technology*, 8, 2019.

[3] Geeks for Geeks. Log transformation of an image using python and opencv. Available at https://www.geeksforgeeks.org/log-transformation-of-an-image-using-python-and-opencv/.

[4] R C Gonzalez and R E Woods. *Digital Image Processing*. Pearson Educational International, 3 edition, 2008.

[5] Jasdeep Kaur Kaur, Manpreet and Jappreet Kaur. Survey of contrast enhancement techniques based on histogram equalization. *International Journal of Advanced Computer Science and Applications*, 2011.

[6] Francis G. Loch. Image processing algorithms part 6: Gamma correction. Available at https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-6-gamma-correction/.

[7] Ganesan P and G.Sajiv. A comprehensive study of edge detection for image processing applications. *International Conference on Innovations in information, Embedded and Communication Systems (ICIIECS)*, 2017.