

# Image Processing

Garima Dewangan

19D070023

IIT Bombay

**Abstract**—The paper describes the GUI developed for image processing and enhancement which are blur, sharpen, histogram equalization, log transform and gamma transform.

**Index Terms**—Image, Processing, Histogram Equalization, Blur ,sharpen

## I. INTRODUCTION

The objective of this paper is to build a GUI that can support various image processing operations namely :-

- 1) Histogram Equalization
- 2) Gamma correction
- 3) Log transform
- 4) Blur
- 5) Sharpen
- 6) Binarize
- 7) Edge Detection

The GUI is implemented using PySimpleGUI library of python.

The code consist of various function definitions to perform the above functions. It also contains the code to implement the GUI.

## II. GUI DESIGN

GUI was build using the PySimpleGUI library of python. I went through the documentation and some examples to learn about the various functions available.

GUI Features

On running the code a window opens.

The theme used is Dark Amber.

It has a browse option which allows the user to select the image from the computer.

After clicking select button the image is displayed on the right side of the window.

The left side of window contains button to perform various functions like sharpen, blur etc.

Various functions ask for parameters as input from user.

It offers other function including UNDO, RESET and SAVE

intensities in the image accordingly.

### 2) Gamma Correction

- The new values for all the intensities are calculated according to gamma :  $I' = k * I^\gamma$

- The values of intensities were updated accordingly

### 3) Log Transform

- The new values for all the intensities are calculated according to :  $I' = k \log(I + 1)$

- The values of intensities were updated accordingly

### 4) Blur

- Based on the filter size and filter type selected by the user the filter is initialized.

- Then the image is padded with zeros to maintain its size

- Then the padded image is convolved with the filter

- The output is displayed on the window

### 5) Sharpen

- The image is first blurred with a mean filter of size 5.

- Then the blurred image is subtracted from original image

- The diff is added to the original image

### 6) Binarize

- All the pixel values below the threshold (taken as input from the user) is set to 0 while others are set to 255

### 7) Edge Detection

- Used Sobel Operator.

- Convolve the image using vertical and horizontal edge filters. Then the rms of the two values is compared to a threshold.

- If greater than threshold then we set intensity to 255 else 0

- Then Image is blurred using a mean filter of size 5.

## III. IMAGE PROCESSING OPERATIONS

### 1) Histogram Equalization

- We calculate the pdf of all intensities in the image by calculating the sum of the no. of time the intensity occurs by the total no. of pixels.

- Then we calculate the cdf by cumulatively adding the pdf values.

- Then we scale the cdf by multiplying it with 255. This gives the new pdf of intensities. And we change the values of the

## IV. EXPERIMENTS AND RESULTS

### A. Histogram Equalization



Fig. 1. Original Image



Fig. 2. Equalized Image

### B. Gamma Correction



Fig. 3. Original Image



Fig. 4. Corrected Image

### C. Blur



Fig. 5. Original Image



Fig. 6. Blurred Image

### D. Binarize



Fig. 7. Original Image

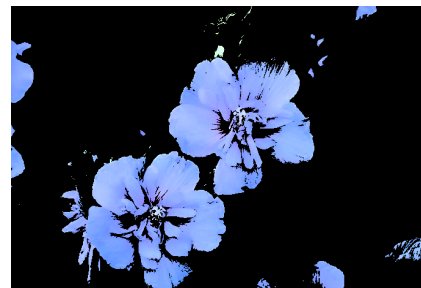


Fig. 8. Binarized Image

### *E. Sharpen Image*

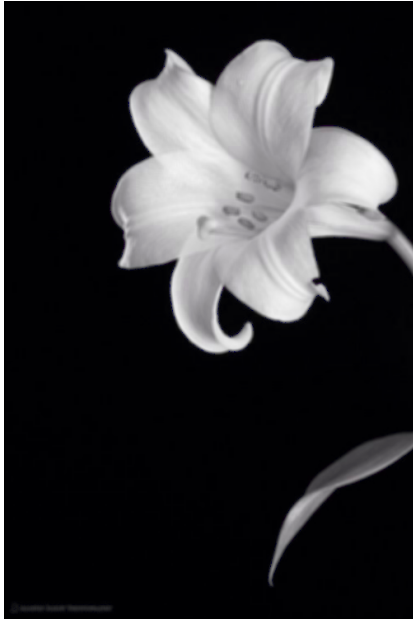


Fig. 9. Original Image

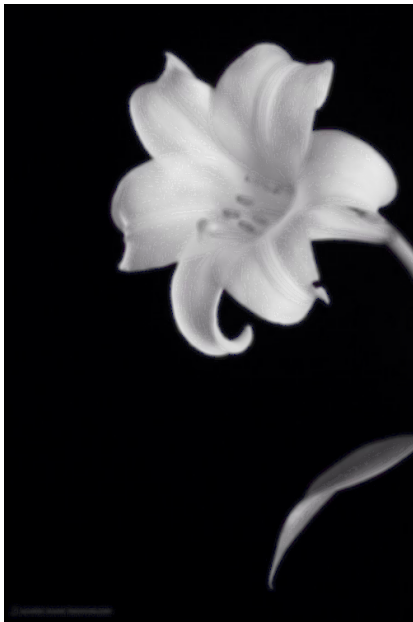


Fig. 10. Equalized Image

### REFERENCES

### REFERENCES

- [1] <https://pysimplegui.readthedocs.io/en/latest/>
- [2] [https://numpy.org/doc/stable/user/absolute\\_beginners.html](https://numpy.org/doc/stable/user/absolute_beginners.html)
- [3] <https://docs.python.org/3/library/tkinter.html>
- [4] <https://www.blog.pythonlibrary.org/2021/02/16/creating-an-image-viewer-with-pysimplegui/>
- [5] [https://docs.opencv.org/master/df/d65/tutorial\\_table\\_of\\_content\\_introduction.html](https://docs.opencv.org/master/df/d65/tutorial_table_of_content_introduction.html)