

Basic Image Editor

EE 610: Assignment 1

Chinmay Bharti

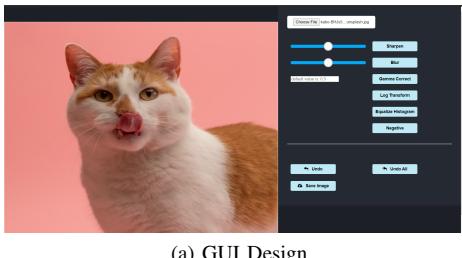
Roll no.: 18D070043
Electrical Engineering, IIT Bombay
Email: 18d070043@iitb.ac.in

Abstract—This report contains information about implementation of 'Basic Image Editor'. The report has 6 main parts. The 'Introduction' part covers the Objectives and main part of the assignment. The 'GUI design' part depicts the standard UI blocks that are used for the application. The 'Image Processing Operations' part covers the main equations used for this application. The 'Experiments and Results' part contains the demonstration of Image Editor. The 'Conclusion' part covers the inference and problems faced while developing the application. The 'Reference' part lists all the references.

I. INTRODUCTION

The 'Basic Image Editor' is coded as a web application. Python, HTML, CSS and JS are the 4 programming languages or typescripts that are used in this application. The Frontend consists of HTML, CSS and JS, While in the backend Python Django framework has been used. The Image transformations like 'Sharpening the Image', 'Blurring the Image', 'Equalization of histogram', 'Log Transformation', 'Gamma Correction' and 'Negative of the Image' are implemented. Other then these transformations, some utility functions like 'Undo', 'Undo All', 'Save Image' are also implemented.

II. GUI DESIGN



HTML and CSS are the type scripts that are used to build the frontend GUI Blocks. A Canvas Component is used to display the image. There are 3 buttons at the bottom namely Undo, Undo All and Save functions which are succeeded by a horizontal line. There are 2 Sliders which are giving for controlling the extent of blur and sharpening. There is an input box which is given to input the gamma correction value. Other than that there are 6 buttons which are given for each Image Processing Operation. This whole configuration is represented as a table.

JQuery Library in JavaScript is used as a middleware to connect frontend and backend. Each form submit event prevents the default page refresh and asynchronously sends the data to backend and then fetches the processed image. This fetched Image is displayed on the canvas.

III. IMAGE PROCESSING OPERATIONS

Following Image Processing Operations are used in this 'Basic Image Editor':

- **Sharpen:** This Algorithm is used to sharpen the image i.e. it enhances the edges of a blurred image and makes it more clearer. Laplacian mask is used to implement the sharpening functions. The Laplacian mask used is as follows:-

$$\begin{bmatrix} -1 * scale & -1 * scale & -1 * scale \\ -1 * scale & +8 * scale & -1 * scale \\ -1 * scale & -1 * scale & -1 * scale \end{bmatrix}$$

Here the scale represents the intensity of the sharpening. This Mask is convolved on the intensity channel of the image. After re-merging the image we got back the sharpened image.

- **Blur:** This Algorithm is used to blur the image. Average pooling filter is used to achieve the blurring algorithm. The intensity of blurring is controlled by the size of the filter. The filter looks as follows :-

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1filter} \\ x_{21} & x_{22} & \dots & x_{2filter} \\ \vdots & \vdots & \ddots & \vdots \\ x_{filter1} & x_{filter2} & \dots & x_{filterfilter} \end{bmatrix}$$

All the values inside the filter are same and are equal to $1/(filter-size*filter-size)$. This Mask is convolved on the intensity channel of the image. After re-merging the image we got back the blurred image.

- **Gamma Correction:** Gamma Correction is used to control overall brightness of the image. The formula used for gamma correction is as follows:-

$$Image_{Corrected} = 255 * (Image/255)^{\text{gamma-value}}$$

Here the Gamma Value is taken as an input. This operation is carried out only on the intensity channel of the image. After re-merging this channel we get the gamma-corrected image.

- **Log Transform:** Log transformation is used to make highly skewed intensities less skewed. The formula used for log transformation is as follows:-

$$s = c * \log(r + 1)$$

Here s is the output pixel value and r is the input pixel value. The value of c is 255/maximum-pixel-intensity. This operation is carried out only on the intensity channel of the image. After re-merging this channel we get the Log-transformed image.

- **Equalize Histogram:** Histogram equalization is used to adjust the contrast of the image using histogram of the image. The formula used for Histogram equalization is as follows:-

$$p_n = \frac{\text{number - of - pixels - with - intensity - } n}{\text{total - number - of - pixels}}$$

$$g_{i,j} = \text{floor}((L - 1) * \sum_{n=0}^{f_{i,j}} p_n)$$

This operation is carried out only on the intensity channel of the image. After re-merging this channel we get the Log-transformed image.

- **Negative:** Negative of an image is used to replace bright areas with dark areas and vice-versa. The formula used for this is as follows:-

$$y = 255 - x$$

Here x is the input intensity and y is the output intensity. This operation is carried out only on the intensity channel of the image. After re-merging this channel we get the Log-transformed image.

IV. EXPERIMENTS AND RESULTS

A. Sharpen:

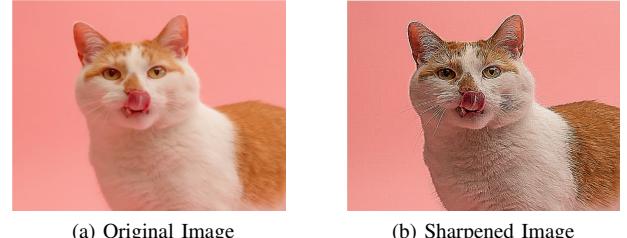


Fig. 1

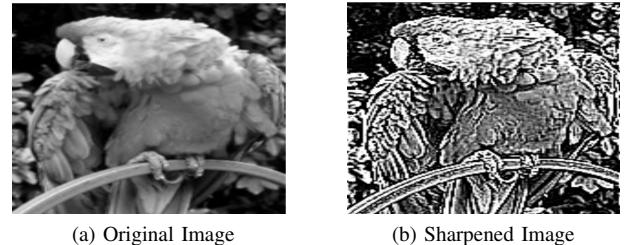


Fig. 2

B. Blur:

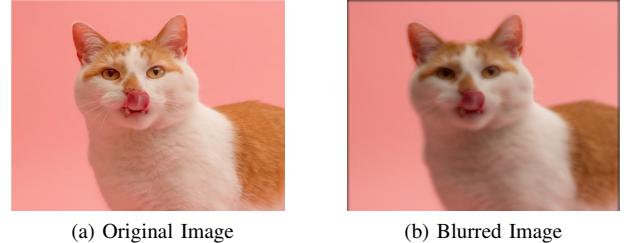


Fig. 3

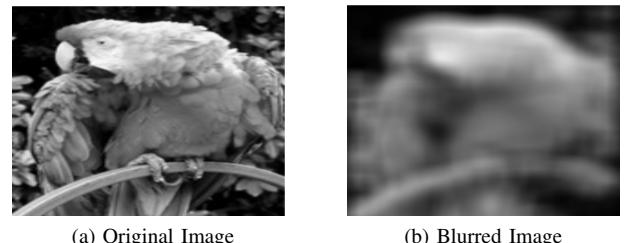


Fig. 4

C. Gamma Correction:



(a) Original Image



(b) Gamma Corrected Image

Fig. 5



(a) Original Image



(b) Histogram Equalized Image

Fig. 10



(a) Original Image



(b) Gamma Corrected Image

Fig. 6

D. Log Transform:



(a) Original Image



(b) Log Transformed Image

Fig. 7



(a) Original Image



(b) Log transformed Image

Fig. 8

E. Equalize Histogram:



(a) Original Image



(b) Histogram Equalized Image

Fig. 9

F. Negative:



(a) Original Image

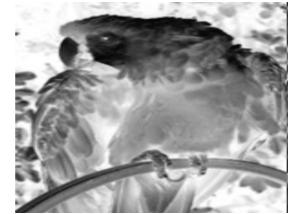


(b) Negative Image

Fig. 11



(a) Original Image



(b) Negative Image

Fig. 12

V. CONCLUSION AND DISCUSSIONS

The biggest challenge was to implement the convolution function. We first have to zero-pad the image then apply the convolution operation. Given more time I would like to make a complete open source image editor which has all the basic image processing functions.

REFERENCES

- [1] Gamma Correction, Wikipedia
- [2] Histogram Equalization, math.uci.edu. math77c course
- [3] Log transformation, geeks for geeks
- [4] Sharpening, bohr.wlu.ca, cp467 course, lecture 6
- [5] Blurring images, datacareentry.com