EE 610 Assignment 1: Making Basic Image Processing Tool

Sachin Goyal, 150020069, EE UG

***Abstract*— Playing with images to make them look more better and to extract more information from them is one of the major research topics. This reports explains in detail the Basic Image Processing Tool made by me as a part of the EE610 assignment. The reports contains description of all features present in the tool and shows some results on selected images.**

I. INTRODUCTION

Images have been a topic of research from long. We generally capture images using cameras of varying quality and moreover not everyone is perfect in handling the DSLR's or the so called upper range cameras. Many a times it is the external factors like the lighting conditions which govern the quality of the image taken and little can be done to improve the image quality via improving the camera quality. Lately, we have now a days become interested in trying to visualize the captured images in multiple lighting conditions. With the boom of many image based apps like instagram, interests and the requirements in this field have increased. Also, many a times image enhancement is necessary to extract more information from the captured image specially in cases of astro-photography. Hence I developed a basic image processing tool as a part of the assignment. The tool has features like blurring images, contrast matching, sharpening of images and embrossing the image (an extra featue). Sometimes in astro-photography we need the requirement of blurring the images so that we can focus on the major stars easily and precisely. Blurring almost makes the small stars a part of the background of the image thus easing the tasks of scientists and at the same time giving more clarity. Contrast has been improved by using transform like histogram equalization, gamma power law transforms, log transforms, etc. We have implemented 2 types of blurring , the gaussian and the normal blurring using ones. Sharpening has been done using unsharp masking. The GUI for this tool was made using PyQt5.

II. BACKGROUND READ

This assignment required intensive knowledge of concepts like histogram equaltization, filters, convolutions and image sharpening. I did an extensive reading of the book *Gonzalvez and Woods* for being thorough with these concepts. Prof. Amit Sethis slides were the primary source of learning.
The task also involved making a GUI framework using python. I used PyQt5 for developing the GUI and hence had to go through PyQt tutorials for learning the same. Some wikipedia searches also helped for learning how to resolve issues of negative pixels. CV2 has been used for tasks of loading images. I had a good knowledge of numpy, else a new person may need to go through some of the very good mathematical functions provided by the library. The last reference section gives links of some of the important resources referred

III. APPROACH

The code consists of 2 python files GUI and GUIfunctions. The functions file deals with all the mathematical calculations like the one in histogram equalization, power law transforms and convolution implementation. The GUI file has the framework code for the user interface implementaion and the various image form conversation.

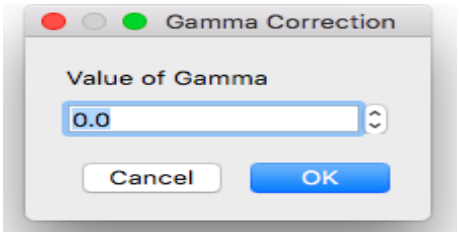
A. GUI

As stated earlier I used PyQt5 for developing the GUI. The GUI is basically a grid of 8 rows and 2 columns. Using the grid feature of pyqt easies out the task of positioning the canvas and the buttons since it divides the screen into subparts. To display the images, pyqt has its own pix based method but I had a couple of issues in using it. Hence I switched to integrating matplotlib canvas to the pyqt displaying the results of all the available transforms side by side of the original image.

B. Features Implemented

The features implemented deal with blurring of the image, enhacing the contrast of the input image, sharpening of the image and some redo and undo options. Let us look each of these features in detail.

1) **Gamma Correction**
Gamma correction provides us with a lot of flexibility in scaling the intensity levels in the image. Gamma transformation amounts to applying a non-linear transform to the image intensity. The shape of the transform will depend on the gamma value provided as input. A value less than 1 in lay man terms brightens the image whereas a value grater than 1 darkens it. On clicking this button the Tool will ask the user to input a gamma value of his choice and display the result hence-after.



2) **Log Transform**
Its a standard tool for intensity transform of images. Applies a log function to the intensity values of each pixel.

3) **Histogram Equalization**
   Histogram equalization is used to spread the most frequent intensity values in the image equally over the whole range of intensity. This contrast enhancement technique can be used to enhance both dark and the light images. Using histogram equalization one can essentially match histograms of images to that of any other desired image or can achieve one of desired shape. The output on clicking this button will also show the histograms of both the input and the resulting output image.
4) **Blur Image**
   This buttons blurs the input image using a kernel of ones and of size as input by the user.
5) **Gaussian Blur**
   This uses a Gaussian kernel instead of the ones kernel in the previous feature. The Gaussian kernel is a zero mean kernel of size and variance as given in the input by the user. Kernel is generated as per the following standard gaussian equation

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

6) **Image Sharpening**
   I have implemented image sharpening using Unsharp Masking. Image sharpening is one of the most used tools to enhance the fine details in the images. Unsharp masking uses a blurred version of image and subtracts it from the original to get a mask. This mask is then added to the image to get an approximately sharpened image with less blurring. The user can input a constant which determines the weight of the mask in the final output image.
7) **Emboss**
   Image embossing is used to create a illusion of a paper or metal embossing of the original image[2]. Each pixel is either replaced by a highlight or a shadow depending upon the boundaries of the original image.
8) **Undo, Save and Restore**

IV. Template Results on Selected Images

In this section, let us see the results of the available features in the tool on some sample test images.

1) Gamma Correction
   We can see in the Fig. 1. An image of a night seen was gamma correction with gamma = 0.4. The output image intensity levels are clearly way more brighter giving a illusion of an early morning or evening scene. Fig2 shows results of transform with gamma = 4.
2) Log Transform
   Results on the image of a boy in a dark background can be seen in Figure 3.
3) Histogram Equalization
   This transform gives some very appealing results as one can see in the Fig4. The input is an image of a room captured in low light conditions. Very less information is conveyed by the input image regarding the items on table and about the 2 people. The histogram equalized image greatly increases the amount of information available.
4) Blur Image
   Fig5 shows the effect of gaussian blurring on a colored cat image with kernel size of 5 and variance of 4.
5) Image Sharpening
   I have used Lena Image to show the results of unsharp masking based sharpening.
6) Image Embossing Some interesting image embossment can be seen in Fig 8

(a) Original (b) Result

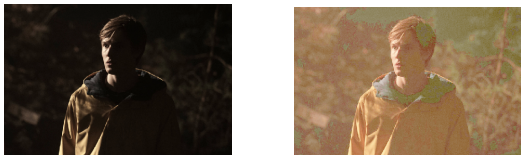
Fig. 1: Gamma Correction Example(Gamma = 0.4)

(a) Original (b) Result

Fig. 2: Gamma Correction Example(Gamma = 4)

V. Discussion

One of the most difficult part of the assignment was implementation of image sharpening. There are abundant number of approaches for image sharpening each of which works on a different set of images. It took me lot of time to get rid of artifacts and some random noisy type output images in sharpening. That was mainly happening due to the issue of negative pixels since the kernel in image sharpening has lot of "-1" value elements. These neagative valued pixels then had to be clipped to 0. Another issue was that due to image sharpening , the image was getting overall darker since I earlier added the negative image pixel value to the whole image to bring it down to 0. This problem was also resolved using clipping. Apart from this, if I had more time I would have liked to implement many cool image processing features based in frequency domain. Frequency domain allows you

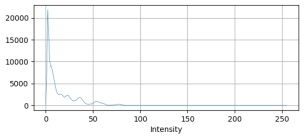


(a) Original (b) Result

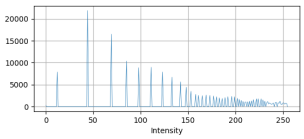
Fig. 3: Log transform Example

(a) Original (b) Result

Fig. 4: Histogram Equalization Example



(a) Original Image Histogram



(b) Equalized Histogram

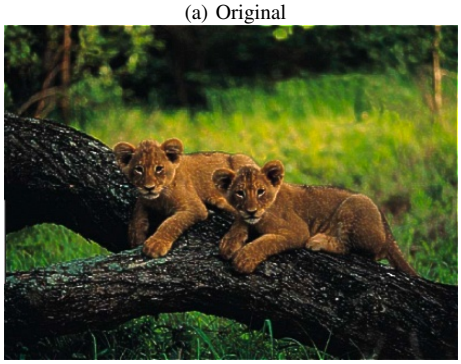
Fig. 5: Corresponding Histograms

(a) Original Image

(b) Blured Image

Fig. 6: Blurring in Images

(a) Original



(b) Result

Fig. 7: Image Sharpening

(a) Original

(b) Result

Fig. 8: Image Embossment

to do a whole different range of operations on the images. I was also lately trying to implement some basic image super resolution using bilinear interpolation which is also the current topic going on in the lecture.

VI. References

1) Gonzalvez and Woods, for learning all the basics about the different filters, spatial and frequency domain filtering
2) Studying about image Embos and its applications : $https://en.wikipedia.org/wiki/Image_embossing$
3) $http://setosa.io/ev/image-kernels/$ for Emboss