INTENSITY TRANSFORMATION

TEAM STARKS

Suchismitha Vedala - 1470929
Yashwanth Reddy - 1644186
Lavanya.S.S - 1616056
Naveen - 1637404

INDEX

INTRODUCTION

Intensity Transformation-

An image is defined as a two dimensional function f(x,y) where the amplitude of “f” at any pair of coordinates (x,y) is called the intensity of an image. The process of modifying the intensity values of an image for different applications is called as intensity transformation. We may want to reverse black and the white intensities [image negatives] or we may want to make the darks darker and the lights lighter. An application of intensity transformations is to increase the contrast between certain intensity values so that you can pick out things in an image. In general, Intensity Transformation is the process of mapping a Transfer Function T to each pixel(R) of the image to get new intensity values(S),

$$S = T(R)$$

OBJECTIVE

The goal of the project is to perform Intensity transformation of images. Various operations have to be performed to obtain the objective. Image negatives, Power Law transformation, histogram matching and histogram equalization operations are the operations considered to enhance the intensity of the images in this project. To visualize the transformation, we need to design a User Interface. The User interface should allow to upload an image and let the user choose an operation that is used to enhance the given image. The objective is to let user visualize the input and the enhanced image that lets the user to compare the outputs. The project is intended to perform operations on both gray scale and color images.

IMPLEMENTATION

While Implementing the project, the first aim was to develop the User Interface. Various technologies have been explored while designing the User Interface . The project is developed using Django, HTML , JavaScript and Python. Django is an Open Source high-level Python Web framework that encourages rapid development and clean, pragmatic design. The webpages of the user interface are developed using HTML and JavaScript. Python is used for performing the image operations. The user interface of our project is divided into two modules:

1. Web Application
2. Image Operations

WEB APPLICATION

Web App is built using Django and Bootstrap templates. It is an MVC (model view controller) Application. All the operations on the image are called from the view function, the operation is performed in python and the result is returned to the view, which is then rendered in HTML. Each operation has a separate URL and View function, which performs the specified operation on the uploaded image.

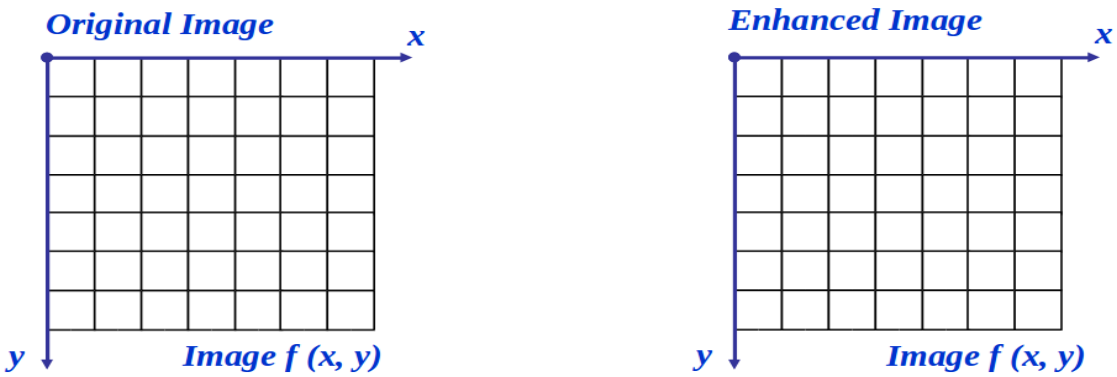
IMAGE OPERATIONS

The following transformation operations are implemented in our project to perform intensity transformation of the given image. Each of the operation is compatible to perform on both gray scale and color images.

Each image operation is in a separate python file, which take the uploaded image as input and return the output. These operations are called from the view function and the result is passed to the HTML

Image Negatives

Negative images are useful for enhancing white or grey detail embedded in dark regions of an image. It is used to enhance medical images, images captured in remote sensing, images from satellite etc. We apply the below formulae for all pixels in the image.
The number of pixels remain same in the original Image and Enhanced Image, but the intensity values change.
For Colour Images we apply the below for each channel i.e R, G and B for every pixel.



$$s = intensity_{max} - r$$

Power Gamma

The gamma is changing the value without changing the intensity of the image. If the gamma value is less than 1 the image will get darker and darker thus the gamma value is reduced correspondingly. As the value of gamma is less than 1 the image will lighter. The gamma value 1 will have no effect on the image as we are not making any change in that. The gamma function which we created take two main argument as parameter first is Input image in which the operation going to perform, then the gamma value we also assign the default value as one if no input is given. The range of gamma function is given in from 0 to 3 in the drag box.

Algorithm:
1. Image pixels intensities are scaled from the range [0, 255].
2. Each pixel of the image is taken then the power gamma function is performed :
O = I ^ (1 / G)
Where I is our input image and G is our gamma value.
3. output image O is then scaled back to the range [0, 255].

Histogram Matching

The histogram matching is the transformation of an image so that its histogram matches a specified histogram. Histogram matching can be used to normalize two images, when those images are acquired at the same local illumination over the same location, but by different sensors, atmospheric conditions or global illumination.

These steps are followed in the process of histogram matching

- Take Two images as inputs, first image is the image to be transformed and the second image is target image

- Convert the images into their respective histogram
- Find out the cumulative distribution function of the histograms, now we have two histograms**,**one is histogram from the image to be transformed and another is histogram of target image
- Shape the histogram of the image to be transformed to that of the histogram of the target image

Histogram Equalization

Histogram Equalization is enhancing the image using the Equalization implies *mapping* one distribution to another distribution so the intensity values are spread over the whole range. To accomplish the equalization effect, the remapping should be the *cumulative distribution function (cdf)*. For the histogram H(i), its *cumulative distribution* function Cdf[i] is:

Cdf[i] = (for 0 <=j<=i; sum(H(j))

Algorithm:
1.Calculate the histogram(H) of the image
2. Calculate the cumulative distribution function(Cdf) of H.
3. Find the largest(l) and smallest(m, greater than 0) number in the Cdf.
4. We equalize the cdf as follows. For every non zero element in the Cdf,

T(i)= (Cdf(i)-m)*255 / (l-m)

5. The enhanced image is thus obtained as:

New_Image = T[Original Image]

The above algorithm is extended to RGB images using three channels.

CHALLENGES :

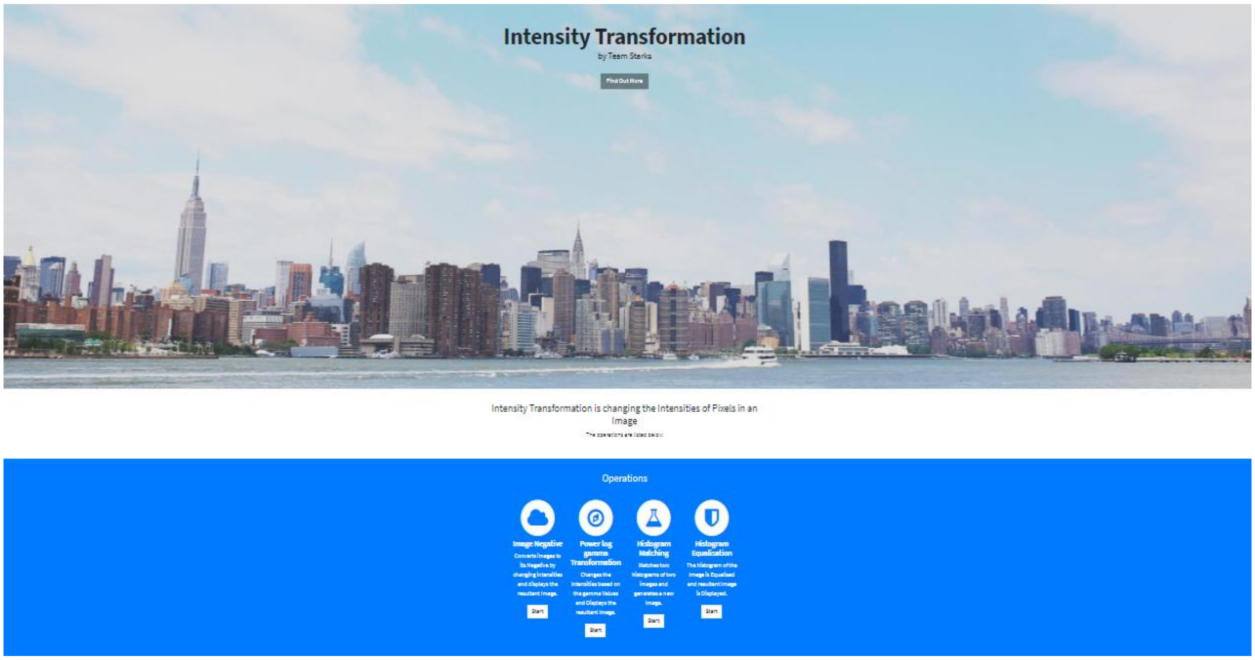
While designing the application, one of the main thing is how do we handle the Images, should they be saved Or Should the computations be done on the Fly and Just display the result.

To perform the operations on the image, we need the image data, but when we upload the image we get the image stream which should be saved on the disk to be read by the algorithm.

So, we are converting the image to base64 from the uploaded stream and using it to convert to array and perform computations and again to display the image, the result is converted to base64 and it is displayed. This is done since HTML supports rendering base64 images.

RESULTS

User Interface



Intensity Transformation

by Team Starks

Intensity Transformation is changing the Intensities of Pixels in an Image

The operations are listed below

Operations

Image Negative

Converts Images to its Negative by changing Intensities and displays the resultant Image.

Power log gamma Transformation

Changes the Intensities based on the gamma Values and Displays the resultant Image.

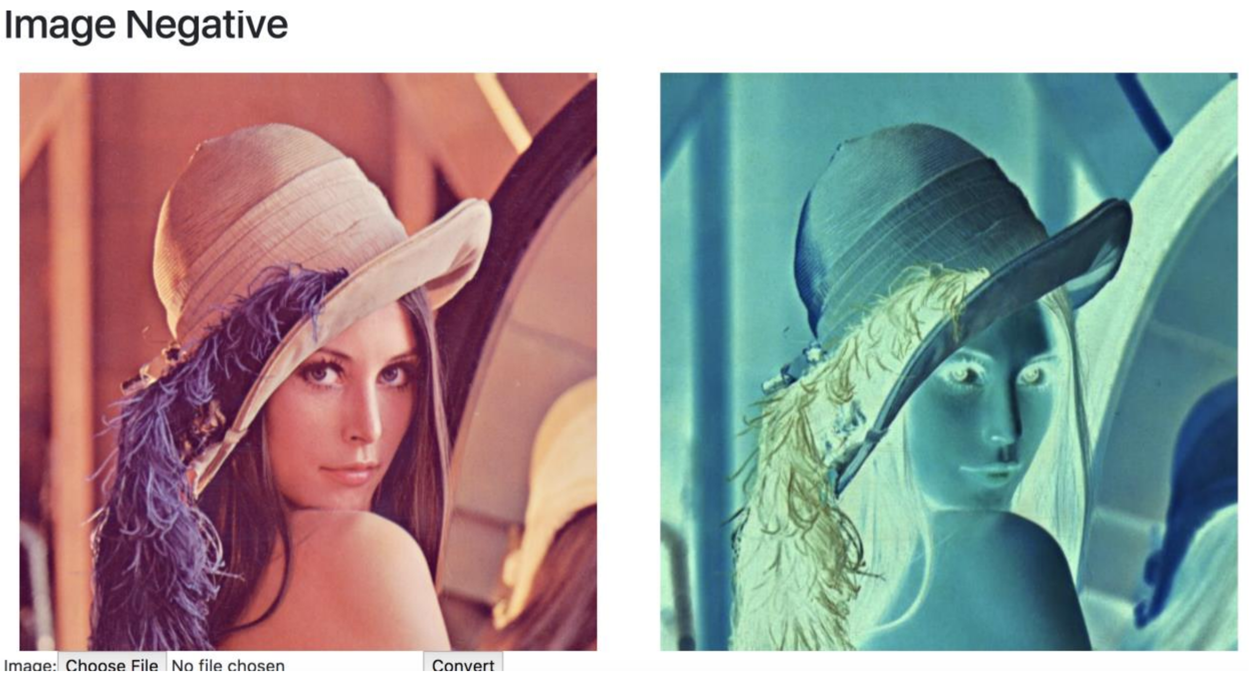
Histogram Matching

Matches two histograms of two Images and generates a new Image.

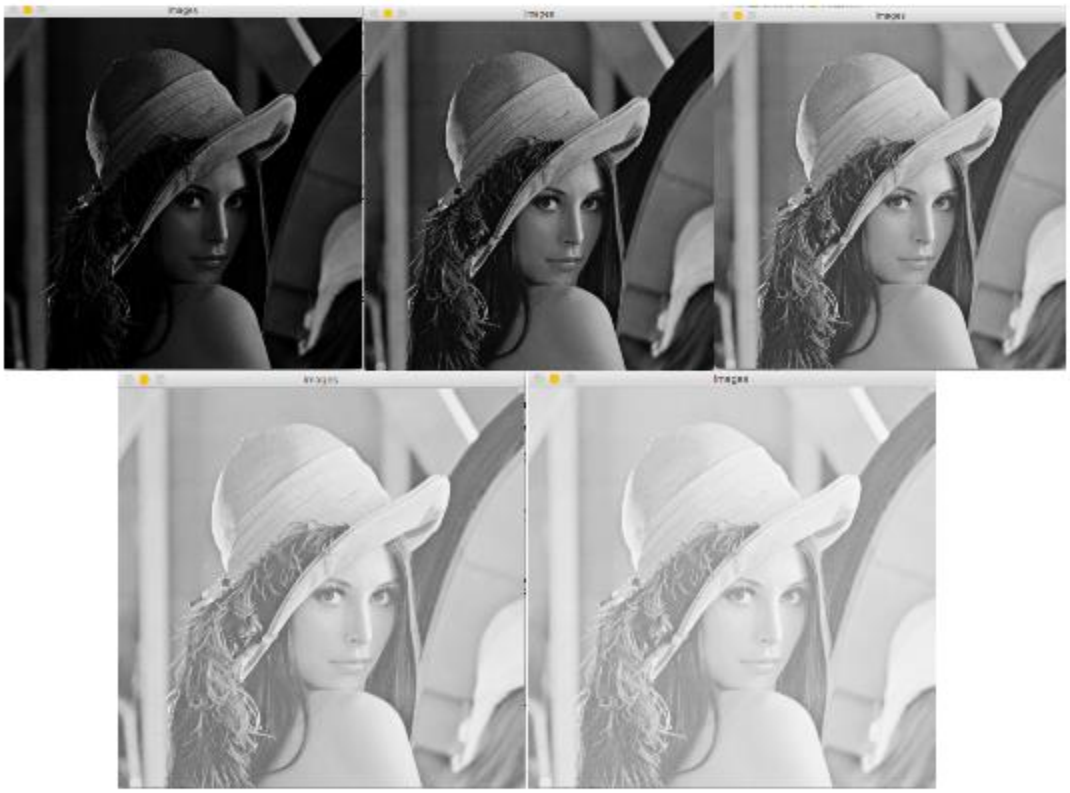
Histogram Equalisation

The histogram of the Image is Equalised and resultant Image is Displayed.

1.image negative

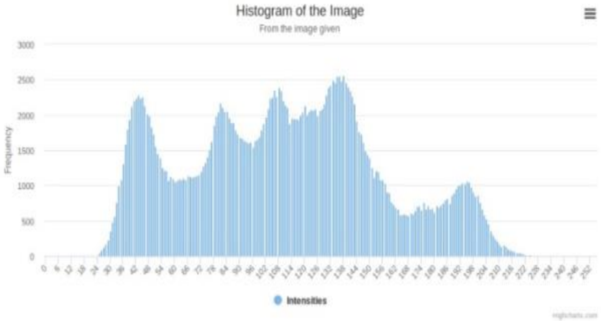


2. power law gamma transformation



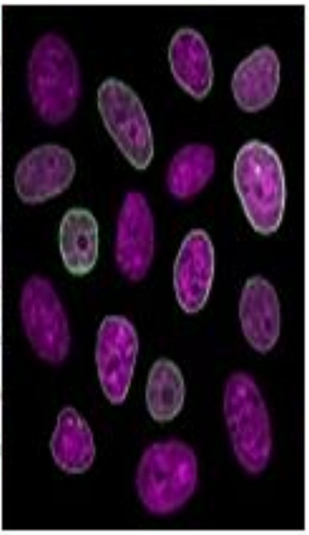
3.image histogram

Image Histogram

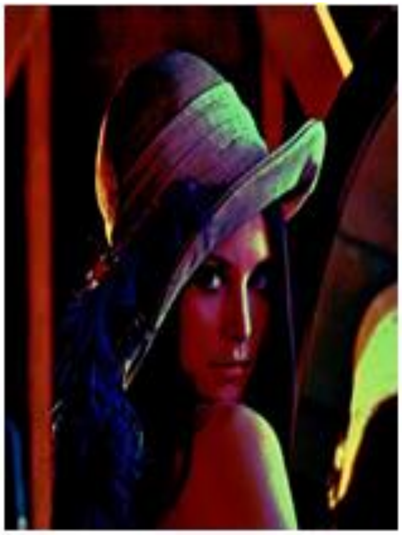


4.histogram matching

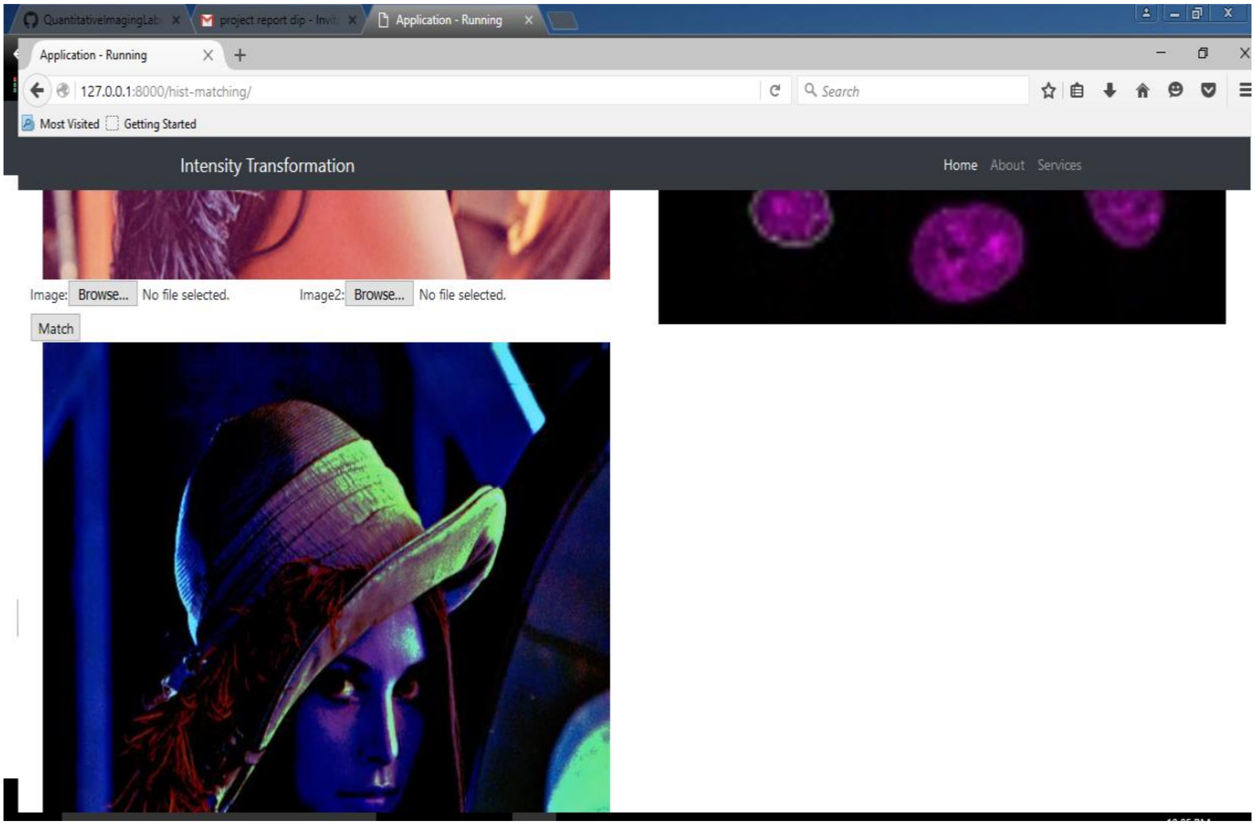
Source image



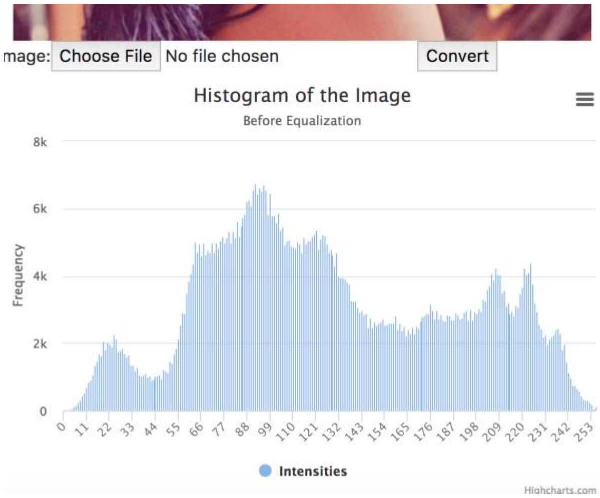
Reference image

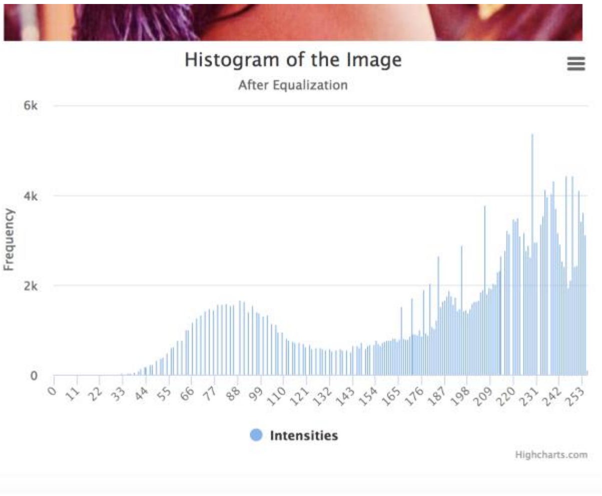


Resulting image



5.Histogram Equalisation





CONCLUSION

The developed project performs the required operation with effective visualization through the web application. The project can be further extended in implementing various other transformation techniques such as histogram shaping.

REFERENCES

- http://www.cs.uregina.ca/Links/class-info/425/Lab3/
- https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html
- https://en.wikipedia.org/wiki/Histogram_matching