**Image Enhancement**

Group Members : Jayant Meshram, Shubham Ghaytadkar, Siddhi Mukkawar

As per our topic goes, Image enhancement, we tried to find out the best approach to enhance the image as there are many. To begin with we started to read about how the intensity of a particular pixel is actually represented in terms of contrast and brightness, what is histogram and how the histogram equalization works and how the gamma parameter affects the pixel intensity by power law.

For our task we have implemented all the methods which we learnt altogether during this course of the task. Namely, we did the Basic linear transformation, Histogram equalization and one with power law transformation.

**1.Basic Linear Transformation:**

So in the Basic Linear transformation we deal with two parameters, alpha and beta, as,

Output Intensity = alpha\*Input Intensity + beta

Alpha and beta representing contrast and brightness respectively. First we were taking the input from the user for alpha and beta value(instructing within specific range) and were changing the intensity using those using inputs. But as it was not most optimal and user friendly solution we decided to make use of trackbar and make a note the values we were finding best for an image and then generalize it. So for a dark image, we often needed the alpha within range of 1.5-3 and beta within range of 30-60. So we come up with a way to calculate alpha and beta automatically for a given image with making use of mean intensity of all the image pixels.(255/img.mean(): would give us suitable alpha most of the time and beta from img.mean()/alpha).

Later as instructed by our mentors, we did this same operation on image but repeatedly on small small parts (5x5 matrix), calculated img.mean() for the same performed linear transformation and slided the matrix all over the image performing operations over and over(as we would do in blurring with use of kernel).

Results we got from both with calculating the whole image’s mean intensity and calculating it with part wise, both were good. But sometimes the it the second one was better than the previous one but it is more required more time for execution than the former.

**2.Histogram equalization:**

So histogram refers to graph and as the name suggest, histogram equalization works towards equalizing the intensity of the pixels. Function cv2.equalizehist() make use of cdf and lower the intensity in the unequally high intervals and increases where it is lowest aiming to get an continuous plotting for intensity.

So as it operates on single channel image, first what we did was we splitted the 3 channel bgr image into 3 separate channels b,g,r, performed equalization over it and merge them together to get the image again.

Result for this were not that good enough, it was exposing lot of contrast in some parts of the image.

So later, as suggested by our mentors we did it by converting it into hsv. We first converted the image into hsv, and as we are only aiming to increase the brightness that is value(v), we splitted the image into three channels h,s and v, performed equalization on v and the merge it together and converted it back into bgr colour space.

Result for this were good than the previous method, where we were equalizing b g and r.

**3.Gamma Correction :**

Gamma correction is also known as the Power Law Transform. First, using table we scale our image pixel intensities from the range [0, 255] to [0, 1.0]. From there, we obtain our output gamma corrected image by applying the following equation:

Output intensity= Input intensity ^ (1 / G)

If the Gamma value is < 1 then the image will darker end of the spectrum while gamma values > 1 will make the image appear lighter.

We tried to implement a way to obtain the suitable gamma automatically but it wasn’t giving the appropriate result always, sometimes the colours were too distorted, but for the same code, with not so dark image, results were okay. So instead making further changes for the same manner we make use of the trackbar ranging values of gamma from 1 to 4, user would just need to set the value suitable as per their convenient.

So with all the methods mentioned above, we tried to implement the given task.

And so here we will end this summarization by briefly reporting the each ones own contribution.

Shubham:

After group task of IP as Image Enhancement get assigned to us we start to finding different techniques for image Enhancement. Specially I had worked in basic linear transformation, gamma correction and blurring techniques of image enhancement. So in basic linear transformation, the operations y = alpha \* x + beta in which y = Output image, x = Input image, alpha and beta are factor related to contrast and brightness. Here we have to manage the values of alpha and beta manually. For choosing value of  alpha and beta automatically, I found lot of difficulties. But manually assigning their value is always better for good quality output. Also we can implement trackbars to choose their values at runtime. For noise removal from image sometimes blurring and filtering techniques also helps in case of lot of the images, I had try that one also. Gamma correction is also one of the best techniques for adjusting brightness and contrast of image. This technique is also called as power law transformation. ie. Y = X^(1/G)In some cases I've observed that the gamma correction provides better results than basic linear transformation. In gamma correction, all entries of matrix of input image firstly we have to convert into of range between 0 to 1 then we prepare a look up table which is done by mathematical calculations mentioned in power law equation. Again the entries of input image replaced by the corresponding entries from look-up table by using inbuilt function LUT  provided by opencv. In such way I had contributed for our group task.

Jayant:

We all the group members contributed equally for a lot of the task. I contributed mainly for the basic linear transformation, and did came up with the way for obtaining the suitable alpha and beta for an image. As suggested later by our mentor to implement the same on parts image by sliding the matrix over it as we do with kernel while performing blurring operations, I implemented it with help of Shubham and Siddhi. Morever, I tried to get better enhancement result by using the histogram equalization, first by splitting it into b,g,r and later upon suggestion by mentors, implemented it by using hsv colour space.

Siddhi:

After doing research on image enhancement topic i found some technique like gamma correction, linear transformation techniques. In linear transformation by taking input (like alpha and beta values) from user image quality is improved but i want to improve my code so that it can predict alpha and beta value according to image. But this method doesn't work properly in all cases so i switched to gamma correction method as this method gives improved output in most of cases. In gamma correction code takes input from user and enhance image according to that, so i want to make it more easy. So i used trackbar to change the value of gamma and it successfully worked. by changing value of gamma image enhanced. Now one more improvement can be done by predicting gamma values according to input image.