



Department of Electronic and Telecommunication Engineering
University of Moratuwa

Assignment I

190610E - Sumanasekara W.K.G.G.

This report is submitted as a partial fulfillment of module EN2550

February 28, 2022

Note: All codes can be found in <https://github.com/GevinduGanganath/EN2550/tree/main/Assignment%201>

Question 1

In the given intensity transformation, pixel values lie within the range 50 to 150 has been increased while other pixels remain same. Above pixel values of a gray scale image generally represent gray colour, hence we can observe that gray colour pixels have been transformed into near white in the output image. Figure 1 shows the original image, intensity transformation function and output image respectively.

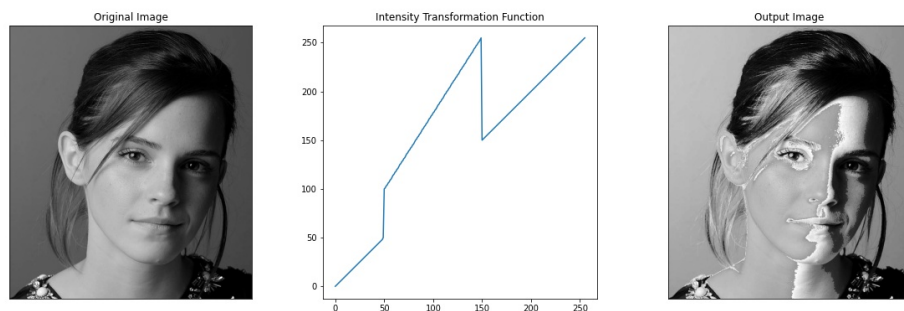


Figure 1: Question 1

Question 2

- (a) In this part the white matter of brain proton density image has been accentuated. Applied intensity transformation is shown in figure 2. Since both the white matter and gray matter [1] have closer pixel values it is very important to select the correct cut-off value. Here 175 was selected as cut-off value and range between 150 and 200 has transformed linearly while others are shifted to pure white or black.

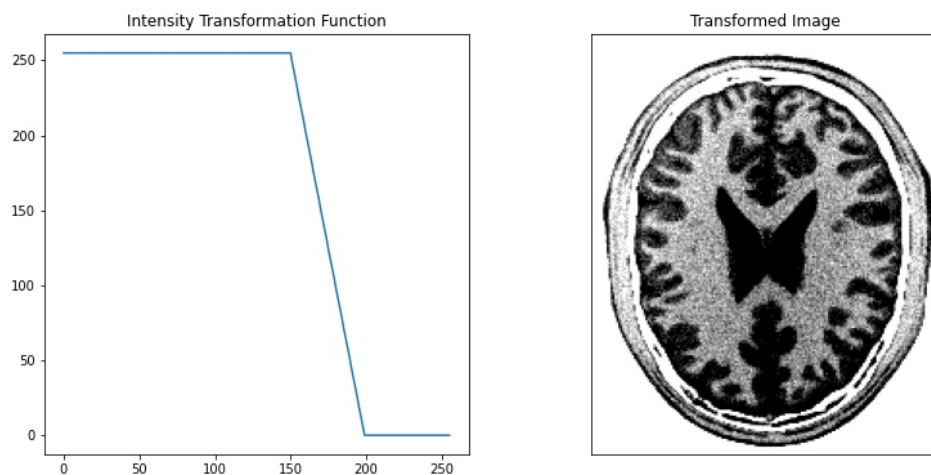


Figure 2: Question 2-a

- (b) As the second part, gray matter of the image has been accentuated. Here the transformation is different from the previous one because if a transformation of the same shape is applied white matter also accentuated and then it is difficult to figure out the features of gray matter. Corresponding transformation function and accentuated image is shown in figure 3.

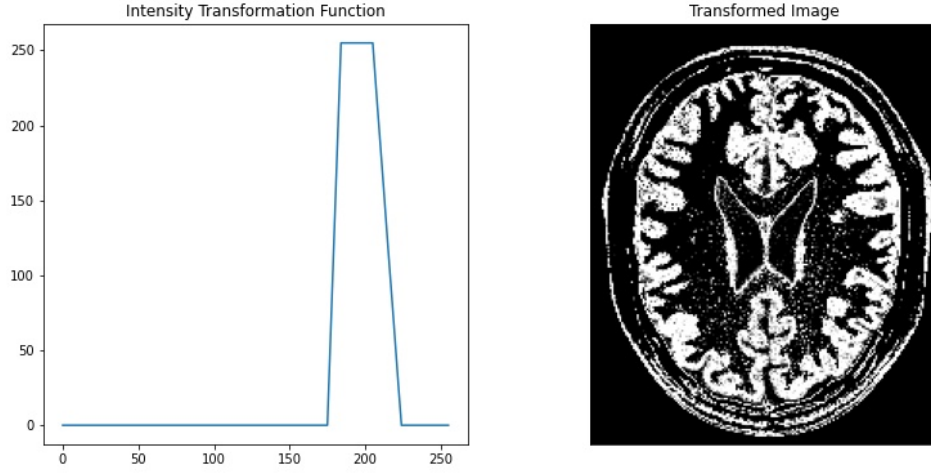


Figure 3: Question 2-b

Question 3

In this exercise a gamma correction ($\gamma = 0.8$) has been performed on the L place of the given image after converting it to the L*a*b colour space [2]. Results are shown in figure 4.

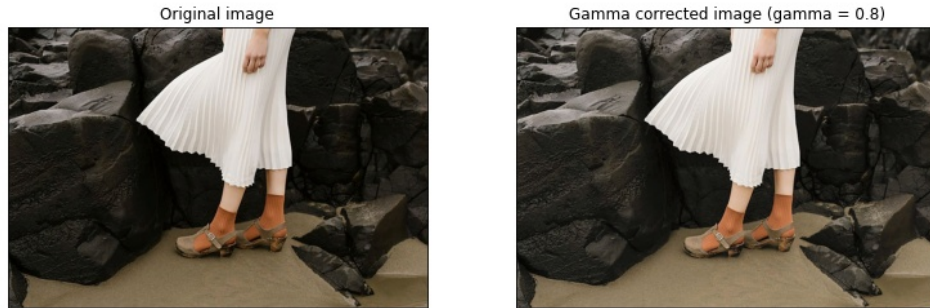


Figure 4: Question 3 input and gamma corrected images

In the L*a*b colour space L represents the lightness of the pixel. According to the equation 1 applying a gamma value less than 1 always produces a new L value which is greater than the previous. Therefore, after the gamma correction output image is lighter than before giving a nice appearance to dark places like rock hollows.

$$\text{new L value} = 255 \left(\frac{\text{current L value}}{255} \right)^{0.8} \quad (1)$$

This can also be represented using the histograms of the two versions of image. As you can

observe in figure 5, after the gamma correction histogram has moved right slightly while storing more pixels in right most bins.

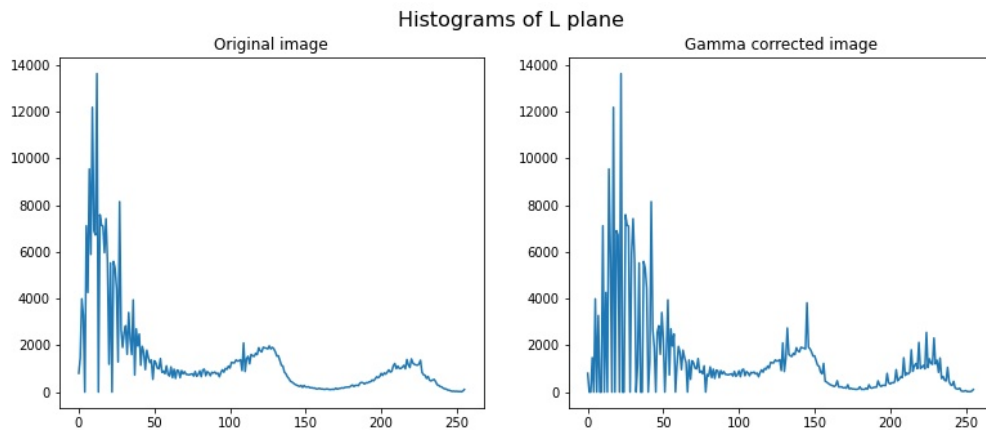


Figure 5: Question 3 histograms

Question 4

Images may have histograms confined into some region, but a histogram of a good image have the values in all regions. Therefore, we need to distribute the pixel values throughout the region. That is what histogram equalization does [3].

In this exercise a python function was written (figure 6) to carry out the histogram equalization on a given image. As the first step histogram of the given image was obtained using the numpy histogram function and the cumulative summation was calculated. Then the histogram equalization equation can be applied resulting a transformation function. Finally, it can be used as a look-up table to generate the equalized image. Resulting histograms are shown in figure 7 and the proper operation of the implemented function can be verified by comparing it with the output of openCV in-built histogram equalization function. Equalized image is shown in figure 8.

```
def equalizeHist(img):
    hist = np.histogram(img.flatten(),256,[0,256])[0]
    cdf = hist.cumsum()
    cdf = np.ma.masked_equal(cdf,0)
    cdf = (cdf - cdf.min())*255/(cdf.max()-cdf.min())
    cdf = np.ma.filled(cdf,0).astype('uint8')
    img_equatized = cv.LUT(img, cdf)

    return img_equatized
```

Figure 6: Histogram Equalization Function

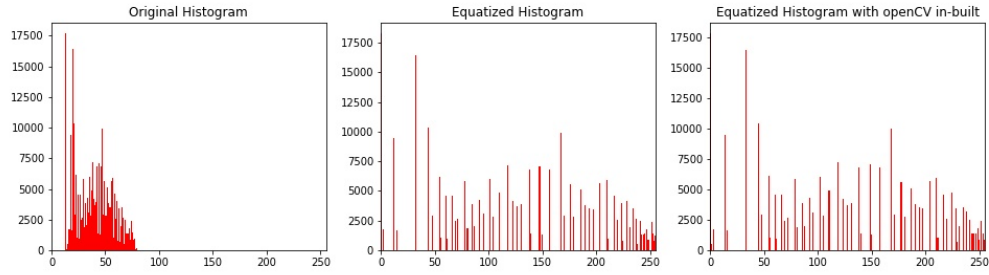


Figure 7: Histograms of shell image

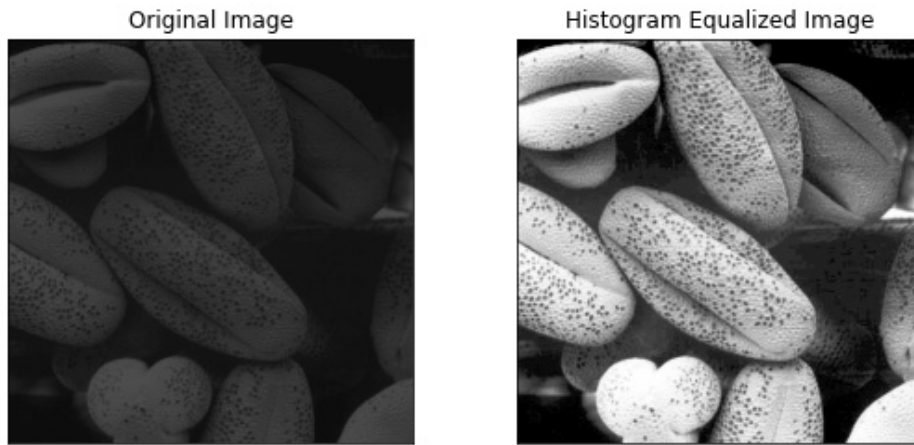


Figure 8: Equalized image

Question 5

Question 6

- (a) Here openCV filter2D function was used to carry out the sobel filter on Einstein image. Sobel kernels detect the edges of a given image. As can be observed in the figure 9 Sobel vertical kernel detects the horizontal edges while Sobel horizontal kernel detects the vertical edges.

Question 7

- (a) Here a grabCut segmentation [4] was carried out to extract the background from the given image. Initially, I tried the segmentation with bounding boxes, but it did not produce the expected output, specially closer to the flower bud and to the petals of flower in question. Then a mask was generated by doing few modifications to the binary thresholded image and grabCut was carried out with mask approximation. In this case the outputs were better than the previous. Results are shown in figure 10.
- (b) In the second part an enhanced image was produced by summing up the foreground image (saturation increased) and background image (Gaussian blurred). Output is shown in figure 11.

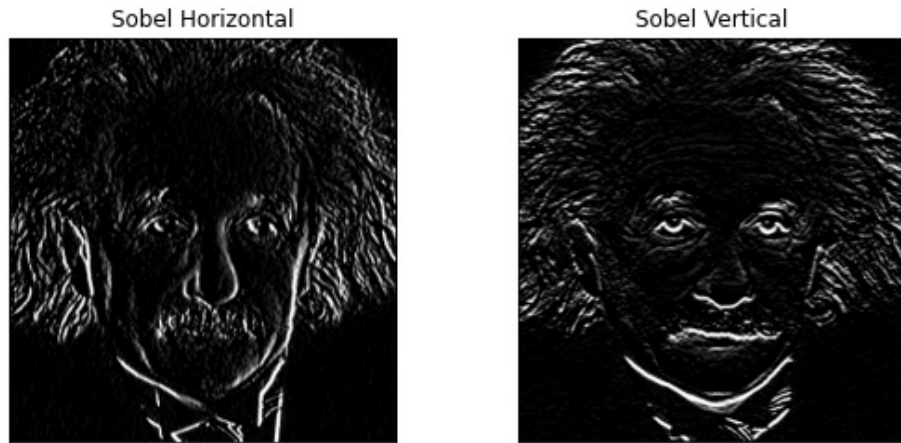


Figure 9: Question 6-a

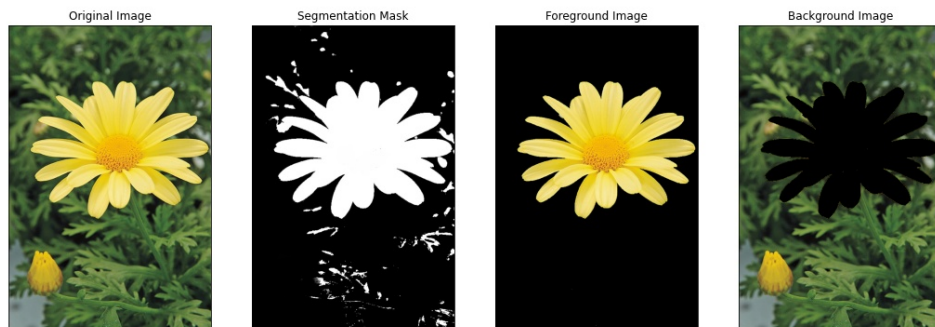


Figure 10: Question 7-a

References

- [1] S. Team, "Grey matter vs white matter in the brain," June 07, 2020. Accessed on: Feb. 14, 2022. [Online]. Available: <https://www.spinalcord.com/blog/gray-matter-vs-white-matter-in-the-brain>.
- [2] T. Mouw, "Lab color values / color spaces," Oct. 08, 2018. Accessed on: Feb. 14, 2022. [Online]. Available: <https://www.xrite.com/blog/lab-color-space>.
- [3] K. S. Htoon, "A tutorial to histogram equalization," Aug. 19, 2020. Accessed on: Feb. 20, 2022. [Online]. Available: <https://medium.com/@kyawsawhtoon/a-tutorial-to-histogram-equalization-497600f270e2>.
- [4] A. Rosebrock, "Opencv grabcut: Foreground segmentation and extraction," July 27, 2020. Accessed on: Feb. 27, 2022. [Online]. Available: <https://pyimagesearch.com/2020/07/27/opencv-grabcut-foreground-segmentation-and-extraction/>.



Figure 11: Question 7-b