Assignment 01

Intensity Transformations and Neighborhood Filtering

submitted for

EN3160 - Image Processing and Machine Vision

Department of Electronic and Telecommunication Engineering
University of Moratuwa

Udugamasooriya P. H. J.
220658U

12 August 2025

1. The given transformation is exactly the identity transformation except in the interval from 50 to 150, where, if the value of a pixel in the original image is denoted by x, and its image under the transformation is denoted by y, we have

$$\frac{y-100}{x-50} = \frac{255-100}{150-50} = 1.55,$$

which gives

$$y = 1.55x + 22.5.$$

This is implemented in the code by first constructing a vector `T` representing the identity transformation and then using `T[50:151] = 1.55 * T[50:151] + 22.5` to alter only the required portion of it as desrcibed above.

The output is as follows;

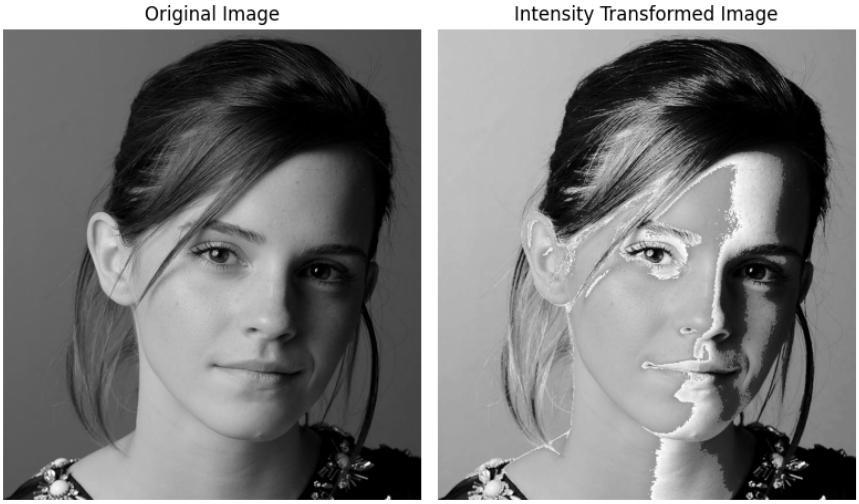

Figure 1: Question 1

2. An eyedropper tool was used to inspect the values of a few representative pixels from the graymatter and whitematter areas and it was observed that pixel intensities less than about 175 correspond to

graymatter. To highlight these pixels, we implement the transformation given by

$$\begin{cases} x \texttt{ // } 6, & x \leq 175, \\ x, & \text{otherwise}, \end{cases}$$

where `//` denotes the integer-division operation (integer quotient upon division); i.e., we proportionally suppress the intensities of the pixels darker than 175, and leave the rest unaffected.

A vector denoting this transformation is easily created by first creating `T` as described above and then executing `T[0:175] = T[0:175] // 6`.

The results are as follows;

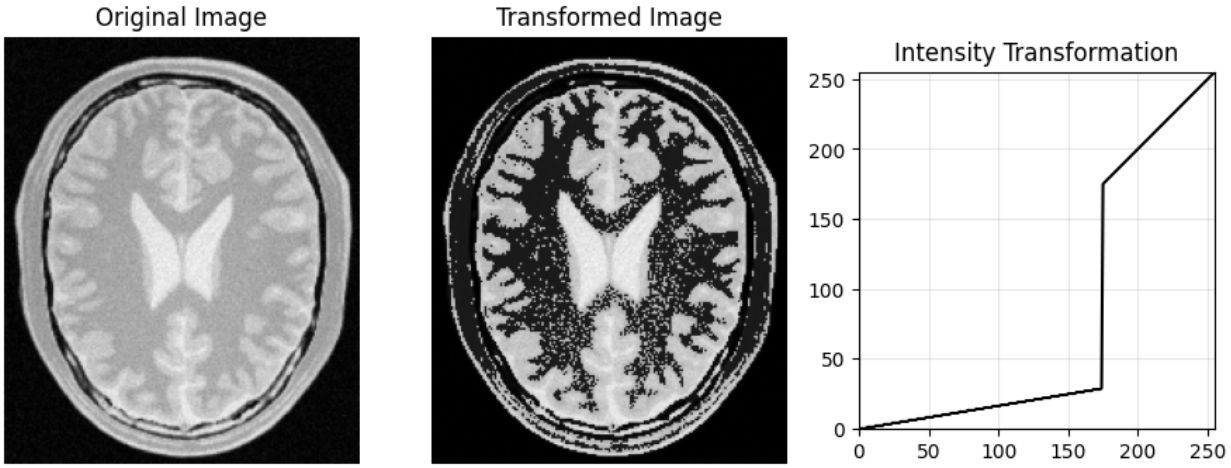

Figure 2: Question 2

3. We open the image, `cv2.cvtColor` it from BGR to the LAB color scheme, and extract the L-channel. We then apply the gamma transformation specified by

$$T(x) = 255 \cdot \left(\frac{x}{255}\right)^{\gamma}$$

to the L-channel. The most aesthetically pleasing result was obtained by setting $\gamma = 0.5$. The results are given below.

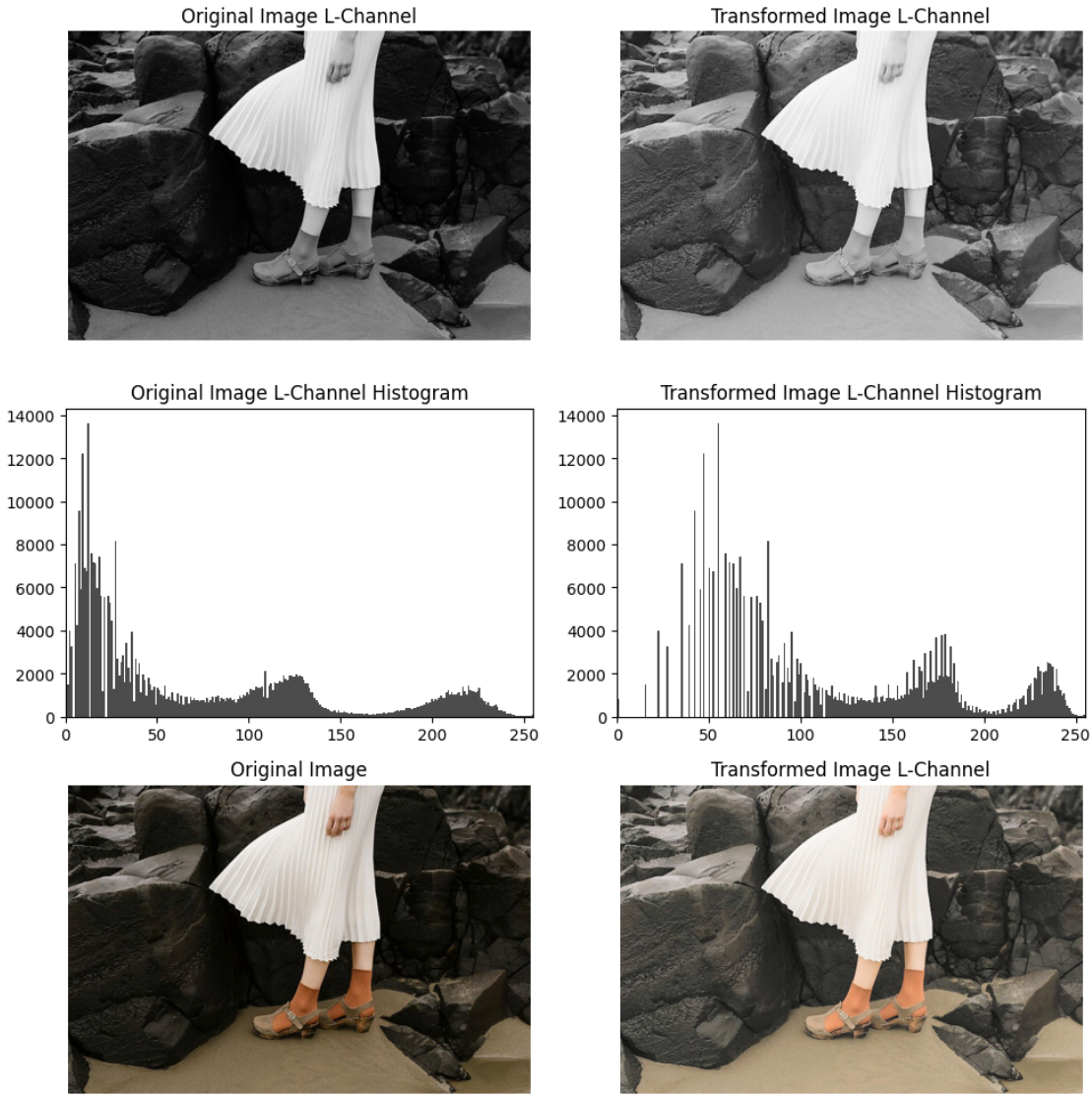

Figure 3: Question 3

4. The lines

```
a = 0.65
sigma = 70

T = np.arange(256, dtype=np.float32)
T = np.minimum(np.ones(256) * 255, T + a * 128 * np.exp(-((T - 128) **
    2) / (2 * sigma ** 2)))
```

implement the intensity transformation. We provide one vector where all entries are 255. and another vector populated with entries computed according the expression given, and use `np.minimum` to pick the smaller of the two. The most aesthetically pleasing result was obtained by setting $a = 0.65$. The results are as follows;

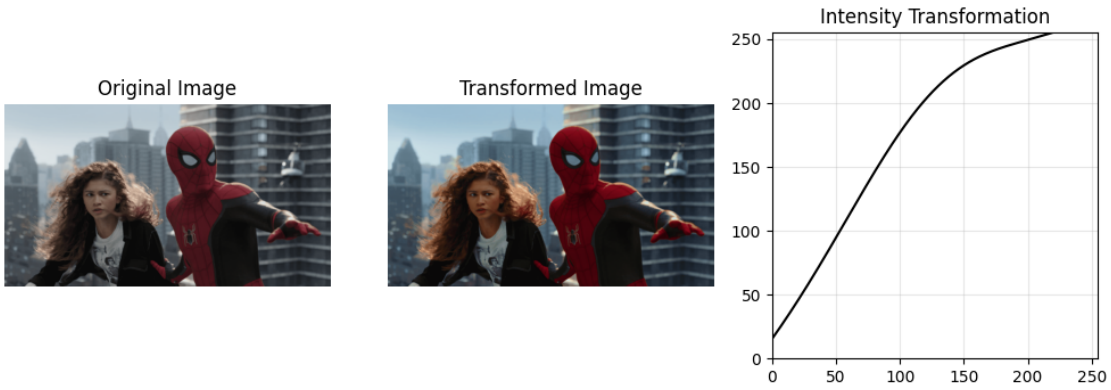

Figure 4: Question 4