Part 1:

Here in this part I have made the Matlab code for transform the image with 4 techniques:

1. Linear transform:

In this technique, I have used the linear contrast stretch Technique. The equation is shown below. And by using the transformed value, I am making my own lookup table.

Here x = transformed value, a = scaling factor (User defined), this factor is multiplying the value (between 0 to 255) to change the slope. And I make sure that if the value which are going above 255, it will consider as 255 by rescaling the matrix. y = Image original value, min = Image minimum value, max = Image maximum value. Here I have used positive and negative slop. Negative slop is used to increase the visibility of the image in above region. In Lookup table, Positive slop = [0,255] and negative slop = [255,0]. The histogram and image is shown below.

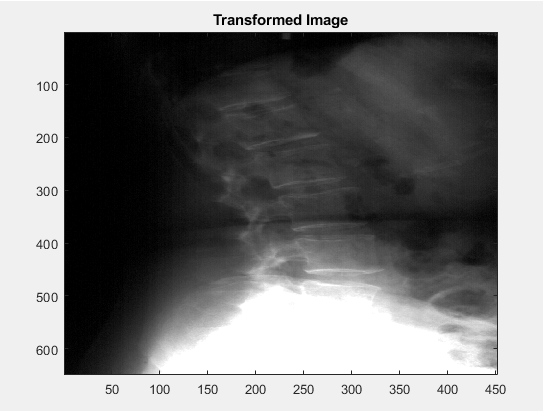


Fig 1: Transformed Image with the user input factor = 2.5, that will increase the upper region visibility.

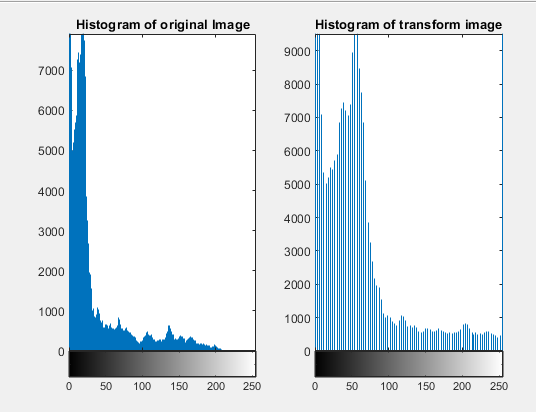


Fig 2: Histogram before and after Transformation, here you can see we can increase the contrast of upper region in the image.

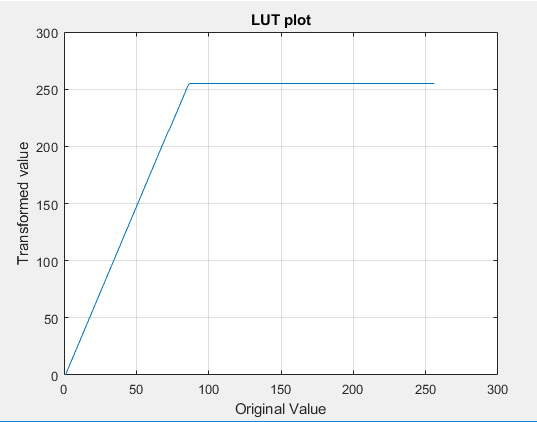


Fig 3: lookup table plot. This plot is used to show the transformation.

1. Log Transformation:

In this technique, I am using 3 log functions.1) log base 10, 2) log base 2, 3) Natural Log. The lut is making according to the Log function. Here in the code, I am using a variable named a. the rezone of using this is to better visualization of log scale on the plot. If this will not use, log scale transformation plot is not looking like log scale plot. And same if the value which is going above 255, it will consider as 255 by rescaling the matrix. The histogram and image is shown below.

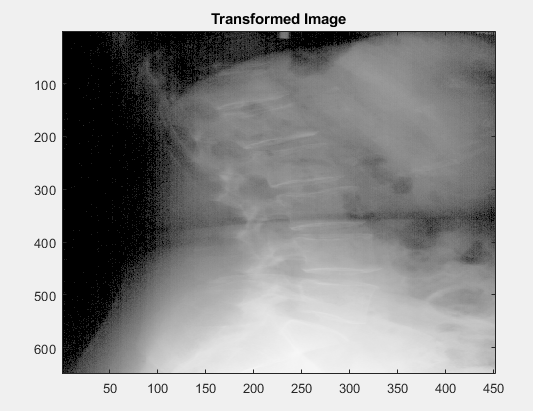


Fig 4: Transformed Image with the user input factor = 1 & used log 10.

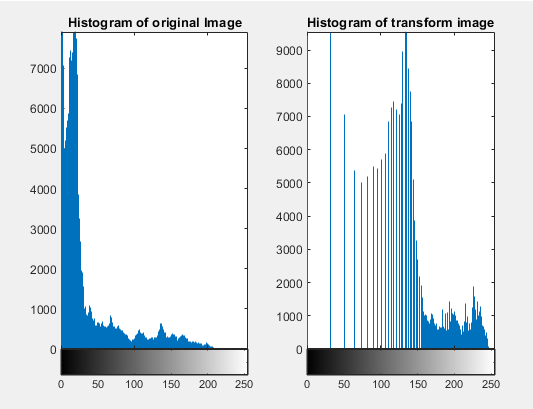


Fig 5: Histogram before and after Transformation

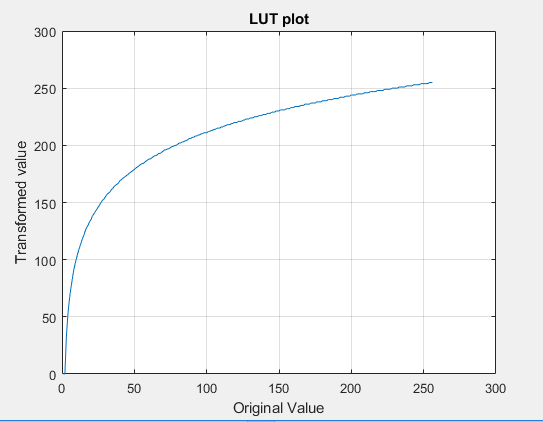


Fig 6: lookup table plot. This plot is used to show the transformation.

1. Exponential Transformation:

In this technique, I am directly using the exponential function. A variable named x1 is used as a user defined variable to change the slop of the function. Here if user want positive exponential, he can use 1 or 2 but if used wants negative exponential, he can use -1 or -2. Here if the value which is going above 255, it will consider as 255 by rescaling the matrix.

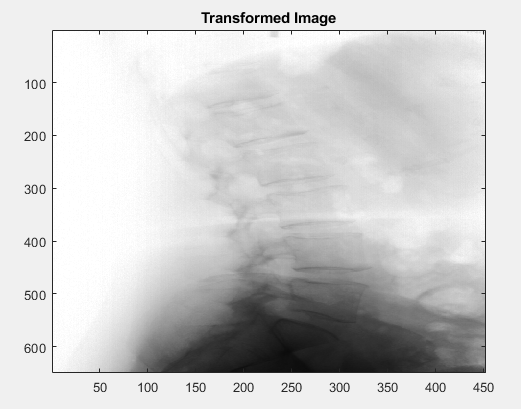


Fig 7: Transformed Image with the user input factor = -2.5.

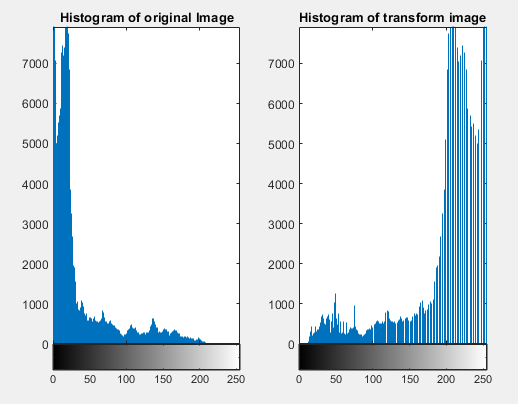


Fig 8: Histogram before and after Transformation.

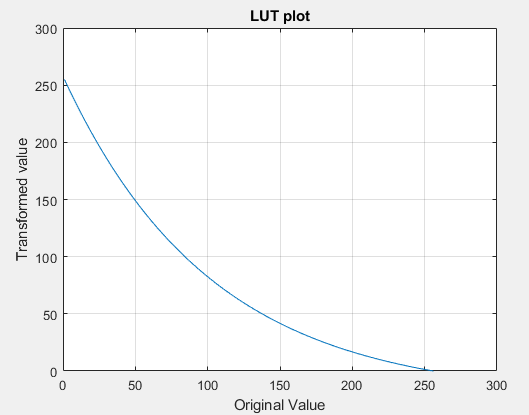


Fig 9: lookup table plot. This plot is used to show the transformation.

1. Break Point Transformation:

This technique is same as linear transformation. The basic difference is between linear transformation and break point transformation is the values between the start and end point is constant, which is the start point value.in the matlab code, lut(start\_point:end\_point,:) = lut(start\_point);. If the start point 100 and end point is 150 (this is basically gray scale value and between 0 to 255), the constant value is lut(100). And a user defined scaling variable is a.

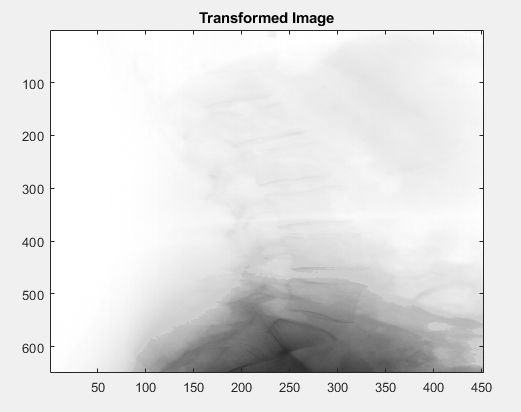


Fig 10: Transformed Image with the Start Point = 50, End Point = 60, a = 2(Choose Negative Slop).

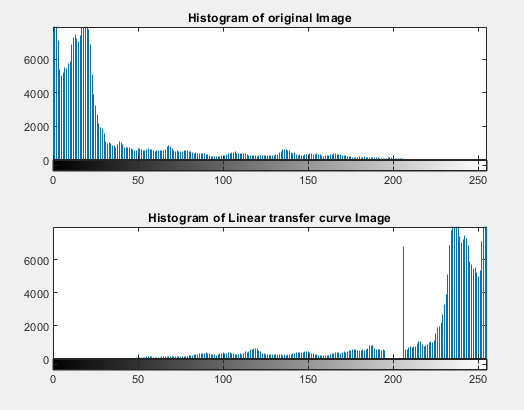


Fig 11: Histogram before and after Transformation.

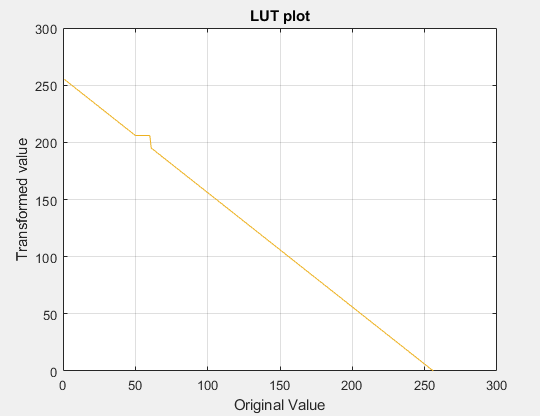


Fig 12: lookup table plot. This plot is used to show the transformation.

Part 2: Histogram Equalization:

1. Global Histogram Equalization:

The idea behind the Global Histogram Equalization is:

1. Find the total number of pixel associated with pixel intensity. In code I am saving this value by using imhist.
2. Calculate the probability of each pixel intensity in the image matrix. We can find this by calculating the total number of pixel in the image. And find the division of number of pixels of individual pixel intensity and total number of pixel.
3. Calculate the cumulative probability.
4. And we need to see the intensity range from 0-255. We will multiply cumulative probability with 255.
5. Round the value from decimal values to integer value (floor rounding).

In the code, the logic behind this is commented.

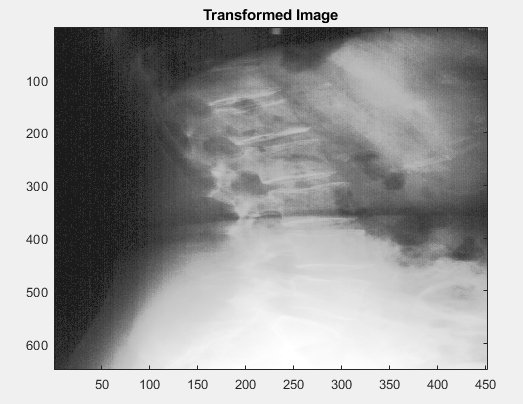


Fig 13: Transformed Image.

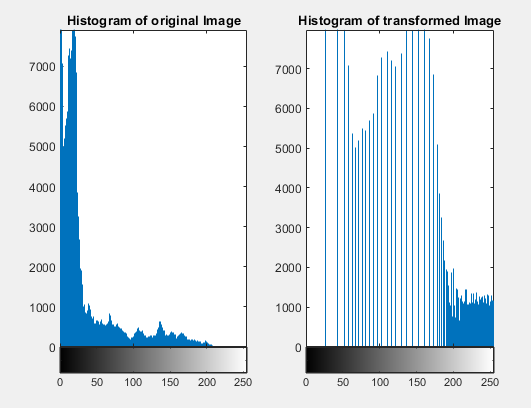


Fig 14: Histogram before and after Transformation.

1. Local Histogram Equalization:

Here for this I am Using the histeq matlab function. histeq function is simple matlab function to enhance the contrast in the image. There are several parameters like histeq(Image,Kernel\_size\*Kernel\_size). This will do the contrast enhancement with the Kernel\_size\*Kernel\_size gray scale level.It is working same as I explained in the global equalization. I am using this direct function to get the accurate gray scale number in the image. Here I am using Kernel\_size\*Kernel\_size(25) gray scale level among 255 gray scale level. For example, if Kernel\_size = 5, I am going to plot 25 gray scale level from 255. This is used because it will set the transformed value (gray scale) according to the total number of pixel. It will transform direct the whole kernel image in to the enhanced contrast image. Here note that Kernel\_size = neighborhood size. The logic behind local Equalization is, I am taking the small neighborhood according to the kernel size. And transform the small neighborhood accordingly the concept. And then change all pixel value with the transformed values. Here if the Kernel\_size is even. It will convert into odd.

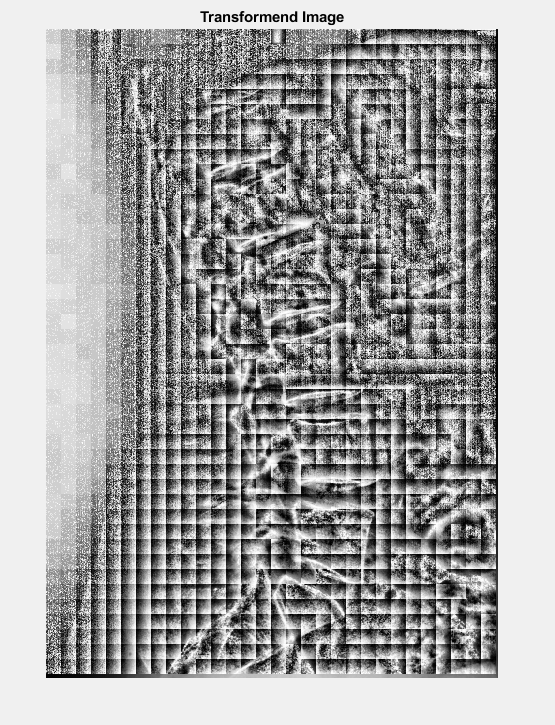


Fig 15: Transformed Image

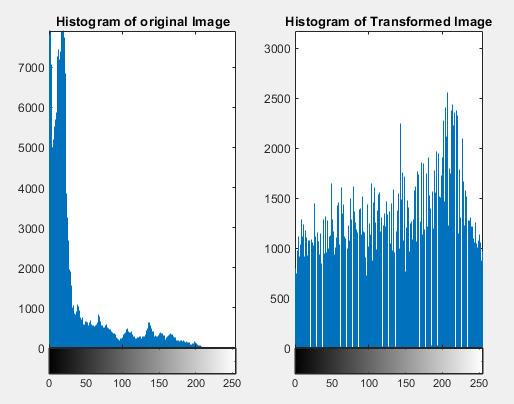


Fig 16: Histogram before and after Transformation.

1. Special Case:

In this case, I am going to merge positive exponential and negative exponential together because of contrast. In my understanding by doing this, the upper part of the image contrast is very optimal. And I am going to decrease the contrast on the below part of the image. This is best suitable combination to enhance the image. And by doing this we can see the most of the details. My approach is very simple. I am assigning the values that from where in the starting gray scale level, I am using the negative exponential. And for ending gray scale level, I am using positive exponential.

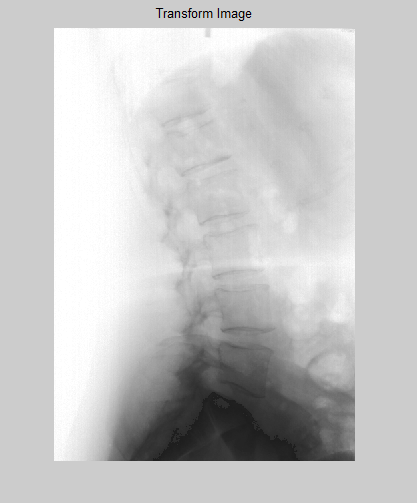


Fig 17: Transformed Image

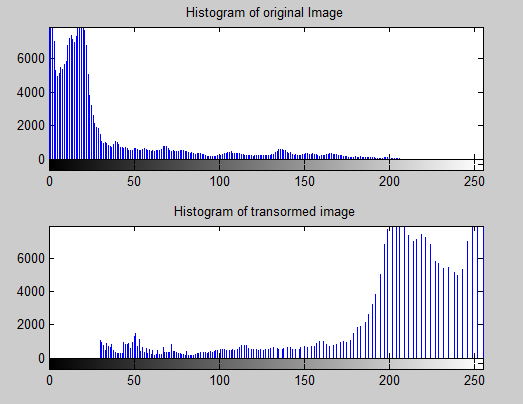


Fig 18: Histogram before and after Transformation.

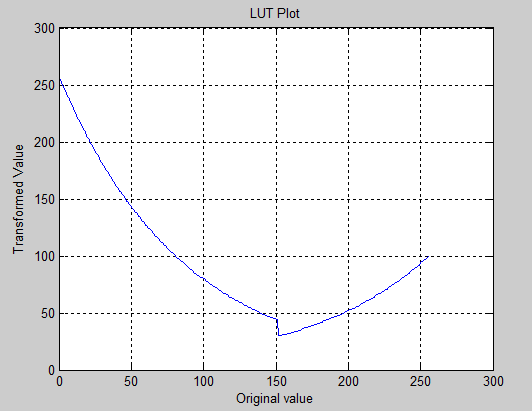


Fig 19: Lut Plot.