Question 2 – Report

In section (A) we implement the log transformation function on an image, first we import opencv library(cv2) and numpy library to use it, then we define a function called log\_transformation with two parameters, first parameter is image path, second one is factor (constant value (c)), then we use opencv library function called imread to read the image as a np.uint8 type, the we define a new variable called out to normalize the original image value from 0.0 to 1.0 to make less computations, the we define a new variable called height to save original image height and another variable called width to save original image width.

After that we loop in the out variable values (i represent height value and j represent width value), then we calculate new image value with this equation:

out[i][j] = factor \* np.log((1 + out[i, j]))

after that we use opencv another time to call imshow function to show original image and the new image after apply log transformation function on the original image.

we test this program on many images, but here we well talk about this function effect on Fig.3.8(a):

this is the original image this is the original image after

apply log transformation function on it with c = 1





this is the original image after this is the original image after

apply log transformation function apply log transformation function

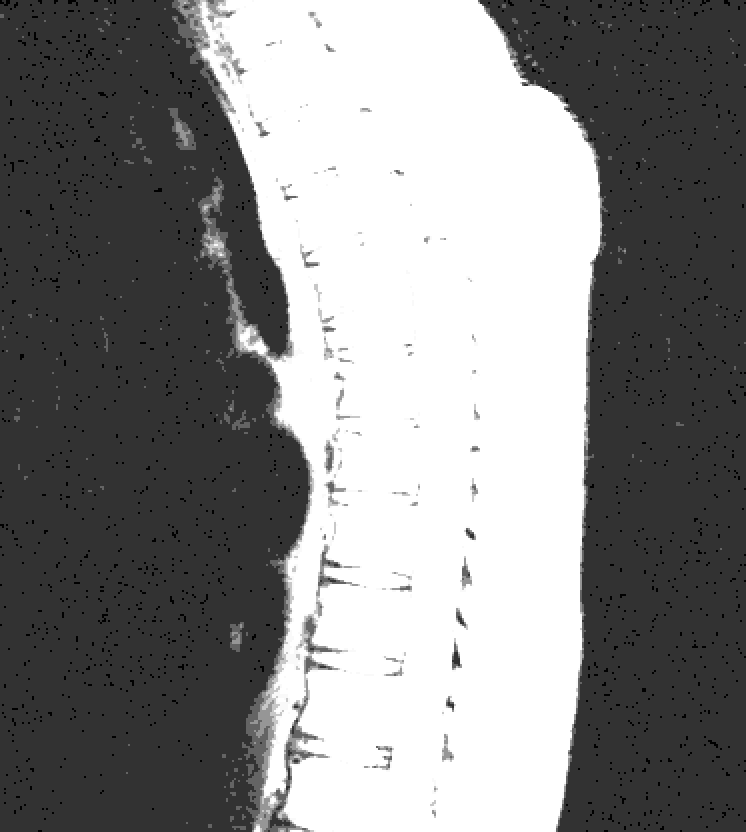
on it with c = 2 on it with c = 5

this is the original image after this is the original image after

apply log transformation function apply log transformation function

on it with c = 10 on it with c = 50

We notice that when we increase the constant value the brightness of the image, because when we increase constant value we’ll multiply the value of (np.log((1 + out[i][j])) with a higher value, so the pixel value well increase and when it increase it well become closer to 255 value and this value represent white color.

In our opinion best value for the constant is between 5 and 10, as we note the dark region in the left it well become more clear and the right side still almost clear.

In section (B) we implement the power transformation function on an image, first we import opencv library(cv2) and numpy library to use it, then we define a function called power\_transformation with three parameters, first parameter is image path, second one is factor (constant value (c)) and the third one is power (y), then we use opencv library function called imread to read the image as a np.uint8 type, the we define a new variable called out to normalize the original image value from 0.0 to 1.0 to make less computations, the we define a new variable called height to save original image height and another variable called width to save original image width.

After that we loop in the out variable values (i represent height value and j represent width value), then we calculate new image value with this equation:

out[i][j] = factor \* np.power((out[i][ j], power))

after that we use opencv another time to call imshow function to show original image and the new image after apply power transformation function on the original image.

we test this program on many images, but here we well talk about this function effect on Fig.3.8(a):

this is the original image this is the original image after

apply power transformation function

on it with c = 1, y = 1





this is the original image after this is the original image after

apply power transformation apply power transformation

function on it with c = 1, y = 2 function on it with c = 2, y = 1

this is the original image after this is the original image after

apply power transformation apply power transformation

function on it with c = 2, y = 2 function on it with c = 10, y = 2

We notice that when we increase the power value the image become more dark, and when we increase the constant value the image become more brightness.

In our opinion the best thing to do is make power value as small as possible and we notice that the value 2 is the best one, and we should make constant value between 10 and 15, so some details well appear more clear.

In the end we notice that the log transformation function is better than power transformation function on this type of images.