

# Computer Vision HW#3

b05902019 資工三 蔡青邑

## Generating the equalized image

In this assignment, I was using python package `skimage` to read the image of `lena` `lena.bmp` as a 2D-list.

And then to find the original distribution for the grey value, I traversed the whole image by for loop and record the distribution in a 1D-list `pixel_count`.

After that, I wrote a function to generate the equalized grey value, `transform()`, which will return a 1D-list with equalized grey value according to the cummulative formula  $s_k = 255 \sum_0^k \frac{n_j}{n}$ . Hence, I got a equalized grey value list, `s`. (I've attached the code of this part tp the following.)

Last , I change each pixel in the image, die to the equalized grey value I got.

After all the works above, below is my image after equalization  
`len_equalized.png`



```
# define transformation function
def transform(pix_count):
    size = 512
    t_function = []
    SUM = 0
    for i in range(len(pix_count)):
        SUM += pix_count[i]
        t_function.append(int(round(255 * SUM / (size * size), 0)))
    return t_function
# generate equalized image and count the frequency of each grey value
s = transform(pix_count)
equalized_pix_count = np.zeros((256,), dtype=int)
lena_equalized = lena.copy()
for i in range(len(lena_equalized)):
    for j in range(len(lena_equalized[i])):
        lena_equalized[i][j] = s[lena_equalized[i][j]]
        equalized_pix_count[lena_equalized[i][j]] += 1
```

## Plotting the Histogram

Similarly, I traversed the equalized image and wrote down the distribution in a 1D-list.

As for drawing histogram, I was using `matplotlib.pyplot` to help myself generate the bar plot `equalized_histogram.png` according to the distribution recorded by that 1D-list.

