multimedia-hw1

> by 108062138 Po-Yu, Wu
> **Readme.md** (http://Readme.md) contains only the *execution* part in `report.pdf`
> **report link** (https://hackmd.io/@sBeNJ4fqRNqa67PhyWWV4A/HJtm95gb2)

1. Color Quantization and Dithering

(a) Perform the median-cut color quantization to transform the given 24-bit color image (Lena image) to an n-bit color image. You need to find the index colors and construct the Look-Up-Table for the bit-reduced color quantization. Show the color-quantized image and compute the color quantization MSE (mean-square-error) error. Perform the above operation for n=3 and 6 and discuss the results

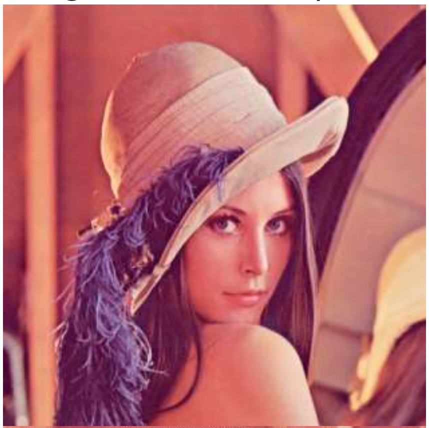
- the color-quantized image in original, n=3 and n=6(original image,n=3,n=6 respectively)

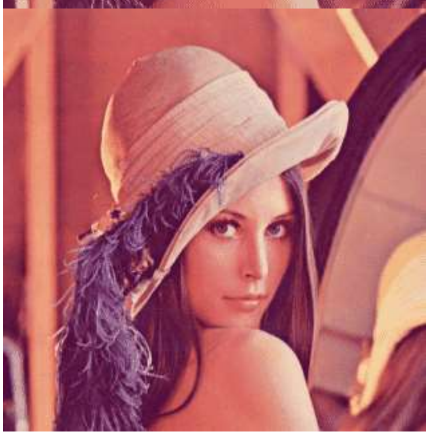
- The color badge size in n=3 is $2^3 = 8$ while the color badge size in n=6 is $2^6 = 64$. The color badge size determines how the image is formed. Smaller color badge is a more coarse grain expression approach to express the image for the number to express the image is less than those with greater color badges. Such limitation makes them face a greater quantization error. Thus, it is quite obvious to guess n=3 has a greater MSE than that in n=6.
    - The following experiment screen shots supports this assumption. MSE in n=6(27.34) < MSE in n=3(66.89)
    - ```
      [44] ✓ 0.1s
      ... mse in n = 3 after the image are quantized by median cut: 66.89740559739359
      ```
    - ```
      [38] ✓ 0.1s
      ... mse in n = 6 after the image are quantized by median cut: 27.3498103936335
      ```
- For the one that owns less types of color(smaller color badge), it is harder for them to express color change in among each pixel. As a result, n=3's image blurs out the delicated, color changing part in the image. Lenna's hair is an instance.

- Lenna's hair: It is comprised of three types of colors in n=3, so the hair strands blurs and a single hair strand can't be easily differientiate.

(b) With the above color quantization (n=3 and 6), apply the diffusion dithering technique to transform the original image to the bit-reduced color image. Show the dithered color image and compute the color quantization MSE error. Discuss the effect of the dithering technique

- the error diffusion image is shown as follow:(original image,n=3,n=6 respectively)



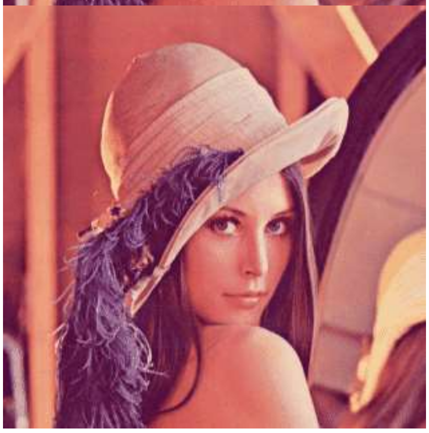


- In (a)'s discussion, we have conclude that the smaller image badge has greater MSE. Thus, it is no doubt that n=6, dithered, image looks greater than that in n=3, dithered image. Thus, it is no doubt that the n=6 has a lower MSE even after dithered. Furthermore, my dithered image has lower MSE in n=3 and n=6 case

- MSE in n=6(27.04) < MSE in n=3(66.84)
- ```
  [29] ✓ 4.9s
  ... mse in n = 3 after the image are quantized by median cut with the help of diffusion dithering: 66.84853522405598
  ```
- ```
  diffusion_dither_img = diffusion_dither(img, colors)
  #show_image(diffusion_dither_img, 'error diffusion dithering')
  img = cv2.imread(target_image, cv2.IMREAD_COLOR)
  print('mse ' +'in n = '+str(n)+' after the image are quantized by median cut with the help of diffusion ditherin
  [39] ✓ 31.7s
  ... mse in n = 6 after the image are quantized by median cut with the help of diffusion dithering: 27.04151978849543
  ```

* The discussion starts when we put quantized image with the dithered image. Let's start at n=3.
  - As we can see, median cut looks rougheset and contains long, same color area while the dithering one remove the long, same color area for it diffuses the error to its neighbors. Although long, same color area is removed in dithered version. It seems that the dithered image looks more "particle". I guess this may arise from the diffusion part in error diffusion due to its error can spread and affect its neighbor pixel.

- At n=6, it seems the “particle” is smaller. I guess the error diffusion in n=6 can find a closer color. Thus the color particle is smaller.



© bonus: Bonus: You can test the above procedure on different images to show better effect on color quantization and dithering

- In my opinion, having better effect on color quantization and dithering is equivalent to having a lower MSE. In other word, our goal is to minimize MSE. To do so, I intentionally pick a mono-colored picture. The first image is the original image, the second is median-cut image, and the last one is the error-diffusion with median-cut. Since the original image focus on the man with umbrella, upper-left corner and lower side of the image are blurred. Since pper-left corner and lower side of the image's color varies a lot, the weakness of quantization can be hidden due to the fact that the quantization does blur the image. As a result, I think this is a good example to demonstrate the power of quantization.
- In my opinion, the error-diffusion with median-cut cheates my eyes and looks almost identical to the original image

(d) execution

- go to `1.ipynb`

- Please adjust `n` to customize the color badges. In this assignment, `n` can be either 3 or 6, generating the color badges with `2^3` and `2^6` respectively.
- click `Run all` to see the output(`{n}` varies according to `n` 's definition)
- output locates at `./out/` folder
  - `error_diffusion_dithering_{n}.png`
  - `median_cut{n}.png`
- To verify the implementation's correctness, I construct `test_color` to verify that each pixel is limited inside the color badge.

2. Interpolation

> Write the image interpolation function to upsample the givenimage to 4 times the original width and height. Implement the following two different interpolation methods and show the 4X (both x and y directions) upsampled image. (You should not use any built-in function for the interpolation.)

(a) Nearest-neighbor (NN) interpolation

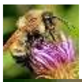
- The original image sits on top of the nearest neighbor interpolation image

(b) Bilinear interpolation

- The original image sits on top of the binary interpolation image


© Compare results from (a) & (b). Discuss what you observe

- let’s line them together!



- The nearest neighbor looks quite “pixel”. The “pixel” style is derived from the characteristic “find the nearest neighbor”! Considering that finding the nearest neighbor means mapping back to the original image, which is smaller. A block of pixels of then maps back to the same pixel, since their interpolation image indexes divided by the interpolation rate usually be rounded to the same index in the original image.
- The bilinear interpolation image blurs out the “pixel” style but blurs out each “pixel” styles’s frame, making the whole image’s color looks quite “continue”. This continuity is derived from the binear property, which takes its bilinear sum of its neighbor. Apart from that, the black bar only apprears on the bilinear interpolation image.

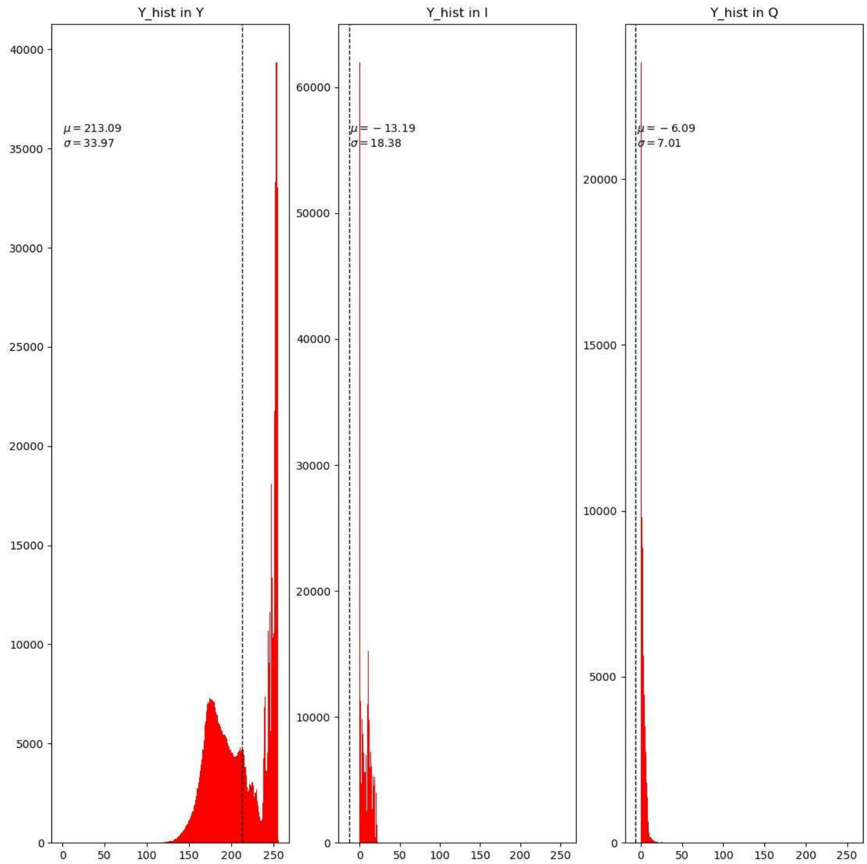
(d) execution

- go to `2.ipynb`
- Please adjust `n` to customize the interpolation ratio. In this assignment, `n` is 4.
- click `Run all` to see the output
- output locates at `./out/` folder
  - `bee_near.jpg`
  - `bee_linear.jpg`

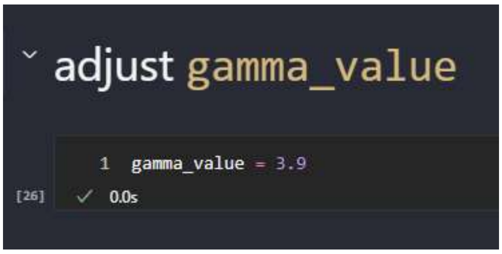
3. Photo enhancement

> implement the following steps to correct the image "ChangKungLake". (You cannot use any
> built-in functions to perform the enhancement procedure)

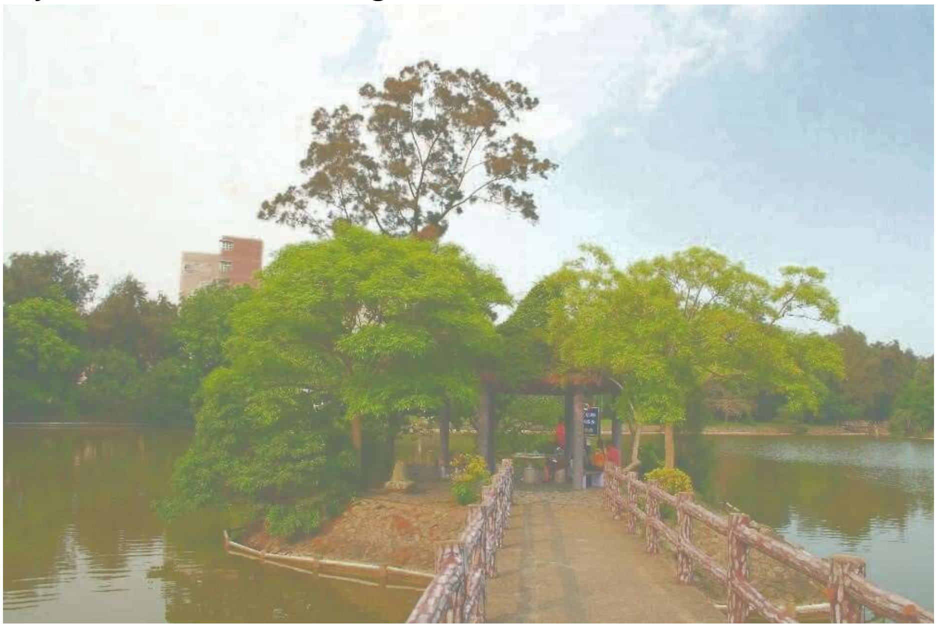
(a) Convert the RGB color space to YIQ, and show the image histogram of Y channel

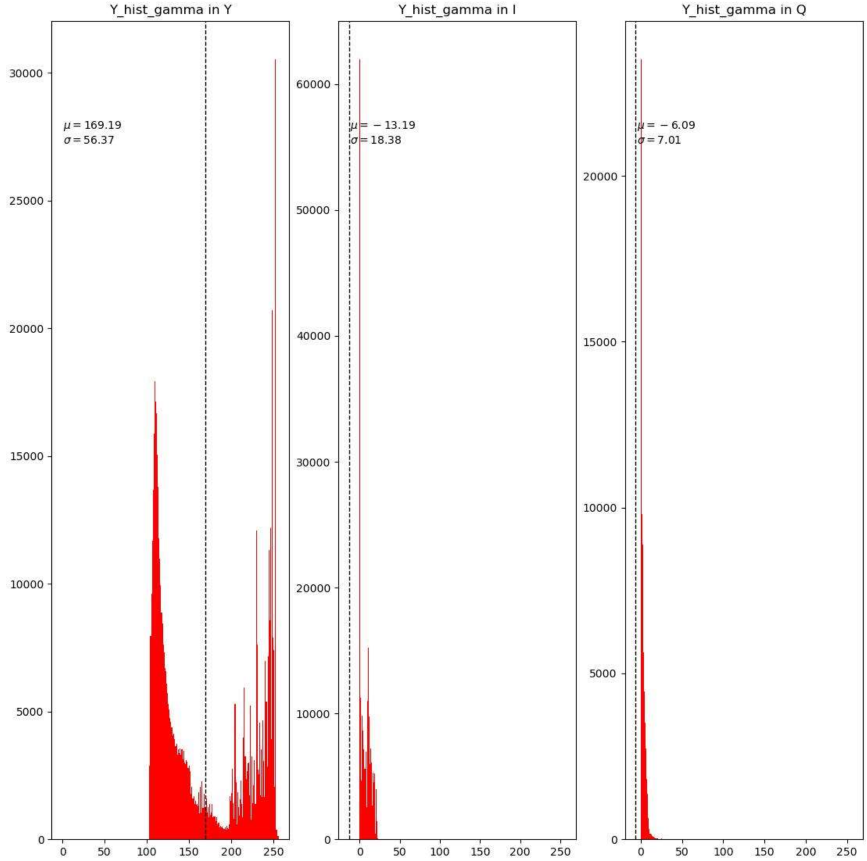
- original image's Y channel's histogram is shown as follow
  - 

(b)Apply gamma transform to Y channel with a suitable gamma value

- set my own `gamma_value`
- 

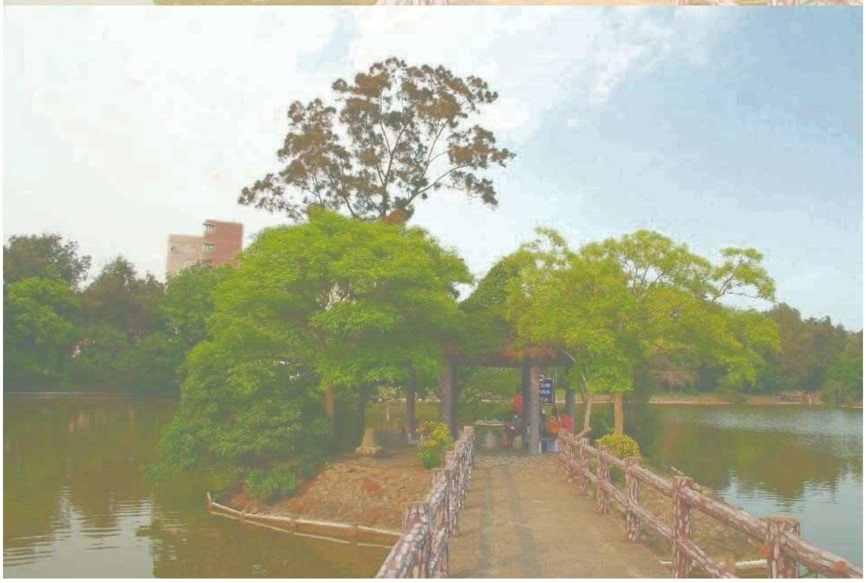
©Convert the transformed image from YIQ color space back to RGB to show the result with the best gamma value. Also show the histogram of Y channel for the transformed image.

- my result for the best gamma: \(3.9\), and its result is

- transformed image's Y channel's histogram is shown as follow

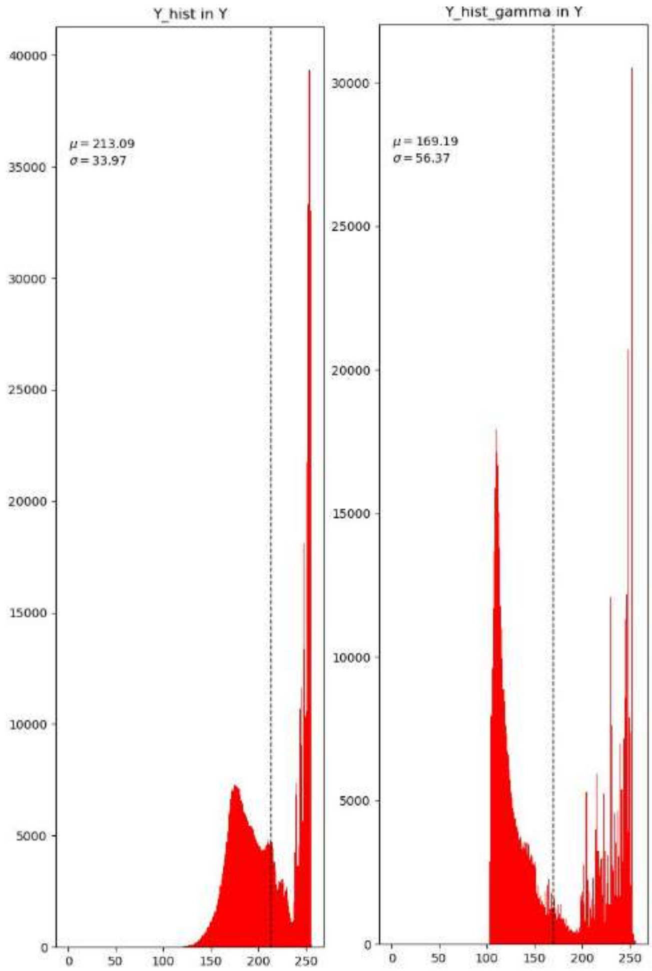
- 

(d) Compare the image and histogram before and after your enhancement. Discuss what you observed.

- First, let’s line up the original image and the transformed image. The original image sits on top of the transformed image.
  - The original image is overexposured and looks too bright, too pale. The transformed image seems more vibrant.

- 

- Second, I pile together original image's Y histogram and transformed image's Y histogram. It seems the overall distribution of the intensity(Y) moves a little bit to the left side.
  - The peak of the transformed distribution remains around 255, but the for the lower peak, which locates around 175 in the original image, shiftes and accumulates around 100.
  - The lower peak's intensity around 100 reaches 17000 as compared to the one in the orginal image reaches only 7500. It turns out the gamma function enlarge the gap, making the distribution M-shaped.
    - M-shaped intensity distribution implies the image has better contrast and becomes more vibrant.

- Aside from that, lower peak's shift from 175 to 100 also makes the overall image darker and alleviate the problem on the overexposure in the original image.
- 

(e) execution

- go to `3.ipynb`
- Please adjust `gamma_value` to customize the interpolation ratio. In this assignment, `gamma_value` is 5
- click `Run all` to see the output
- output locates at `./out/` folder
    - `Y_hist.jpg`
    - `Y_hist_gamma.jpg`
    - `gamma_img.jpg`