Problem Set #2, SP part

- Issue date: Nov. 13, 2022; Deadline: 23:59, Nov. 27, 2022
- use **a=imread('a.jpg')** to load picture and use **load('b.mat')** to load matrix
- Please use **figure**, **subplot(x,x,x)**and **title(' ')** to make your output beautiful.
- Please use matlab to output ".pdf" and upload to Gradescope.
- **Good luck and have fun!**

1. Color balance , Saturation,and Demosaicing

(1)White balance

Apply the white balance on "sky.jpg" and name the output image as "WhiteBalanceImage".

$$I(x,y) = 0.299f_R(x,y) + 0.587f_G(x,y) + 0.114f_B(x,y)$$

$$k_R = \frac{\overline{I}}{\overline{f_R}}, k_G = \frac{\overline{I}}{\overline{f_G}}, k_B = \frac{\overline{I}}{\overline{f_B}}$$

$$\begin{bmatrix} g_R(x,y) \\ g_G(x,y) \\ g_B(x,y) \end{bmatrix} = \begin{bmatrix} k_R & & \\ & k_G & \\ & & k_B \end{bmatrix} \begin{bmatrix} f_R(x,y) \\ f_G(x,y) \\ f_B(x,y) \end{bmatrix}$$

(2)Adjust Saturation

Increase the saturation of "sky.jpg" by 50% ,and name the output image as "SaturationImage".

Also,You cannot use the function"rgb2hsv()or hsv2rgb()"

$$I = \frac{1}{3}(R+G+B)$$

$$S = 1 - \frac{3min(R,G,B)}{R+G+B}$$

$$H = \begin{cases} \theta . G \geq B \\ 2\pi - \theta, G < B \end{cases}$$

$$\theta = cos^{-1}[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}]$$

$1^\circ if\ H \in [0^\circ, 120^\circ)$:
$H = H$
$\begin{cases} B = I(1-S) \\ R = I[1+\frac{S\cos H}{\cos(60^\circ - H)}] \\ G = 3I - (R+B) \end{cases}$

$2^\circ if\ H \in [120^\circ, 240^\circ)$:
$H = H - 120^\circ$
$\begin{cases} R = I(1-S) \\ G = I[1+\frac{S\cos H}{\cos(60^\circ - H)}] \\ B = 3I - (R+G) \end{cases}$

$3^\circ if\ H \in [240^\circ, 360^\circ)$:
$H = H - 240^\circ$
$\begin{cases} G = I(1-S) \\ B = I[1+\frac{S\cos H}{\cos(60^\circ - H)}] \\ R = 3I - (G+B) \end{cases}$

(3)Demosaicing

Alice has an old camera with a picture on it.But this picture is in RAW format. Please use the demosaicing algorithm to convert this single-channel image to a three-channel RGB image. (Linear interpolation is recommended)

The initial image is given as "flower.mat". Name the output image as "RGBImage". **You cannot use the function"demosaic()"**

2. Noise generation and degeneration

(1) *Noise generation*

Add the following noise to the initial image separately：

Gaussian noise： mean and variance are 25 and 25 respectively;

Salt-and-pepper noise： intensities are 0 and 255 with probabilities $P_{salt} = P_{pepper} = 0.05$;

Name the output images as "NoiseImage1", "NoiseImage2" and display both images. **You are free to use the function "imnoise" or other built-in functions to generate the noise.**

(2) *Noise degeneration*

Please design a median filter and an average filter, and apply both your filters to the "NoiseImage1" and"NoiseImage2". Then display all your output image results.

3. Log/gamma Transformation and histogram equalization and matching

(1)*Histogram*

Develop a function which computes the **gray-level histogram** of the "origin.png". Then using this function to find the histogram and display it.**You cannot use the function “imhist()”**

(2)Log Transformation and Gamma Transformation

use transform function $s = c\log(1 + r)$ and $s = c \cdot r^{\gamma}(\gamma = 0.4)$ to deal with the image.Display the output image and its gray-level histogram.

(3)*Histogram equalization*

Use the function histeq() to apply histogram equalization on the "Lena.png". Then using this function to equalize the histogram which you develop in 3.1. Display the output image and its gray-level histogram.

If you can design a function by youself instead of using function histeq() in this part, you can get a Bonus scores(5’)

(4)*Histogram matching*

Use the function histeq() to apply the histogram matching between the source image and the target image. Use this function to display the output image. And you also need to display the gray-level histogram of the source image, the target image and the output image.

If you can design a function by youself instead of using function histeq() in this part, you can get a Bonus scores(5’)