**BE2080 Week#7**

**1. Please submit your Matlab programs (.m files and .fig files if applicable). Please name your files clearly, for example, Wk7\_Q1.m.**

**2. Please check and make sure all your files are submitted.**

1) (10 points) Write a function to return the type (color or gray) of an input image. The function takes one input argument, which is the file path of an image; and returns one output argument in the form of a string array, which is the type of the image.

You can use the Matlab stock images such as onion.png or pout.tif to test your function. Below are example calls of the function:

>> type = CheckImage('onion.png')

type =

“true color”

>> type = CheckImage('pout.tif)

type =

“gray”

2) (10 points) Write a function to show a square “random” image. The function will take in “at least” one input argument which is the size of the image in pixels. As a default, the image will be shown in black-white. If a second input argument is provided, it will be the colormap. Below are examples of images shown by calling to the function:

|  |  |
| --- | --- |
| randimg(200) | randimg(200, hsv) |
|  |  |

3) (10 points) Write a function with one input argument and two output arguments. The one input argument is the file path of an image on your computer.

The function will first read in the image. If the image is a gray scale image, the function will return the string “gray” as the first output argument, and the **mean** pixel value of the image as the second output argument. If the image is a true color image, this function will calculate the **mean** pixel values for the red, green, and blue components (or channels) of the image. It will then return the color channel with the highest mean intensity in the form of a string as the 1st output argument, and the corresponding mean pixel value as the 2nd output argument.

As an example:

>> [MaxChannel, MeanInt] = ColorChannel('onion.png')

MaxChannel =

“red”

MeanInt =

137.3282

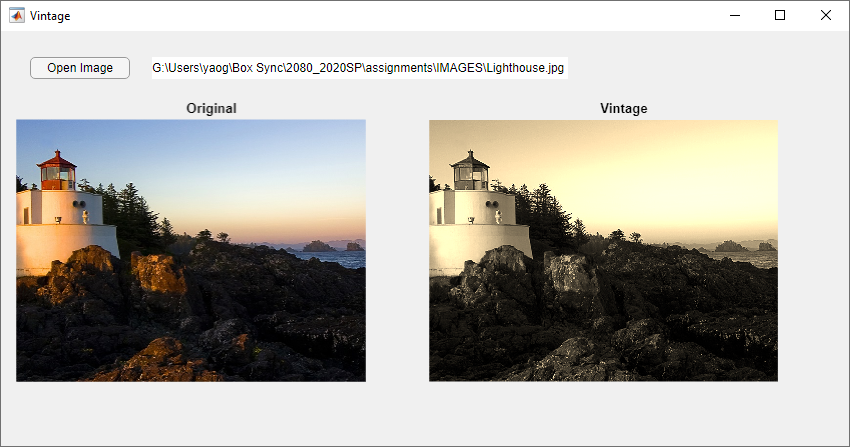
(4) (10 points) Write an Matlab APP named Vintage() to convert a true color image to a vintage style (sepia tone) image. The formula to convert the RGB color channels is:

New\_Red = (Old\_Red \* .393) + (Old\_Green \*.769) + (Old\_Blue \* .189)

New\_Green = (Old\_Red \* .349) + (Old\_Green \*.686) + (Old\_Blue \* .168)

New\_Blue = (Old\_Red \* .272) + (Old\_Green \*.534) + (Old\_Blue \* .131)

Your APP should look the same as the example below. Make sure that: (1) the function should be able to open an image saved anywhere on the computer, not just from the current folder; (2) the full file path should be displayed in the Label next to the “Open Image” button.



5) (20 points) Write a function that will create and show an image as illustrated in the examples below. The function shall be able to receive up to three variable numbers of input arguments.

If no input is provided, the function generates and shows an image of three stripes (of 120 x 120 pixels in size) with a black stripe at top and bottom, and a white stripe in the middle.

If one input argument is provided, it must be an integer number that represents the size of the square-shaped image in number of pixels. The default color for top/bottom stripe is still black, and the default color for the middle stripe is still white.

If a second input argument is provided, it should be the color of the middle stripe. If a third input argument is provided, it will be the color of the top and bottom stripes.

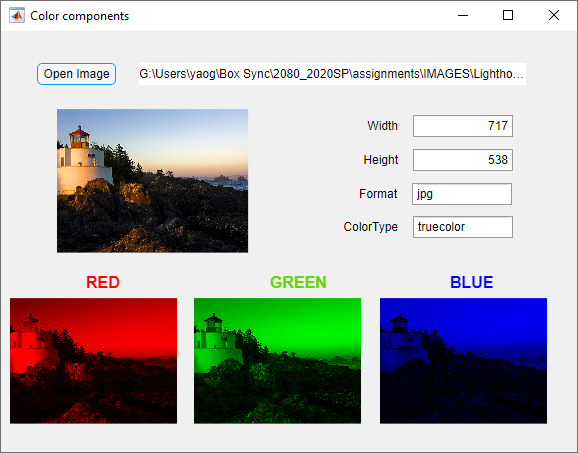
All boundaries of the image must be shown in the same color as the top and bottom stripe.

For example, a call to this function will show the following image based on the input arguments provided:

|  |  |  |  |
| --- | --- | --- | --- |
| **CreateImage()** | **CreateImage(200)** | **CreateImage(200, [1,0,0])** | **CreateImage(200, [1,0,0],[0,0,1])** |
|  |  |  |  |

(6) (20 points) Write an Matlab APP to open and display a color image, and show the information of the image size, format, and color type. These information can be obtained using the imfinfo() function. If this image is a “truecolor” image, the program should show the red, green and blue channel of the image in the other three axes (shown below). If the image is a gray scale image, all three sub-images are the same as the original gray scale image.

Your APP should look the same as the example below. Make sure that: (1) the function should be able to open an image saved anywhere on the computer; (2) the full file path should be displayed in the Label next to the “Open Image” button.

****