

Hands-on

I . Image Type

1. Load the file 'koreaUniversity.jpg' and save as the variable 'a' (imread())
2. Display the variable 'a' on screen. (imshow())
3. 'a' is consist of 3 tables which have RGB information.(a(row, col, color))
display the each color of 'a'. (imshow())

Tip. Using figure, you can see the result of imshow(), or plot() in a new window.

(figure, imshow(image))

4. Make a gray scale image of 'a' and save as the variable 'b'.
(a) Make a function 'RGB_to_gray' with a RGB image as an input parameter and a gray scale image as an output parameter.

*Tip. RGB to gray scale formula ($gray = 0.3 * red + 0.59 * green + 0.11 * blue$)*

- (b) Using 'RGB_to_gray', change 'a' into a gray scale image and save as the variable 'b'.

- (c) Using the function 'rgb2gray' which is provided from the Matlab, change 'a' into a gray scale image and save as the variable 'b2'.

Tip. 'b' and 'b2' aren't totally same. Because the formula of 'rgb2gray' has the fourth decimal places precision and the formular of 'RGB_to_gray' has the second decimal places precision.

5. Display the image of 'b'. (imshow())
6. Make a binary image of 'b' and save as the variable 'c'. (binary image is consist of 0 and 1.)
7. Display the image of 'c'. (Hint. normalization)
8. save 'a' as the file 'RGB.png', save 'b' as the file 'gray.png', and save 'c' as the file 'binary.png'. (imwrite())

Tip. When you save a binary image, put '1' in 'bitdepth' information. (search using 'help')

9. Compare the sizes of the three file.

Which file has more information? (color, brightness)

Which file's size is smaller?

II . Sampling

1. Load the file 'lena.bmp' and save as the variable 'lena'. (imread())

2. Display the image of 'lena'. (imshow())

3. Display the size information of 'lena'. (whos() or size())

4. Resize the image of 'lena'.(imresize())

(a) Resize the image of 'lena' into 1/2 rows and 1/2 columns and save as the variable 'lena2'.

(b) Resize the image of 'lena' into 1/4 rows and 1/4 columns and save as the variable 'lena3'.

(a) Resize the image of 'lena' into 1/8 rows and 1/8 columns and save as the variable 'lena4'.

5. Resize the images of 'lena2', 'lena3', and 'lena4' into the size of the original image of 'lena'. (imresize())

6. Display the images of 'lena', 'lena2', 'lena3', and 'lena4'. (imshow(), subplot())

7. Find the damaged parts of the displayed images of 6.. (eg. hair, face, hat)