How To Handle Image Data Type In Matlab

Tsung-Yi Lin

In homework 2, you are asked to compute MSE of denoised images. Many of you did totally wrong MSE computation because of data type issue. The content below can help you handle image data type.

When you read in an image by imread()

```
I = imread(filename);
```

The data type of I depends on the values saved in the image. If image were saved as a binary image, its type will be type *logical*, which only has 1 bit representation. Generally, the image is saved as type *uint*8 and with one channel as gray image or three RGB channels as color image. The *uint*8 type means unsigned 8 bit integer, that is, the value within the type can only be positive integer between 0 to 255.

The problem happens if you do not change data type before processing. Take MSE computation as an example

```
I1 = imread(filename);
I2 = imread(filename);
MSE = sum(sum( (I1-I2).^2 ));
```

There are two problems in the program. First, *uint*8 can not take negative value, so if the subtraction is negative, *uint*8 will represent it as 0. Second, *uint*8 can not be larger than 255, so it will truncate every number greater than 255 to 255. As a result, we should take care of data type very carefully. Cast the image data type to double before any processing usually is a safe way to avoid above truncation issue.

```
I = double(imread(filename));
```

Some people will use Matlab build-in function

```
I = im2double((imread(filename)));
```

which can also change the data type from *uint*8 to *double*. But the value range of image will be converted to [0, 1] instead of [0, 255].

At last, if you want to use imshow() to show your image, do not forget to convert its data type back to *uint*8. imshow() has two function overloading. It takes image with type *uint*8 as it ranges between [0, 255], and takes image with type *double* as it ranges between [0, 1].