**Assignment 2 –common Report**

**Task 1**

**Function introduction:**

Totally, I defined 6 sub functions inside function *CartoonNizer\_30988519*:

distance: to calculate euclidean distance of positions of two pixels.

*color\_diff* : distance in intensity space.

*gaussian :* gaussian kernel.

*color\_distance* : kernel of range filter.

*bilateral\_filter* : my bilateral filter function to process single pixel.

The parameters are: src(image), filtered\_img(copy of image), diameter, x(x\_index), y(y\_index), sigma\_s(in space), sigma\_r(in range)

First of all, for each pixel according to neighbour calculate the domain and range filtering. Then combine two filtering, the new value of that pixel is the multiplication between original value and combined filtering kernel. Finally, implement normalization by averaging weights.

*my\_filter* : Integrate the process result of each pixel.

# it will take a long time. I have given the result in out folder.

**Implementation:**

At first, read the image, then process the data of three channels separately using *my\_filter.* After that combine the results of three channels into the 3D array *bilateral.* At last, show the original image, my bilateral filtering image and the image using built-in function.

In order to preserve the edge information, and blur image to get a cartoon effect, select sigma\_s=150, sigma\_r=20 after testing.

**Task 2**

At first, read the image, then preprocess the data using gaussian filter and canny edge detect.

After that initialize the radius range between 15 and 120 with step:2. Using *transform.hough\_circle* in skimage to get the 3D array hough\_res. The first dimension represents radius, the next represent: width\_index and height\_index of circle's center.

Later, set the threshold\_abs to select peaks of each circle’s center. That is if the hough\_res is higher than threshold\_abs, I will select it. (Peaks denote: There is a high probability that it is the center of a certain radius.)

Then add the coordinate of center, hough\_res and radius to three list: centers[], accums[] and radii[].

Finally according to the peaks to draw possible circles and count the number.