

## Homework 1 for Kun

Introduce to image process

All codes are attached on the last page.

Q1a:

```
1 % Q1a
2 I = imread('lena_std.tif');
3 imshow(I), title("Original Image");
```

Res:



Q1b & Q1c & Q1d:

```
5 % Q1b
6 [row, col] = size(I);
7 cnt_pixels = row * col;
8 disp("The size of the image is: " + cnt_pixels);

10 % Q1c
11 max_val = max(I(:));
12 disp("The maximum pixel value is: " + max_val);

14 % Q1d
15 mean_val = mean(I(:));
16 disp("The mean pixel value is: " + mean_val);
```

Res:

```
The size of the image is: 262144
The maximum pixel value is: 245
The mean pixel value is: 124.0425
```

Q1e:

```
18 % Q1e
19 bw = I;
20 for c = 1:col
21     for r = 1:row
22         if bw(r,c) >= mean_val
23             bw(r,c) = 1;
24         else
25             bw(r,c) = 0;
26         end
27     end
28 end
29 imshow(bw,[0 1]), title("Image binarize");
```

Res:



It is image binarize.

Q2:

```
31 % Q2
32
33 factor = 0.5;
34
35 % nearest neighbor interpolation
36 nn = imresize(I,[row*factor,col*factor],'nearest');
37 imshow(nn), title("nearest neighbor interpolation");
38 nn_aft = imresize(nn,[row,col],'nearest');
39 imshow(nn_aft), title("nearest neighbor interpolation return");
40
41 % bilinear interpolation
42 bl = imresize(I,[row*factor,col*factor],'bilinear');
43 imshow(bl), title("bilinear interpolation");
44 bl_aft = imresize(bl,[row,col],'bilinear');
45 imshow(bl_aft), title("bilinear interpolation return");
46
47 % bicubic interpolation
48 bc = imresize(I,[row*factor,col*factor],'bicubic');
49 imshow(bc), title("bicubic interpolation");
50 bc_aft = imresize(bc,[row,col],'bicubic');
51 imshow(bc_aft), title("bicubic interpolation return");
```

Res:

Nearest neighbor interpolation:

nearest neighbor interpolation



nearest neighbor interpolation return

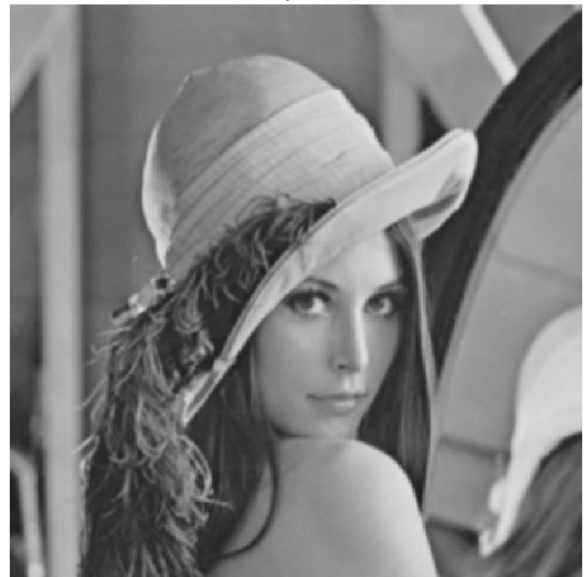


Bilinear interpolation:

bilinear interpolation



bilinear interpolation return



Bicubic interpolation:

bicubic interpolation



bicubic interpolation return



Customized factor:

```
53 % personalized factor
54 factor = input("Input resize factor: ")
55 nn = imresize(I,[row*factor,col*factor],'nearest');
56 imshow(nn), title("interpolation with factor "+factor);
57 nn_aft = imresize(nn,[row,col],'nearest');
58 imshow(nn_aft), title("interpolation with factor "+factor+" reduced");
```

Res:

```
Input resize factor: 0.1
```

interpolation with factor 0.1



interpolation with factor 0.1 reduced



Q3:

```
60 % Q3
61 red_dict = [256,128,64,32,16,8,4,2];
62 for i=1:length(red_dict)
63     redu = I*((red_dict(i)-1)/255);
64     imshow(redu,[0 red_dict(i)-1]), title("Gray reduced with "+red_dict(i)+" level");
65     drawnow;
66     pause(0.5);
67 end
```

Res:

Gray reduced with 256 level



Gray reduced with 128 level



Gray reduced with 16 level



Gray reduced with 64 level



Gray reduced with 8 level



Gray reduced with 32 level



Gray reduced with 4 level





Customized Gray reduce factor:

```
69 ngray_lev = input("Input gray reduce factor: ");  
70 redu = I*((ngray_lev-1)/255);  
71 imshow(redu,[0 ngray_lev-1]), title("Gray reduced with "+ngray_lev+" level");
```

Res:





```

% Q1a
I = im2uint8(rgb2gray(imread('lena_std.tif')));
imshow(I), title("Original Image");

% Q1b
[row, col] = size(I);
cnt_pixels = row * col;
disp("The size of the image is: " + cnt_pixels);

% Q1c
max_val = max(I(:));
disp("The maximum pixel value is: " + max_val);

% Q1d
mean_val = mean(I(:));
disp("The mean pixel value is: " + mean_val);

% Q1e
bw = I;
for c = 1:col
    for r = 1:row
        if bw(r,c) >= mean_val
            bw(r,c) = 1;
        else
            bw(r,c) = 0;
        end
    end
end
imshow(bw,[0 1]), title("Image binarize");

% Q2

factor = 0.5;

% nearest neighbor interpolation
nn = imresize(I,[row*factor,col*factor],'nearest');
imshow(nn), title("nearest neighbor interpolation");
nn_aft = imresize(nn,[row,col],'nearest');
imshow(nn_aft), title("nearest neighbor interpolation return");

% bilinear interpolation

```

```

bl = imresize(I,[row*factor,col*factor],'bilinear');
imshow(bl), title("bilinear interpolation");
bl_aft = imresize(bl,[row,col],'bilinear');
imshow(bl_aft), title("bilinear interpolation return");

% bicubic interpolation
bc = imresize(I,[row*factor,col*factor],'bicubic');
imshow(bc), title("bicubic interpolation");
bc_aft = imresize(bc,[row,col],'bicubic');
imshow(bc_aft), title("bicubic interpolation return");

% personalized factor
factor = input("Input resize factor: ");
nn = imresize(I,[row*factor,col*factor],'nearest');
imshow(nn), title("interpolation with factor "+factor);
nn_aft = imresize(nn,[row,col],'nearest');
imshow(nn_aft), title("interpolation with factor "+factor+" reduced");

% Q3
red_dict = [256,128,64,32,16,8,4,2];
for i=1:length(red_dict)
    redu = I*((red_dict(i)-1)/255);
    imshow(redu,[0 red_dict(i)-1]), title("Gray reduced with "+red_dict(i)+"
level");
    drawnow;
    pause(0.5);
end

ngray_lev = input("Input gray reduce factor: ");
redu = I*((ngray_lev-1)/255);
imshow(redu,[0 ngray_lev-1]), title("Gray reduced with "+ngray_lev+" level");

```