

Digital Image Processing

Lab #2. Image resize and bwlabel



학과 : 전자공학과

학번 : 21611648

이름 : 유준상

담당교수 : 김성호

Table of Contents

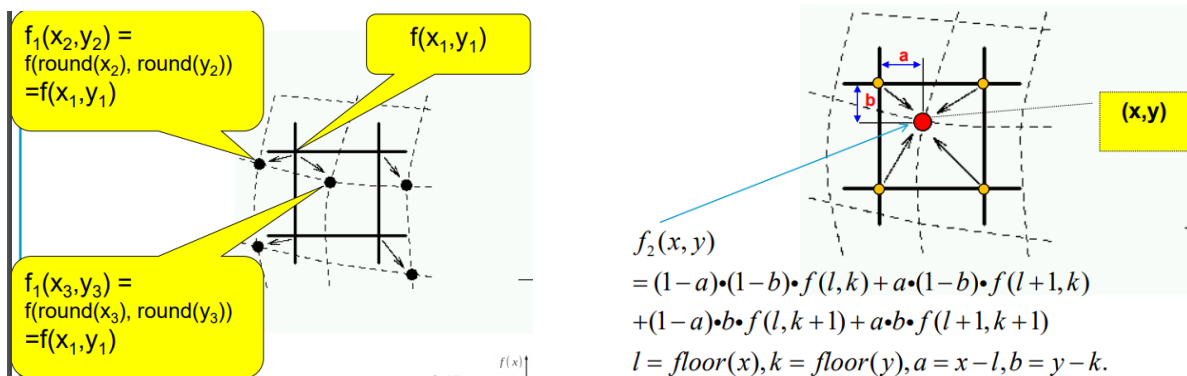
1. Introduction

2. Experiment

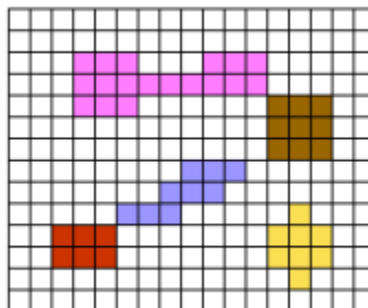
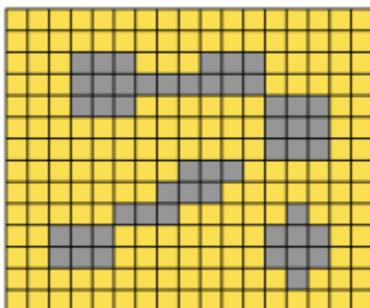
3. Conclusion

1. Introduction

이번 Lab #2에서는 interpolation을 활용한 이미지 resize와 blob labeling에 대해 코드를 직접 작성한다. 두 interpolation은 Nearest Neighbor와 Bilinear이다. 간단히 설명하자면, NN은 resize할 이미지의 한 점의 좌표에 대해 원본 이미지에서 제일 가까운 점으로 반올림을 사용해서 interpolation하는 방법이다. Bilinear는 인접 4점 또는 8점에 대해 interpolation하는 방법이다.



Blob labeling은 한 픽셀이 1의 값을 가지고 있다면, adjacency를 설정한 것에 따라 인접한 픽셀들이 1을 가지고 있다면 같이 라벨링하는 것이다. 한 픽셀 값에 대해 이어지는 인접 픽셀들이 이어지지 않을 때까지 탐색과 라벨링을 계속한다.



Lab. 2-1 : imresize

1) source code

```
편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\lab2_1.m
lab2_4.m x lab2_1.m x lab2_2.m x lab2_3.m x myResizeNN.m x +
1 - I=imread('jsyoo.png'); % read an gray image
2 - figure(1); % make a window
3 - imshow(I); % display the image
4 - title('Input image by JSYoo-21611648','fontsize',14);
5
6 %% image resize processing using resampling
7 % Nearest Neighbor
8 - Inn=imresize(I,0.4,'nearest');
9 - Ibil=imresize(I,0.4,'bilinear');
10 - Ibic=imresize(I,0.4,'bicubic');
11 - figure(2);
12 - subplot(1,3,1); imshow(Inn); |
13 - title('Resizing using NN by JSYoo-21611648','fontsize',14);
14 - subplot(1,3,2); imshow(Ibil);
15 - title('Resizing using Bilinear by JSYoo-21611648','fontsize',14);
16 - subplot(1,3,3); imshow(Ibic);
17 - title('Resizing using Bicubic by JSYoo-21611648','fontsize',14);
```

2) result

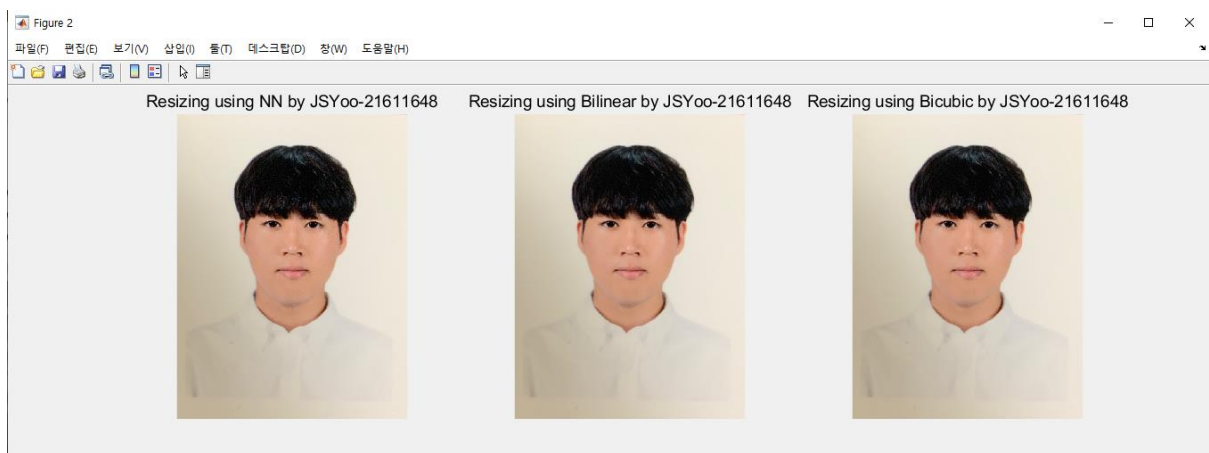
```
>> size(I)
ans =
    2030    1534     3

>> size(Inn)
ans =
     812     614     3

>> size(Ibil)
ans =
     812     614     3

>> size(Ibic)
ans =
     812     614     3
```

3) result figure



4) discussions

원본 이미지를 figure 1에 디스플레이 한 후, figure 2를 생성하였습니다. `imresize`(built-in function)을 사용하여 Nearest Neighbor, Bilinear, Bicubic 세가지 interpolation method를 사용하여 원본 이미지를 크기 조정하여 디스플레이 했습니다. 크기 조정이 잘 되었는지 확인하기에 위해 size를 통해 확인하였고 제대로 되었습니다.

Lab. 2-2 : imresize

1) source code

1. Nearest Neighbor

```
편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\myResizeNN.m
lab2_1.m x lab2_2.m x lab2_3.m x myResizeBil.m x myResizeNN.m x
1 % input argument : i = image, scale = resize scale
2 function inn=myResizeNN(i,scale)
3 [row,col,dim]=size(i);
4 out_row=row*scale; out_col=col*scale;
5 i=[out_row,out_col,dim];
6
7 for d=1:dim
8 for r=1:out_row
9 for c=1:out_col
10 i(r,c,d)=i(round(r/scale),round(c/scale),d);
11 end
12 end
13 end
14 inn=uint8(i);
15 end
```

```
편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\lab2_2.m
lab2_1.m x lab2_2.m x lab2_3.m x myResizeBil.m x my
1 %% Resize NN
2 figure(1); % make a window
3 I=imread('jsyoo.png');
4 imshow(I); title('Original Image by JSYoo-21611648');
5 % imreszie(built-in func)
6 figure(2);
7 inn=imresize(I,0.84,'nearest');
8 imshow(inn); title('imresize func NN by JSYoo-21611648');
9 % made by jsyoo
10 figure(3);
11 my_inn=myResizeNN(I,0.84);
12 imshow(my_inn); title('My Resize NN by JSYoo-21611648');
```

2. Bilinear

```
편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\my_ResizeBil.m
lab2_4.m x lab2_1.m x lab2_2.m x lab2_3.m x detect.m x my_ResizeBil.m x
1 function ibil=my_ResizeBil(im,scale)
2 %원본 영상의 크기 읽어들이기
3 [r c d]=size(im); %r:original rows, c:original cols, d:image data
4 %s배만큼 scale된 영상의 너비,높이
5 out_row=floor(scale*r); %resized row
6 out_col=floor(scale*c); %resized col
7 out=zeros(out_row,out_col,d); %출력시킬 영상값 초기화
8 for i=1:out_row
9 x1=cast(floor(i/scale),'uint32'); %uint32 : 32bits 정수형 반환
10 x2=cast(ceil(i/scale),'uint32');
11 if x1==0 %매트랩은 (1,1)부터이므로 x1,y1값이 0일 때 1로 지정해준다.
12 x1=1;
13 end
14 x=rem(i/scale,1);
15 for j=1:out_col
16 y1=cast(floor(j/scale),'uint32');
17 y2=cast(ceil(j/scale),'uint32');
18 if y1==0
19 y1=1;
20 end
21 y=rem(j/scale,1);
22 %밝기값을 구하고자 하는 픽셀에서 가장 가까운 네 픽셀의 좌표
23 f1=im(x1,y1,:);
24 f2=im(x2,y1,:);
25 f3=im(x1,y2,:);
26 f4=im(x2,y2,:);
27 %수학식을 이용한 Bilinear Interpolation
28 out(i,j,:)=f1*(1-x)*(1-y)+f2*x*(1-y)+f3*(1-x)*y+f4*x*y;
29 end
30 end
31 %scale된 영상 출력하기
32 ibil=cast(out,'uint8'); %8bits 정수형 반환
33 end
```

```

13 %% Resize Bilinear
14 - figure(1); % make a new window
15 - imshow(I); title('Original Image by JSYoo-21611648');
16 % imresize(built-in func)
17 - figure(2);
18 - ibil=imresize(I,0.84,'bilinear');
19 - imshow(ibil); title('imresize func Bil by JSYoo-21611648');
20 % made by jsyoo
21 - figure(3);
22 - my_ibil=my_ResizeBil(I,0.84);
23 - imshow(my_ibil); title('My Resize Bil by JSYoo-21611648');

```

2) result

1. Nearest Neighbor (x0.84)

```

>> size(I)

ans =

    2030    1534     3

>> size(inn)

ans =

    1706    1289     3

>> size(my_inn)

ans =

    1705    1288     3

>>

```

2. Bilinear (x0.84)

```

>> size(I)

ans =

    2030    1534     3

>> size(ibil)

ans =

    1706    1289     3

>> size(my_ibil)

ans =

    1705    1288     3

>>

```

3. Nearest Neighbor (x1.24)

4. Bilinear (x1.24)


```
>> size(l)

ans =

    2030    1534     3

>> size(inn)

ans =
|
    2518    1903     3

>> size(my_inn)

ans =

    2517    1902     3
```

```
>> size(l)

ans =

    2030    1534     3

>> size(ibil)

ans =

    2518    1903     3

>> size(my_ibil)

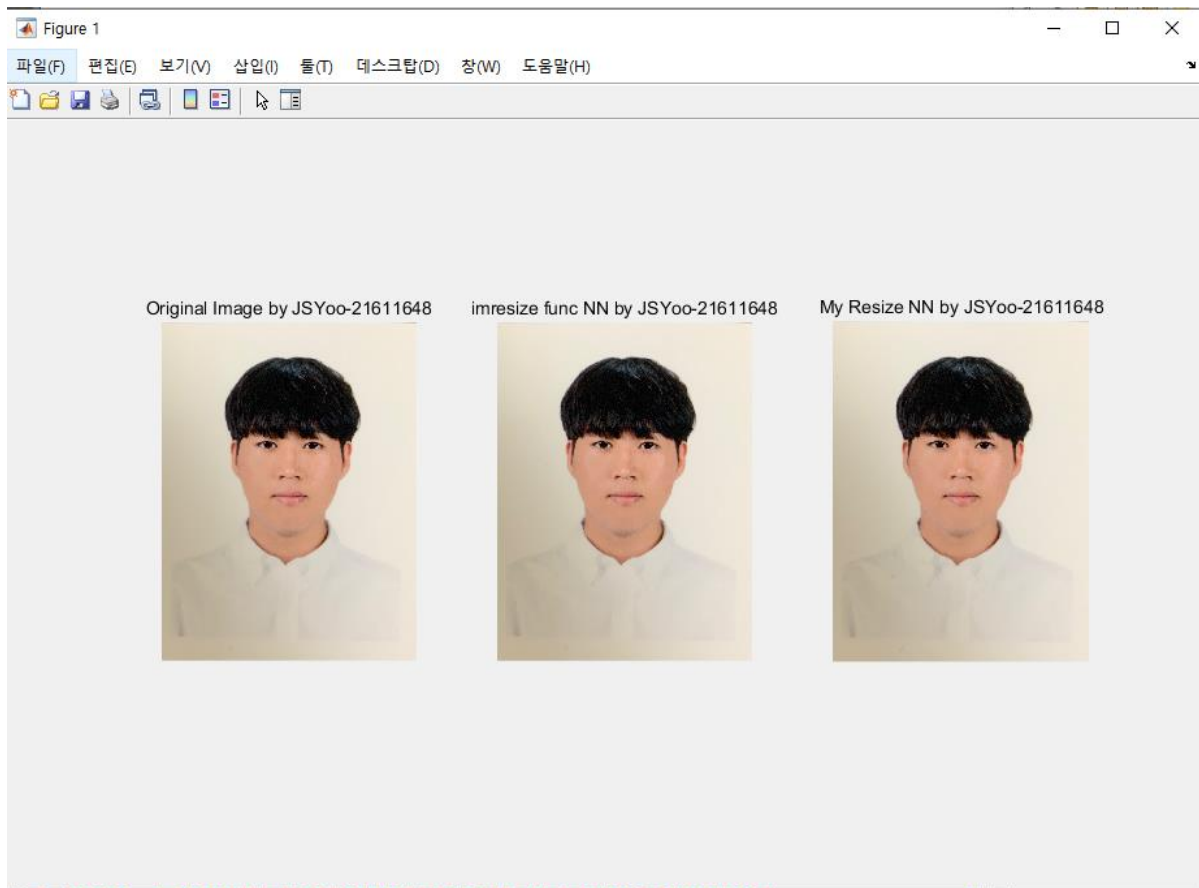
ans =

    2517    1902     3
```

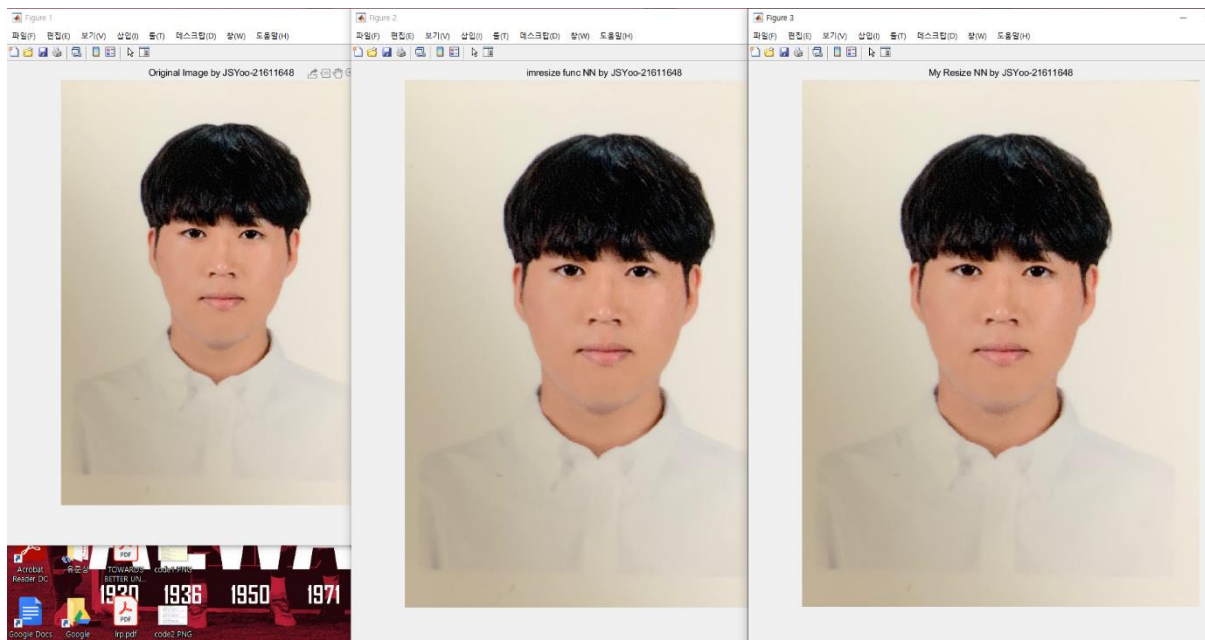
3) result figure

1. Nearest Neighbor

1-1) x0.84

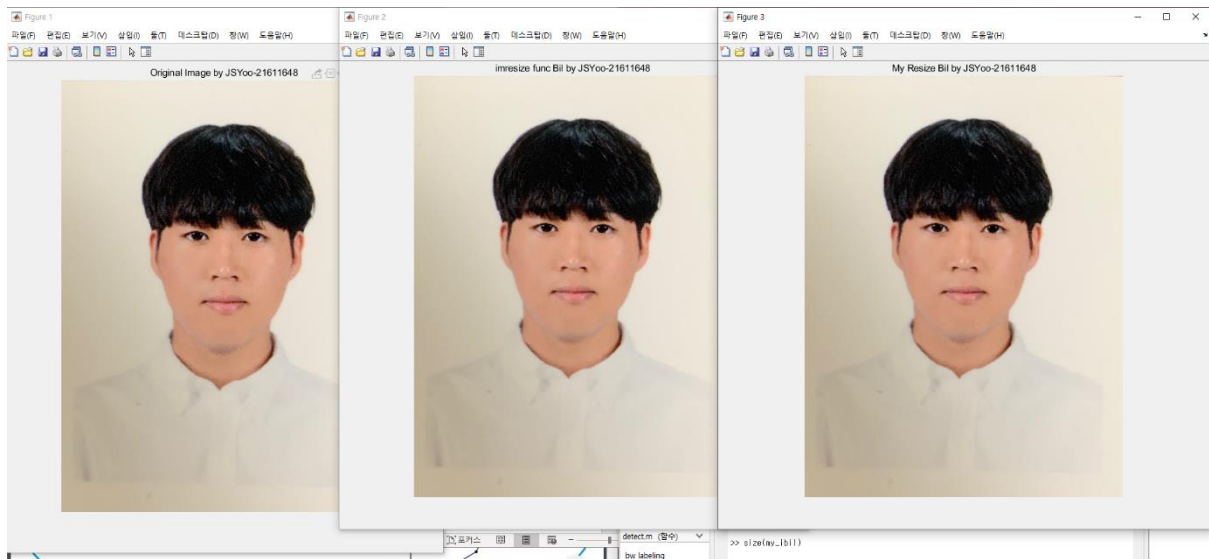


1-2) x1.24

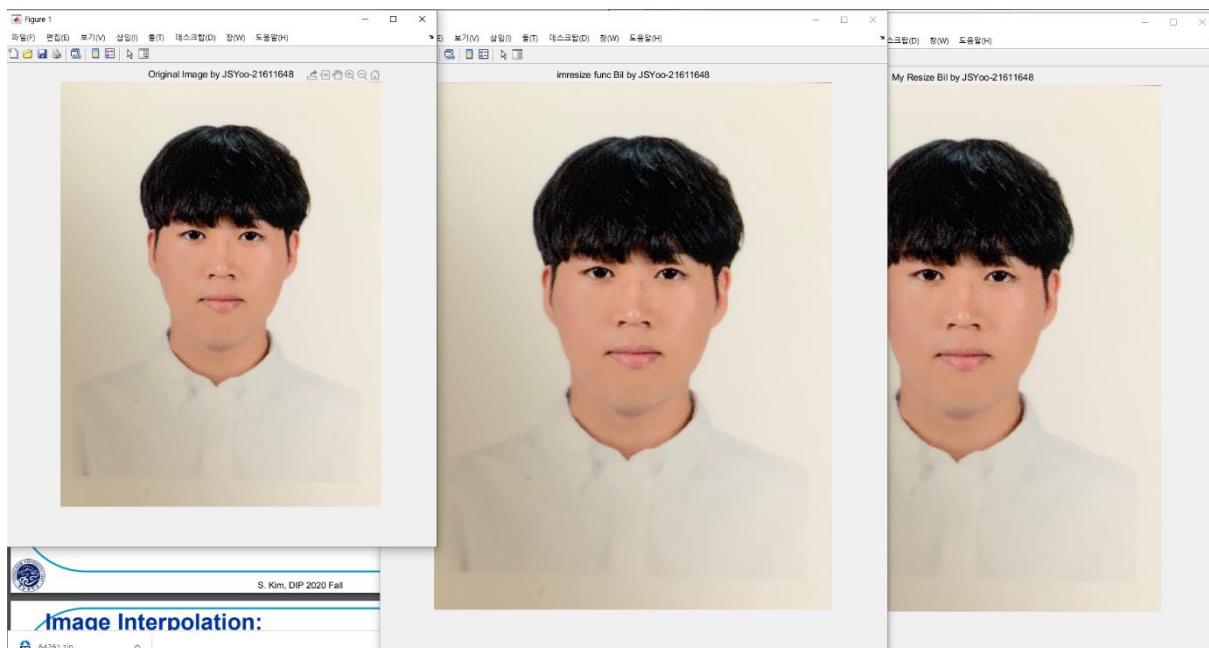


2. Bilinear

2-1) x0.84



2-2) x1.24



4) discussions

원본 이미지와 매트랩에서 제공하는 built-in function인 imresize를 사용한 결과와 직접 만든 함수로 resize한 이미지를 디스플레이 했다. Interpolation 방법으로는 Nearest Neighbor와 Bilinear 두 가지를 이용했다.

Lab. 2-3 : Simple Labeling using bwlabel

1) source code

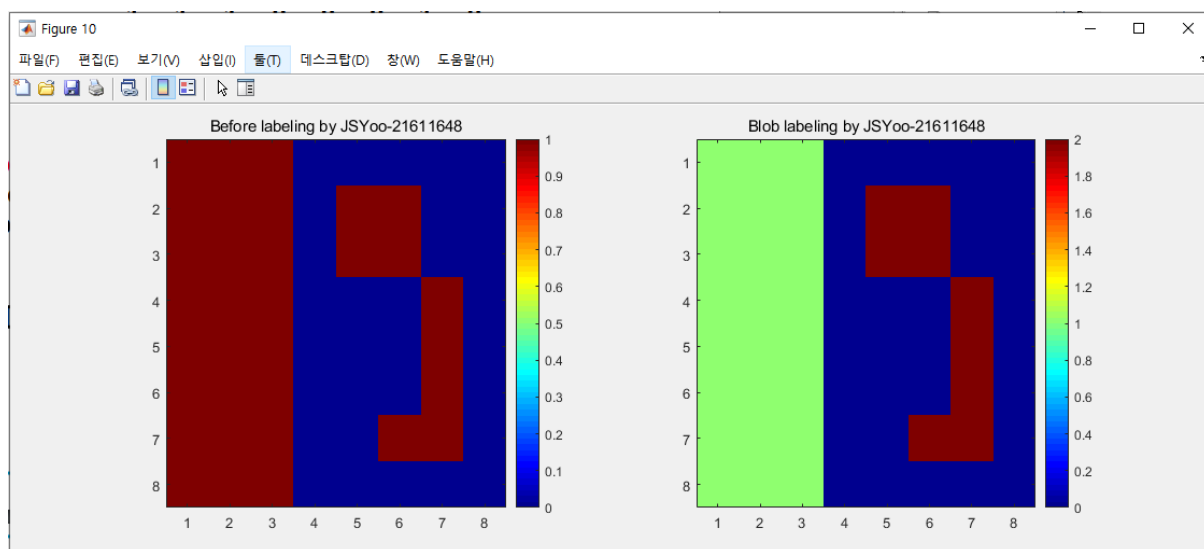
```

편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\lab2_3.m
lab2_4.m x lab2_1.m x lab2_2.m x lab2_3.m x myResizeNN.m x +
1 % source data
2 BW=logical([1 1 1 0 0 0 0 0
3 1 1 1 0 1 1 0 0
4 1 1 1 0 1 1 0 0
5 1 1 1 0 0 0 1 0
6 1 1 1 0 0 0 1 0
7 1 1 1 0 0 0 1 0
8 1 1 1 0 0 1 1 0
9 1 1 1 0 0 0 0 0]);
10
11 L=bwlabel(BW,8); % blob labeling with 4/8 adjacency
12 figure(10); subplot(1,2,1); imagesc(BW), colorbar;
13 title('Before labeling by JSYoo-21611648');
14 subplot(1,2,2); imagesc(L); colormap jet; colorbar;
15 title('Blob labeling by JSYoo-21611648');
16 |
17 [r,c]=find(L==2); % find the index of Label==2
18 rc=[r c];

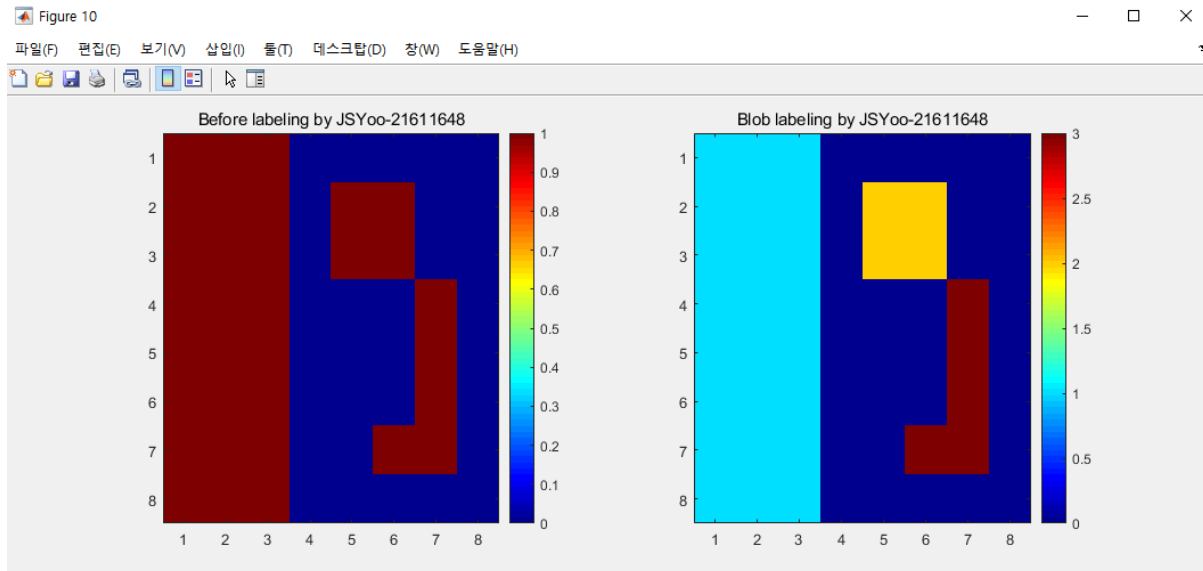
```

2) result figure

8-adjacency



4-adjacency



3) discussions

숫자형 행렬을 논리형 배열로 바꿔주는 logical 을 생성한다. 원본을 subplot 의 왼쪽에 디스플레이 한 후, blob labeling 처리한 것을 subplot 의 오른쪽에 디스플레이 했습니다. 눈으로 보았을 때, 확연히 차이가 보였습니다. 4/8 adjacency 간의 차이를 보기 위해서 두가지 다 출력해 보았습니다.

Lab. 2-4 : Simple Image Labeling using bwlabel

1) source code

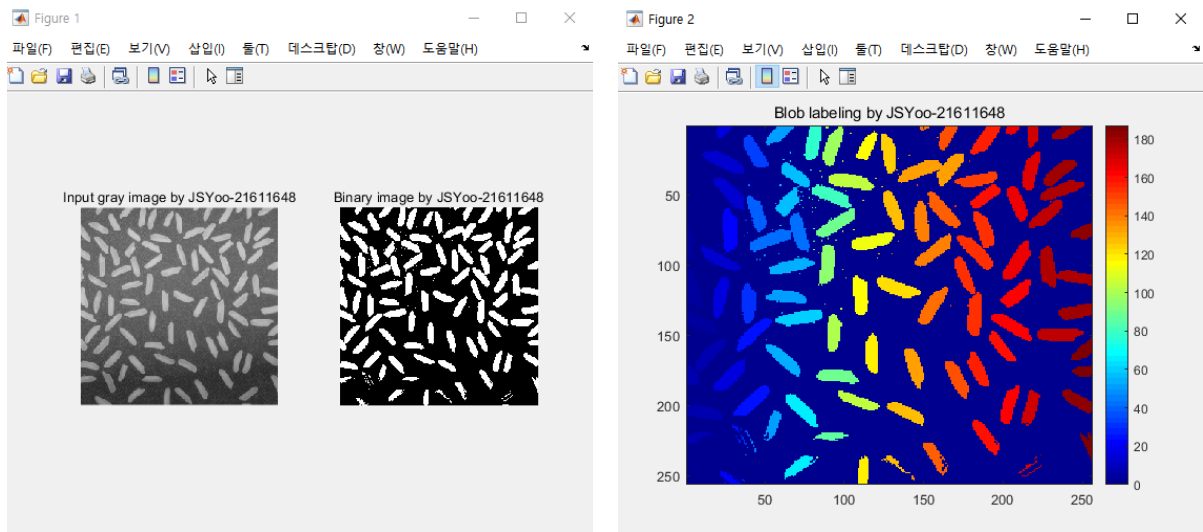
```

편집기 - C:\Users\beatr\Documents\MATLAB\DI\week3\lab2_4.m
lab2_4.m x lab2_1.m x lab2_2.m x lab2_3.m x myResizeNN.m x +
1 % load a gray image
2 I=imread('rice.png');
3 Ig=rgb2gray(I);
4 figure(1), subplot(1,2,1); imshow(I);
5 title('Input gray image by JSYoo-21611648');
6
7 % thresholding
8 brightBlobs=I>128; % Find bright things. 미진화시키기
9 figure(1), subplot(1,2,2); imshow(brightBlobs);
10 title('Binary image by JSYoo-21611648');
11
12 % labeling
13 L2=bwlabel(brightBlobs,8); % blob labeling with 8 adjacency
14 figure(2), imagesc(L2); colormap jet
15 title('Blob labeling by JSYoo-21611648'); colorbar;

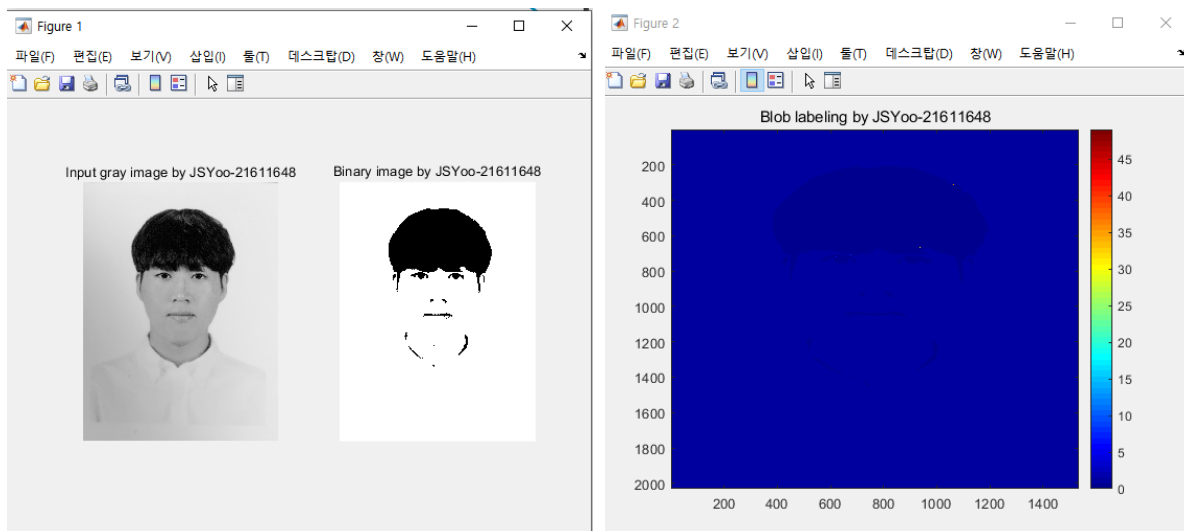
```

2) result figure

1. rice.png



2. my photo



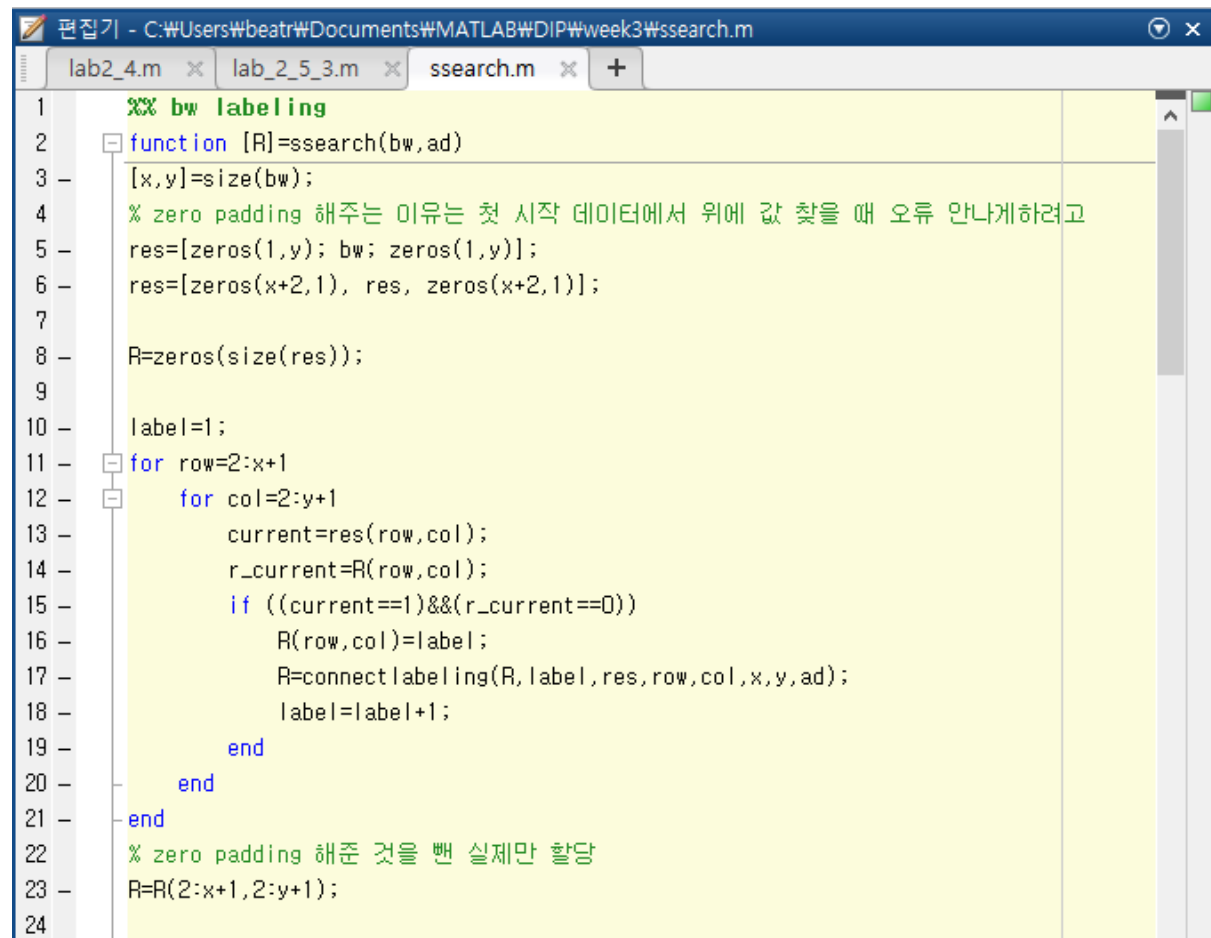
4) discussions

먼저 matlab 에서 제공하는 rice.png 이미지를 이용해보았습니다. 원본 이미지를 불러오고, 원본 이미지를 threshold 값으로 128 을 정해서 이진화시킨 후, 원본 이미지와 이진화된 이미지를 디스플레이 했습니다. 그 후 bwalabel 을 이용하여 라벨링한 후 디스플레이 했습니다. 제 사진을 가지고도 실습해보았습니다. 이진화 까지는 무난하게 된 것으로 보이지만, 라벨링 과정에서 이미지의 구성 때문에 라벨링해서 디스플레이해서 bwalabel 을 이해하기에는 부적절한 것이라고 생각했습니다.

Lab. 2-5 : Simple Image Labeling using my code.

1. lab 2-3 using my code

1) source code



The image shows a MATLAB script editor window titled '편집기 - C:\Users\beatr\Documents\MATLAB\DI\week3\ssearch.m'. The editor contains the following code:

```
1 %% bw labeling
2 function [R]=ssearch(bw,ad)
3 [x,y]=size(bw);
4 % zero padding 해주는 이유는 첫 시작 데이터에서 위에 값 찾을 때 오류 만나게하려고
5 res=[zeros(1,y); bw; zeros(1,y)];
6 res=[zeros(x+2,1), res, zeros(x+2,1)];
7
8 R=zeros(size(res));
9
10 label=1;
11 for row=2:x+1
12     for col=2:y+1
13         current=res(row,col);
14         r_current=R(row,col);
15         if ((current==1)&&(r_current==0))
16             R(row,col)=label;
17             R=connect_labeling(R, label, res, row, col, x, y, ad);
18             label=label+1;
19         end
20     end
21 end
22 % zero padding 해준 것을 뺀 실제만 할당
23 R=R(2:x+1,2:y+1);
24
```

```
편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\ssearch.m
lab2_4.m x lab_2_5_3.m x ssearch.m x +
25 %% Find connected Labels around current coordinate
26 function [R]=connectlabeling(R, label, res, row, col, x, y,ad)
27 % 4-adjacency
28 if ad==4
29     a=res(row+1,col); b=res(row-1, col); c=res(row,col+1); d=res(row,col-1);
30     ra=R(row+1,col); rb=R(row-1, col);
31     rc=R(row,col+1); rd=R(row,col-1);
32
33     if ((a==1)&&(ra==0))
34         R(row+1,col)=label;
35         [R]=connectlabeling(R, label, res, row+1, col, x, y,ad);
36     end
37     if ((b==1)&&(rb==0))
38         R(row-1,col)=label;
39         [R]=connectlabeling(R, label, res, row-1, col, x, y,ad);
40     end
41     if ((c==1)&&(rc==0))
42         R(row,col+1)=label;
43         [R]=connectlabeling(R, label, res, row, col+1, x, y,ad);
44     end
45     if ((d==1)&&(rd==0))
46         R(row,col-1)=label;
47         [R]=connectlabeling(R, label, res, row, col-1, x, y,ad);
48     end
49 end
```



```

50 % 8-adjacency
51 if ad==8
52     a=res(row+1,col); b=res(row-1, col); c=res(row,col+1); d=res(row,col-1);
53     e=res(row+1,col+1); f=res(row+1,col-1);
54     g=res(row-1,col+1); h=res(row-1,col-1);
55     ra=R(row+1,col); rb=R(row-1, col);
56     rc=R(row,col+1); rd=R(row,col-1);
57     re=R(row+1,col+1); rf=R(row+1, col-1);
58     rg=R(row-1,col+1); rh=R(row-1,col-1);
59
60     if ((a==1)&&(ra==0))
61         R(row+1,col)=label;
62         [R]=connectlabeling(R, label, res, row+1, col, x, y,ad);
63     end
64     if ((b==1)&&(rb==0))
65         R(row-1,col)=label;
66         [R]=connectlabeling(R, label, res, row-1, col, x, y,ad);
67     end
68     if ((c==1)&&(rc==0))
69         R(row,col+1)=label;
70         [R]=connectlabeling(R, label, res, row, col+1, x, y,ad);
71     end
72     if ((d==1)&&(rd==0))
73         R(row,col-1)=label;
74         [R]=connectlabeling(R, label, res, row, col-1, x, y,ad);

```

```

74         [H]=connectlabeling(H, label, res, row, col-1, x, y,ad);
75     end
76     if ((e==1)&&(re==0))
77         R(row+1,col+1)=label;
78         [R]=connectlabeling(R, label, res, row+1, col+1, x, y,ad);
79     end
80     if ((f==1)&&(rf==0))
81         R(row+1,col-1)=label;
82         [R]=connectlabeling(R, label, res, row+1, col-1, x, y,ad);
83     end
84     if ((g==1)&&(rg==0))
85         R(row-1,col+1)=label;
86         [R]=connectlabeling(R, label, res, row-1, col+1, x, y,ad);
87     end
88     if ((h==1)&&(rh==0))
89         R(row-1,col-1)=label;
90         [R]=connectlabeling(R, label, res, row-1, col-1, x, y,ad);
91     end
92 end
93 end
94 end

```

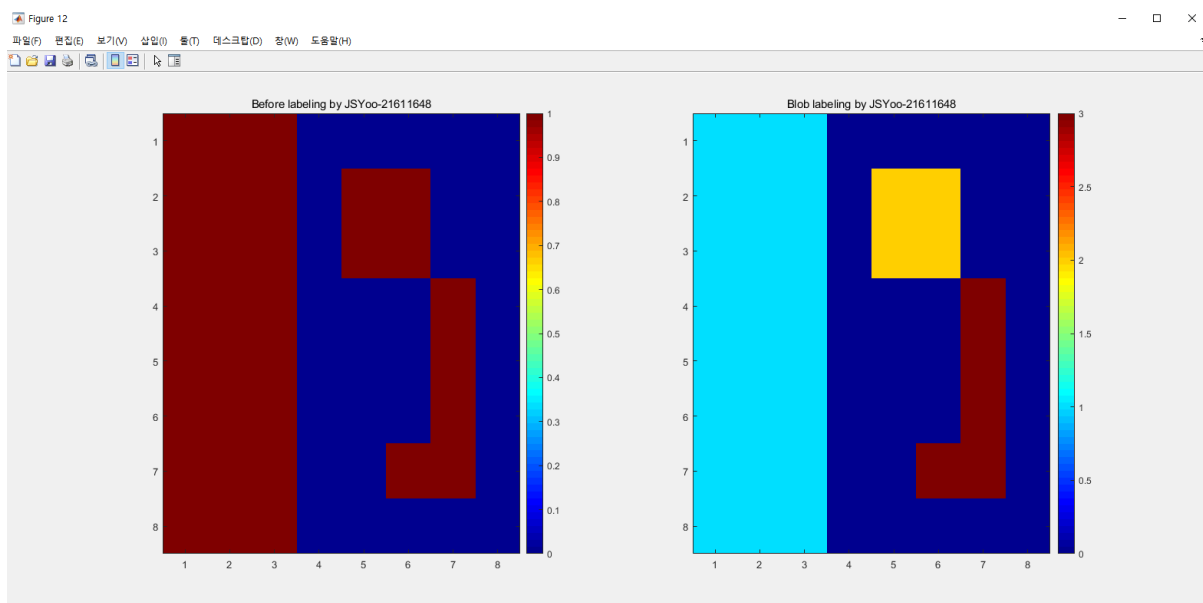
```

편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\lab_2_5_3.m
lab2_4.m x lab_2_5_3.m x ssearch.m x +
1 % source data
2 BW=logical([1 1 1 0 0 0 0 0
3           1 1 1 0 1 1 0 0
4           1 1 1 0 1 1 0 0
5           1 1 1 0 0 0 1 0
6           1 1 1 0 0 0 1 0
7           1 1 1 0 0 0 1 0
8           1 1 1 0 0 1 1 0
9           1 1 1 0 0 0 0 0]);
10
11 L=ssearch(BW,8);
12 figure(11); subplot(1,2,1); imagesc(BW), colorbar;
13 title('Before labeling by JSYoo-21611648');
14 subplot(1,2,2); imagesc(L); colormap jet; colorbar;
15 title('Blob labeling by JSYoo-21611648');

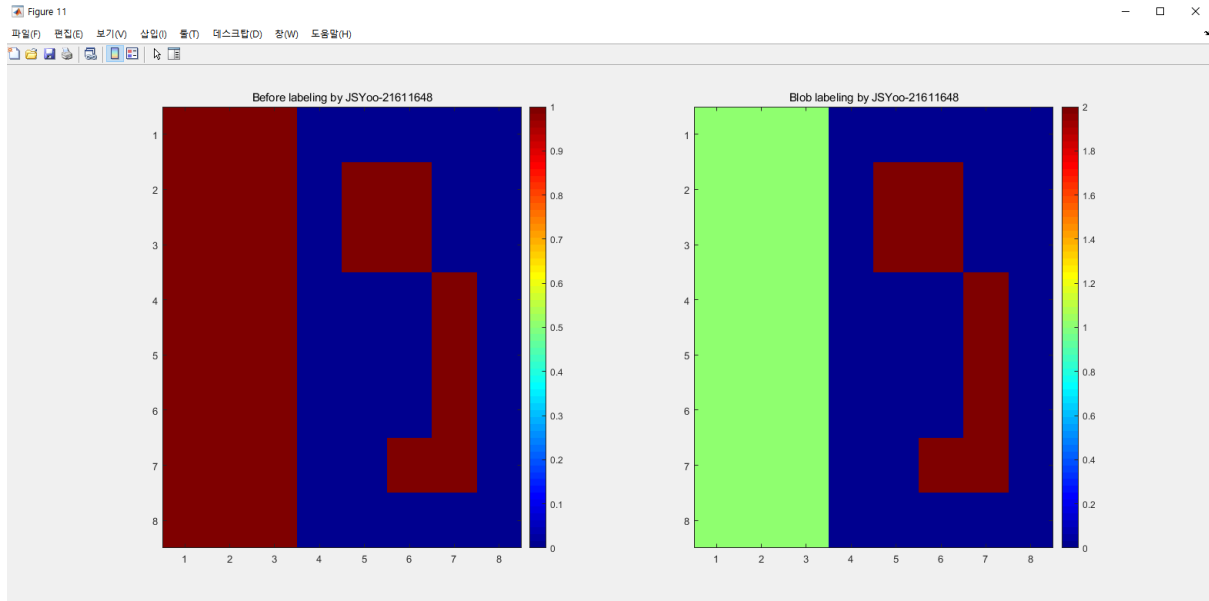
```

2) result figure

2-1. 4adjacency



2-2. 8adjacency



3) discussion

2. lab 2-4 using my code

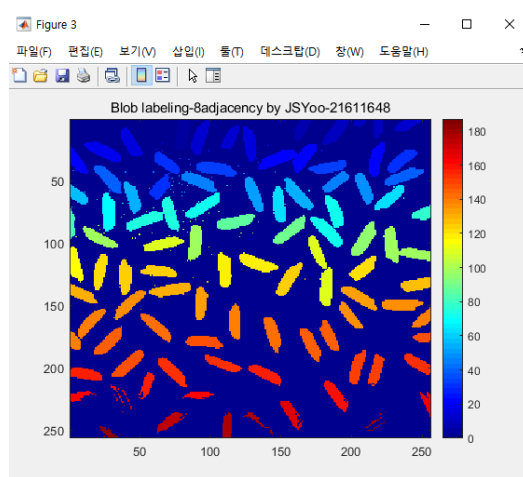
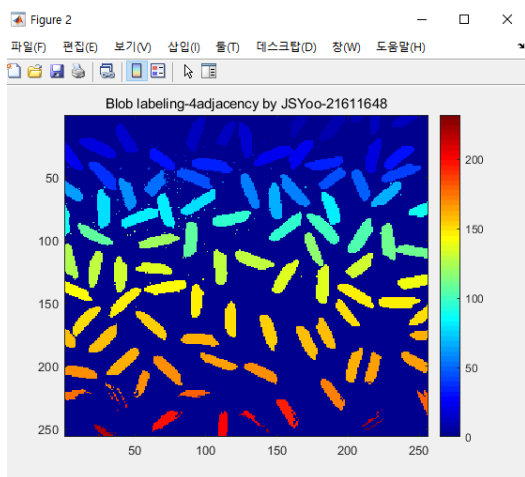
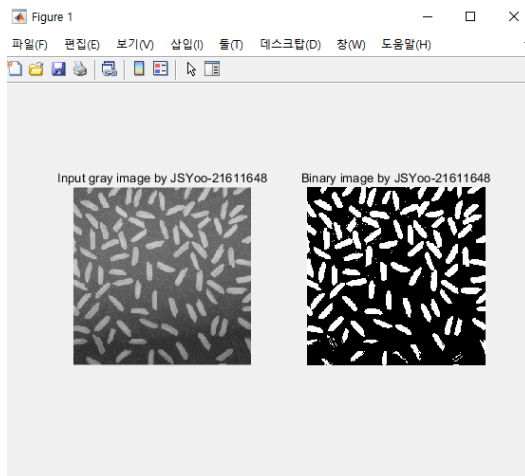
1) source code

```

편집기 - C:\Users\beatr\Documents\MATLAB\DIP\week3\lab_2_5_4.m
lab_2_5_4.m x ssearch.m x +
1 % load a gray image
2 - I=imread('rice.png');
3 - figure(1), subplot(1,2,1); imshow(I);
4 - title('Input gray image by JSYoo-21611648');
5
6 % thresholding
7 - brightBlobs=I>128; % Find bright things. 이진화시키기
8 - figure(1), subplot(1,2,2); imshow(brightBlobs);
9 - title('Binary image by JSYoo-21611648');
10
11 % labeling
12 - L2=ssearch(brightBlobs,4); % blob labeling with 8 adjacency
13 - figure(2), imagesc(L2); colormap jet
14 - title('Blob labeling-4adjacency by JSYoo-21611648'); colorbar;
15 - L3=ssearch(brightBlobs,8);
16 - figure(3), imagesc(L3); colormap jet
17 - title('Blob labeling-8adjacency by JSYoo-21611648'); colorbar;

```

2) figure



3) discussions

위에서 직접 작성한 코드를 사용하여 blob labeling 한 것을 컬러맵으로 디스플레이 했다.

Conclusion

Built-in function을 사용하지 않고, 이미지를 resize 해보았다. 크게 어렵지 않을 거 같았는데, 막상 해보니까 매우 힘들었다. 특히 blob labeling을 짜보는 과정이 매우 힘들었지만 제일 기억에 남고, 어려운 것을 오랜 노력 끝에 해결하고 나니까 되게 뿌듯하고 자신감이 생깁니다.

김성진, 배성찬 학생과 머리 맞대어 고민해보았습니다.