APPLICATIONS OF MATLAB IN ENGINEERING

Yan-Fu Kuo

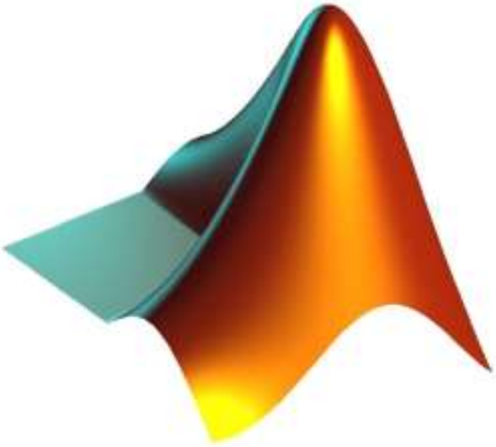
Dept. of Bio-industrial Mechatronics Engineering

National Taiwan University

Fall 2015

Today:

- Image thresholding
- Background image estimation
- Component-connected labeling



Problem Setup

- Count the rice grains and identify their sizes in this image
- What are your strategies?

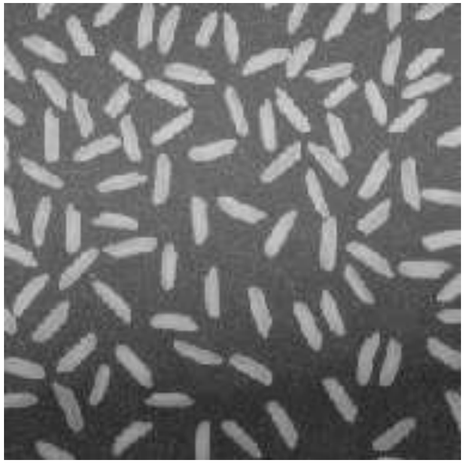
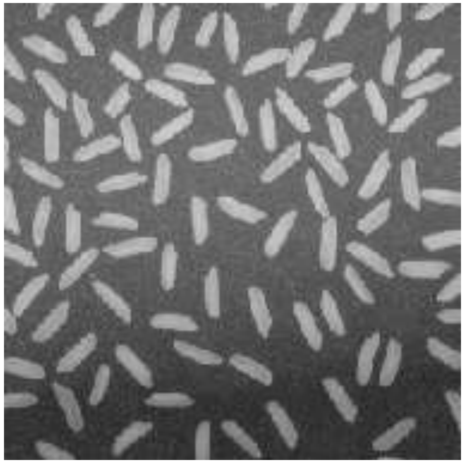
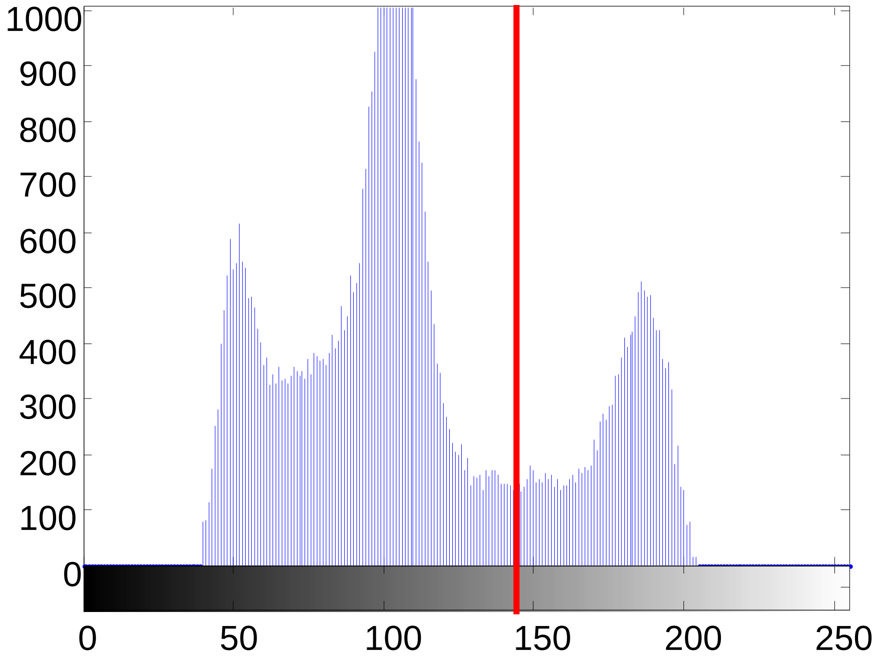


Image Thresholding

- A gray-level image can be turned into a binary image by using a threshold

```
I = imread('rice.png'); imhist(I);
```

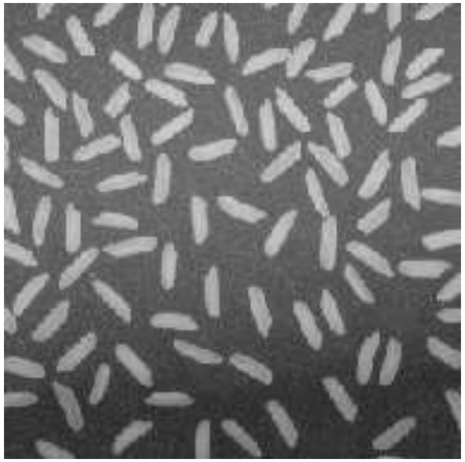


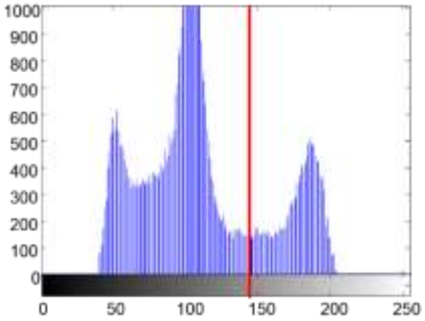


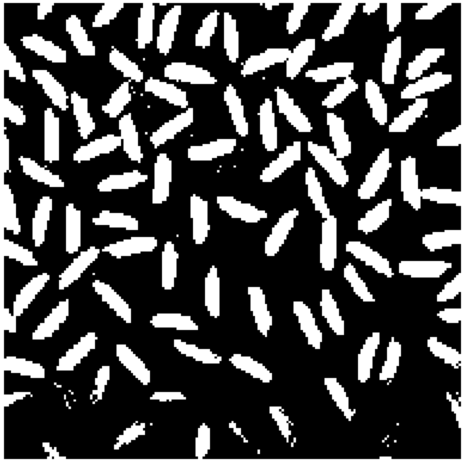
graythresh() and im2bw()

- graythresh() computes an optimal threshold level
- im2bw() converts an images into binary image

```
I = imread('rice.png'); level=graythresh(I);
bw=im2bw(I, level); subplot(1,2,1); imshow(I);
subplot (1,2,2); imshow(bw);
```







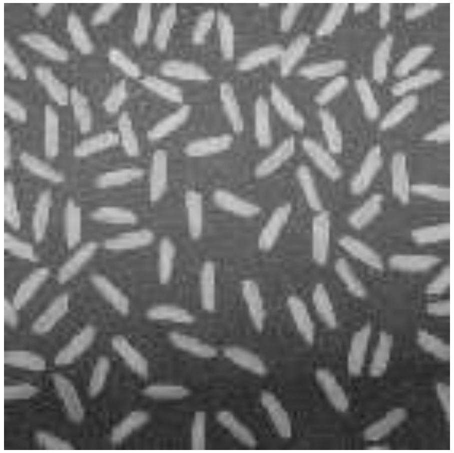
Practice

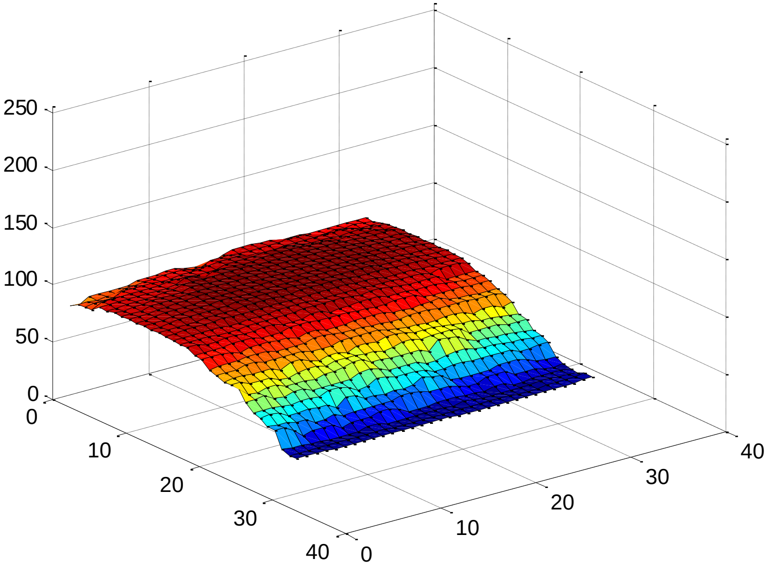
- Write a program to convert the image `rice.png` into a binary image using a threshold
- Do NOT use `im2bw()`
- Try different threshold values to see if you program works

Background Estimation

- Estimation for the gray level of the background :

```
I = imread('rice.png');
BG = imopen(I, strel('disk', 15));
imshow(BG);
```





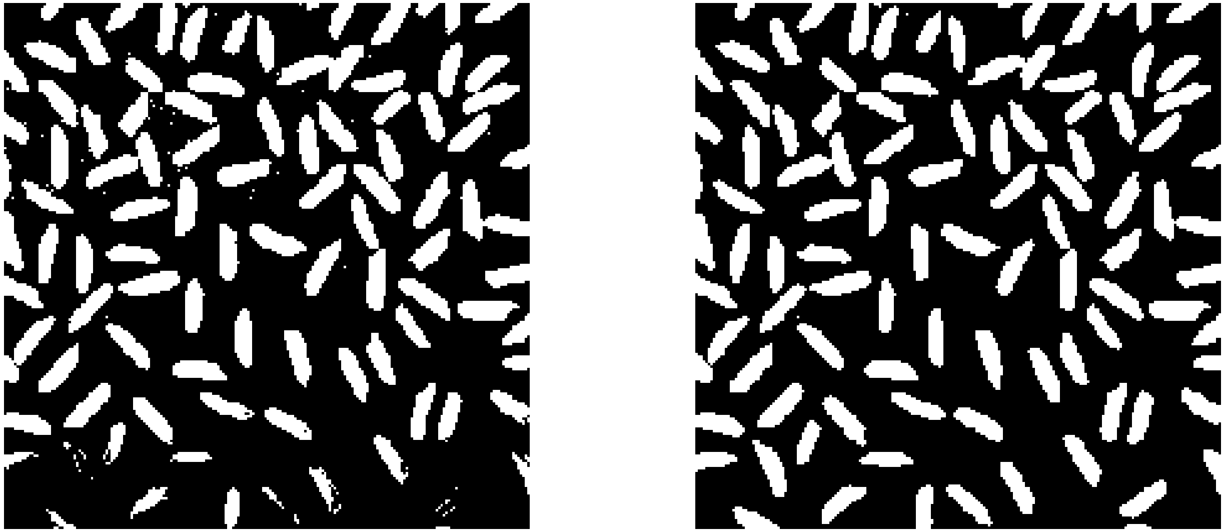
Background Subtraction

```
I = imread('rice.png');
subplot(1,3,1); imshow(I);
BG = imopen(I, strel('disk', 15));
subplot(1,3,2); imshow(BG);
I2 = imsubtract(I, BG);
subplot(1,3,3); imshow(I2);
```

Thresholding on Background Removed Image

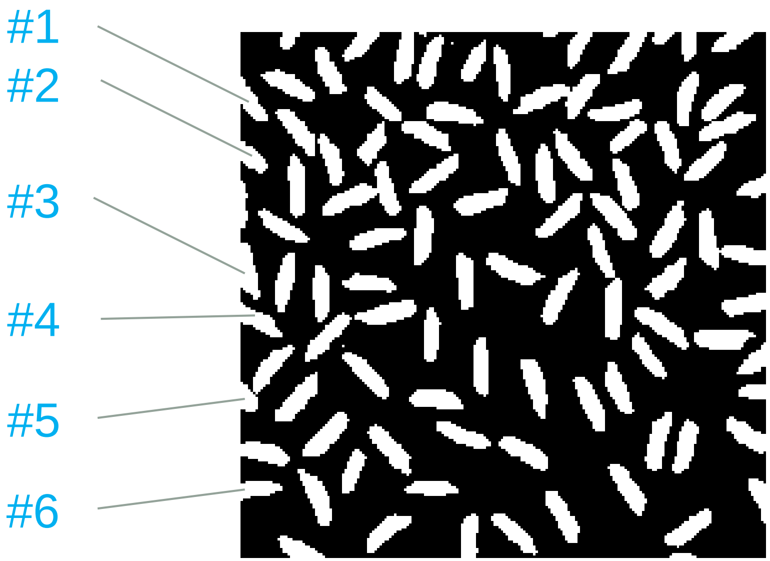
```
I = imread('rice.png'); level=graythresh(I);
bw = im2bw(I, level); subplot (1,2,1);

imshow(bw); BG = imopen(I, strel('disk', 15));
I2 = imsubtract(I, BG); level=graythresh(I2);
bw2 = im2bw(I2, level);
subplot(1,2,2); imshow(bw2);
```



Now What?

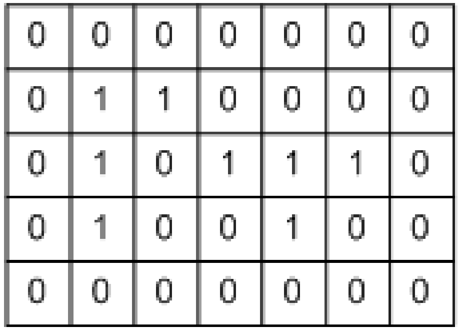
- Want to identify how many grains there in the image
- How?



Connected-component Labeling

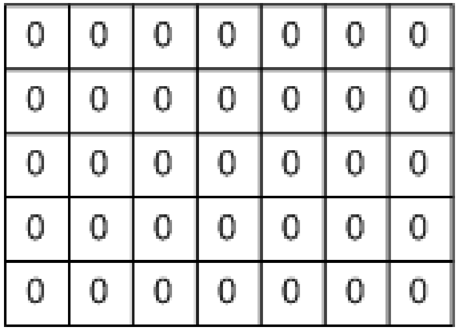
- A procedure for assigning a unique label to each object

Binary image



0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

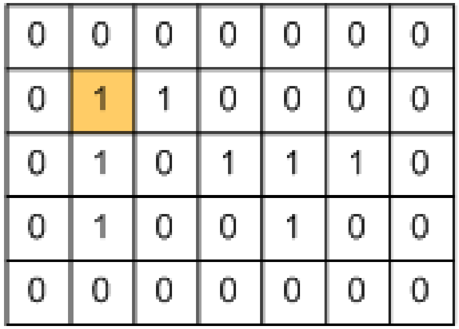
Label matrix



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

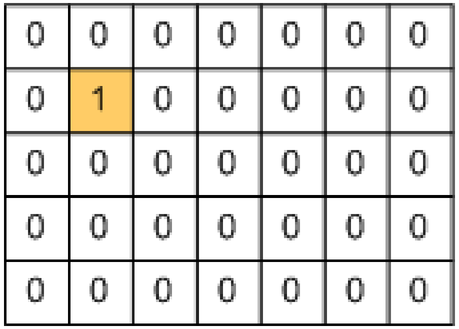
Step 1:

Binary image



0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

Label matrix



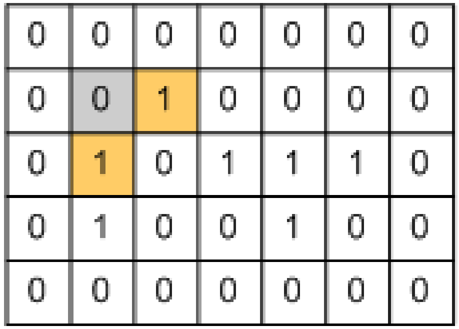
0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Connected-component Labeling (Cont'd)

- Finish labeling of a component

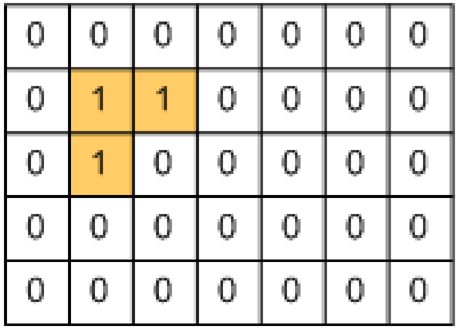
Step 2:

Binary image



0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

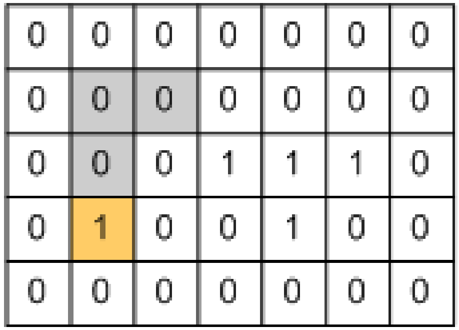
Label matrix



0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

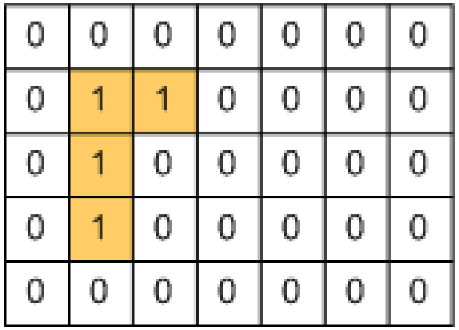
Step 3:

Binary image



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	1	0	0	1	0	0
0	0	0	0	0	0	0

Label matrix



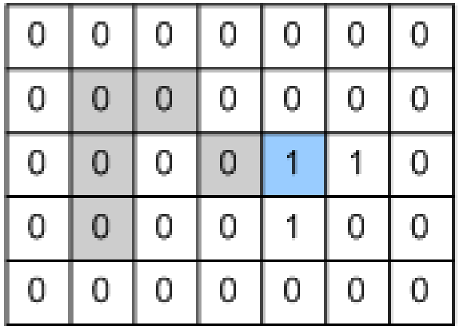
0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

Connected-component Labeling (Cont'd)

- Iterative process until all the pixels are checked

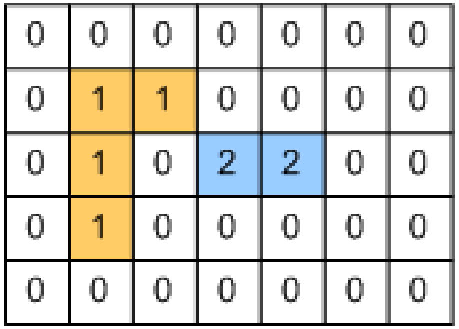
Step 4:

Binary image



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0

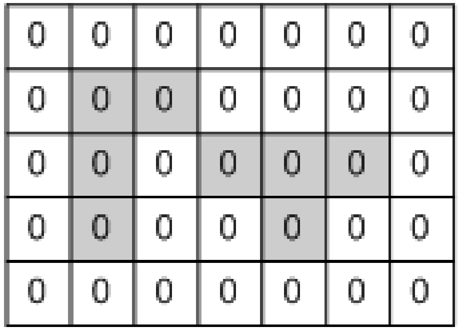
Label matrix



0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

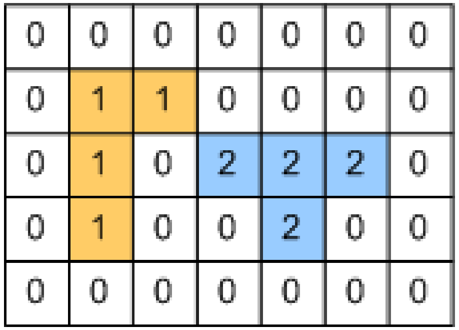
Step 5:

Binary image



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Label matrix



0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	2	0
0	1	0	0	2	0	0
0	0	0	0	0	0	0

Connected-component Labeling: `bwlabel()`

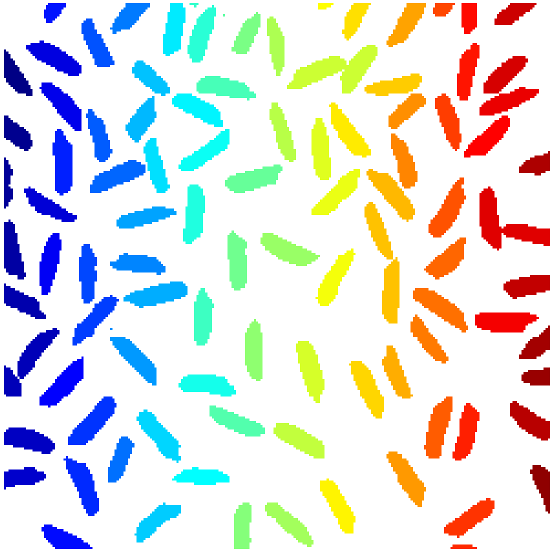
- Built-in connected-component labeling algorithm

```
I=imread('rice.png');
BG=imopen(I, strel('disk', 15));
I2=imsubtract(I, BG); level=graythresh(I2);
BW=im2bw(I2, level);
[labeled, numObjects]=bwlabel(BW, 8);
```

- Check the matrix `labeled`

Color-coding Objects: label2rgb()

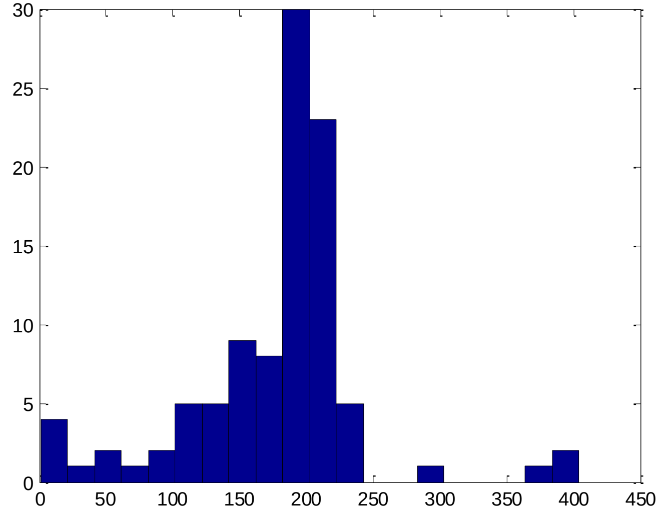
- Converts a label matrix into an RGB color image
- Visualize the labeled regions

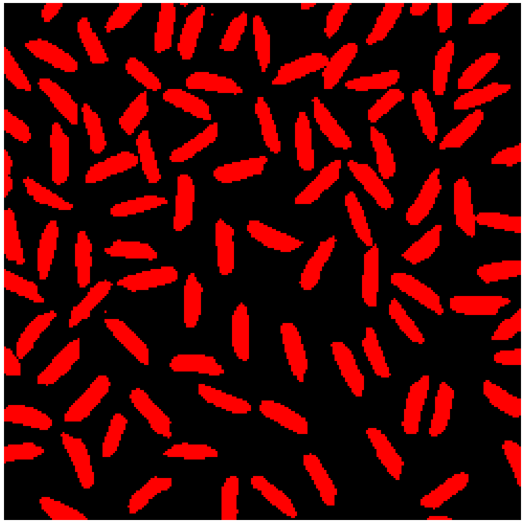


```
I=imread('rice.png');
BG=imopen(I, strel('disk', 15));
I2=imsubtract(I, BG); level=graythresh(I2);
BW=im2bw(I2, level);
[labeled, numObjects]=bwlabel(BW, 8);
RGB_label=label2rgb(labeled); imshow(RGB_label);
```

Practice

- Plot the histogram of grain size
- Identify all the grains in the image by painting them in red





- Wait, is it perfect?

Object Properties: `regionprops()`

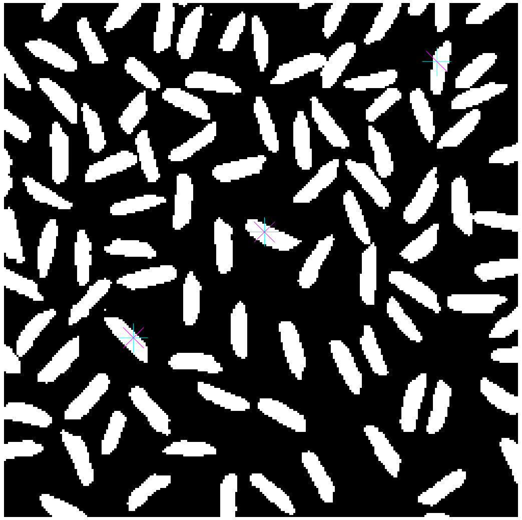
- Provides a set of `properties` for each connected component
- Example:

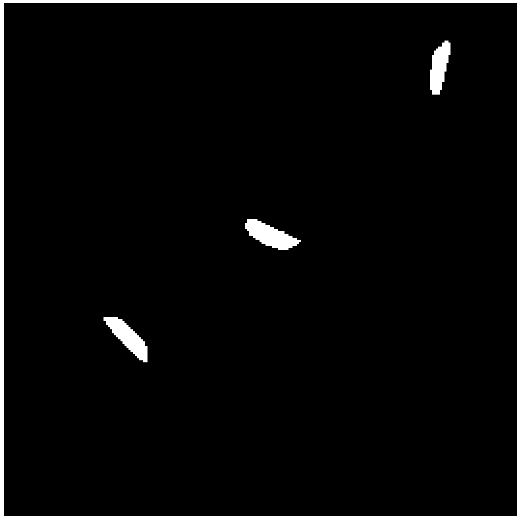
```
I=imread('rice.png');
BG=imopen(I, strel('disk', 15));
I2=imsubtract(I, BG); level=graythresh(I2);
BW=im2bw(I2, level);
[labeled, numObjects]=bwlabel(BW, 8);
graindata = regionprops(labeled, 'basic');
graindata(51)
```

Interactive Selection: `bwselect()`

- Lets you select objects using the mouse

```
I=imread('rice.png'); level=graythresh(I);
BG=imopen(I, strel('disk', 15));
I2=imsubtract(I, BG); BW=im2bw(I2, graythresh(I2));
ObjI = bwselect(BW); imshow(ObjI);
```





End of Class