Lecture 18:

Intensity histograms and Otsu's method

Lecturer: Jian Wei Tay

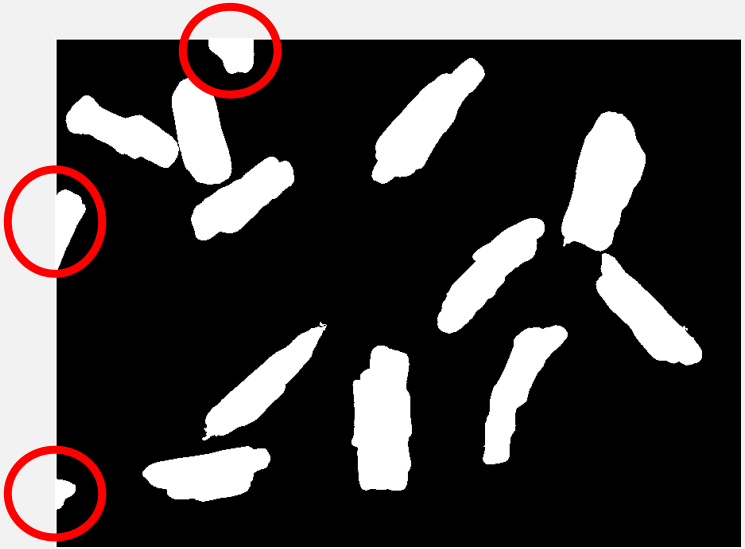
Date: 29 September 2021

Learning objectives

- Image intensity histograms
- Automatic threshold finding with Otsu’s method

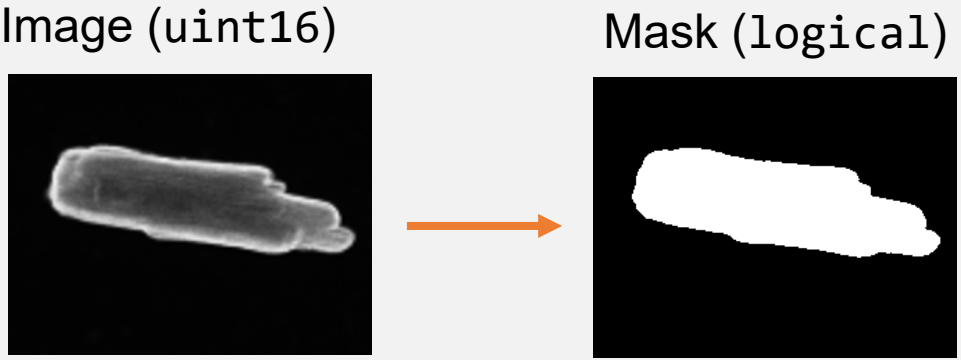
For problem set 5

- To remove masks of objects that are only half in the field of view, you can use `imclearborder(mask)`



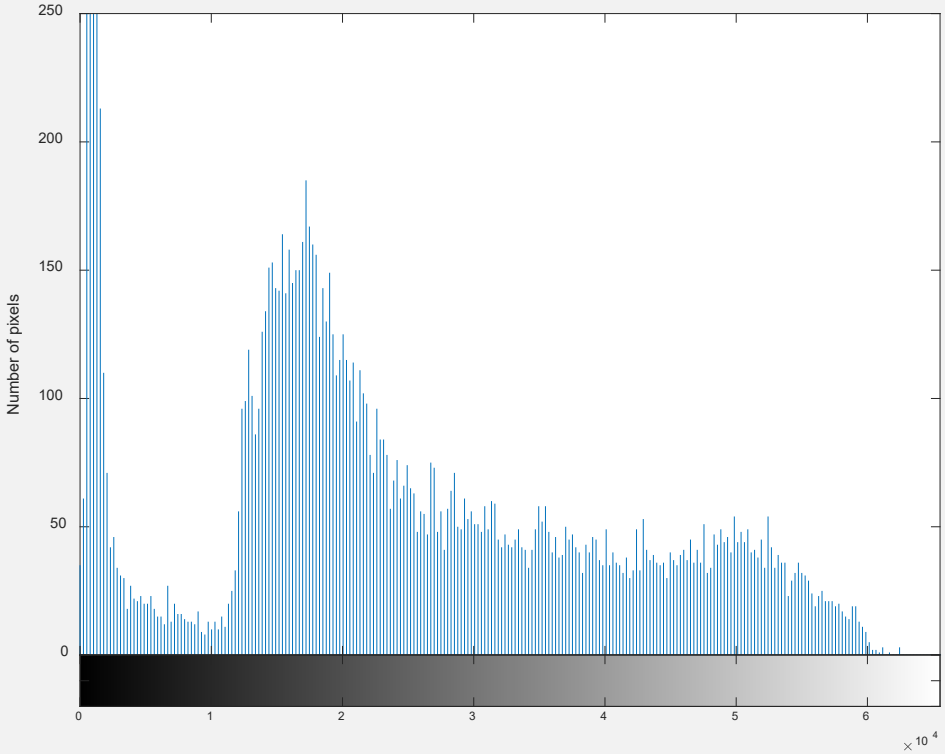
Please don't use `imbinarize` for PS 5

Last week



- Manually choosing a threshold
- Using `improfile` to get an intensity profile

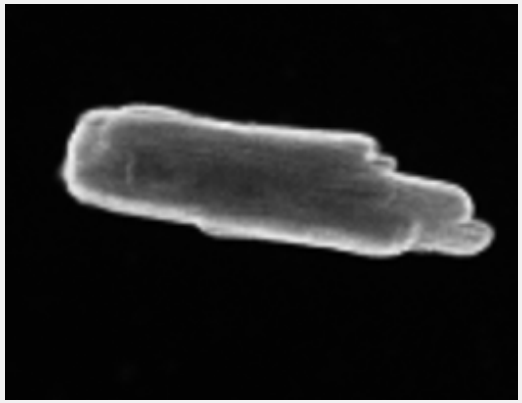
Image intensity histogram



- Shows the distribution of pixel values (intensities) in the image
- X-axis is pixel value or grayscale bin
- Y-axis is number of pixels

Image intensity histogram

Which parts of the histogram correspond to the background/cell?



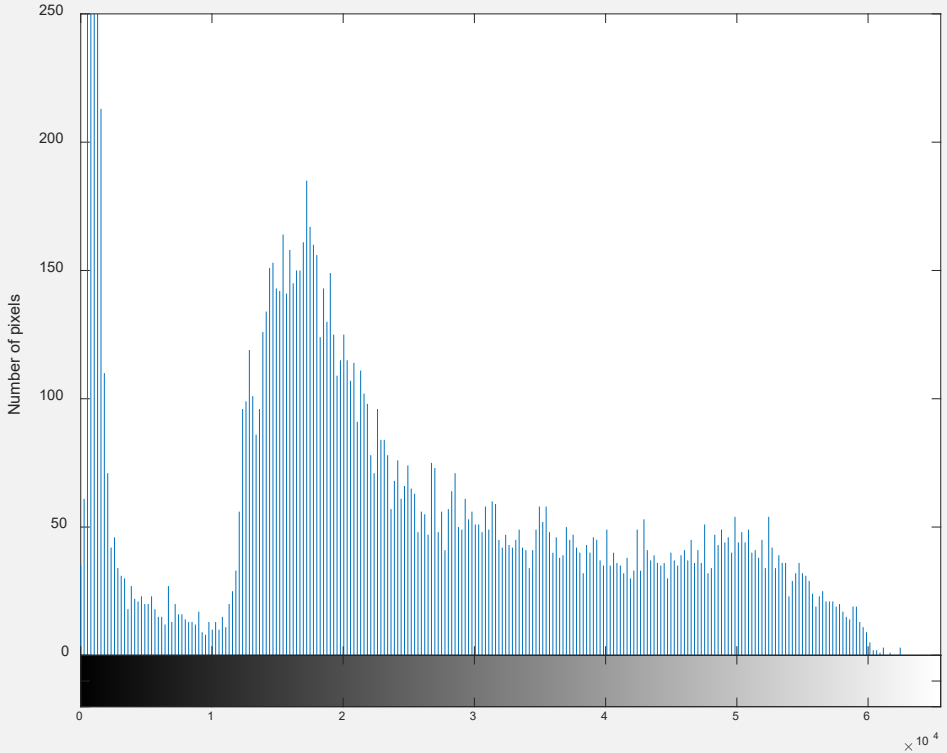
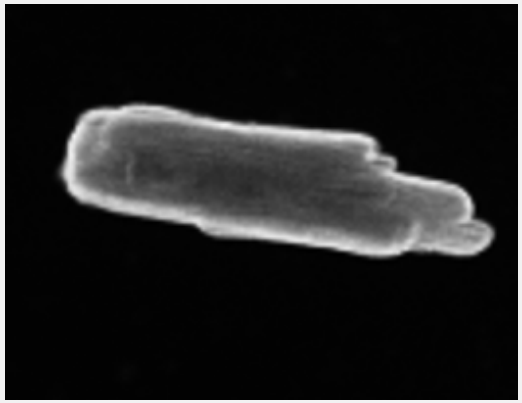
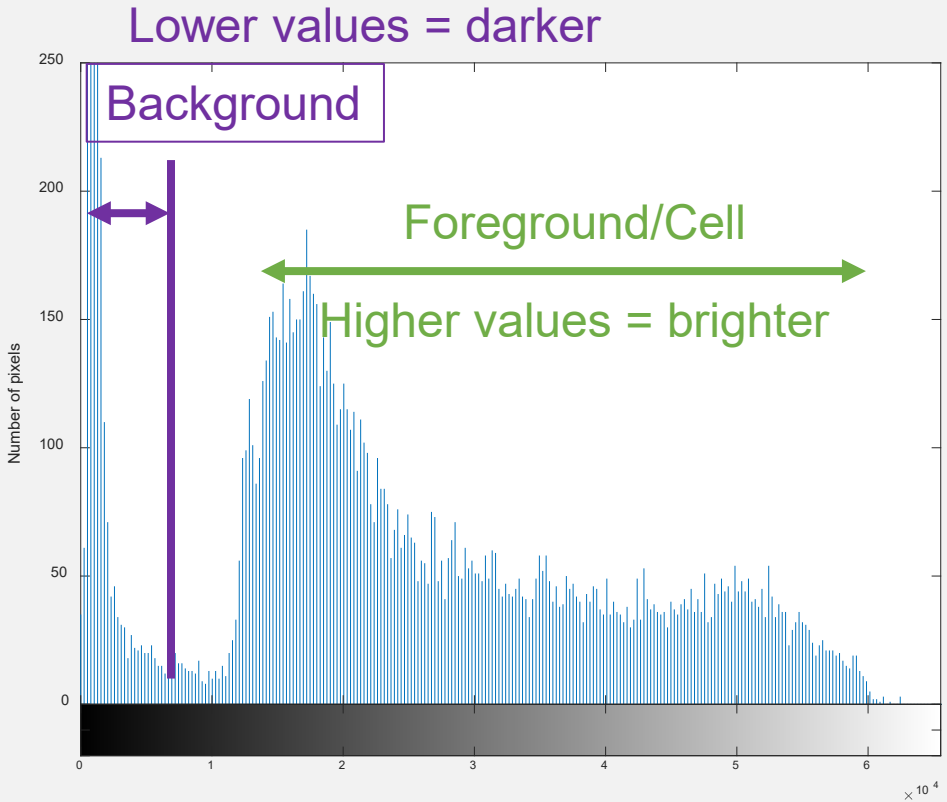


Image intensity histogram

Which parts of the histogram correspond to the background/cell?





Practice

- Read in the image `l11_moreCardiomyocytes.tif` from last week

- Plot an image intensity histogram, using

```
imhist(I)
```

Note: You can also use the regular histogram function `histogram(I(:))`

Useful plot manipulation functions

- To zoom in on the y-axis, use

```
ylim([min max])
```

Example:

```
ylim([0 150])
```

Note: You can use the function `xlim` to zoom in on the x-axis

Useful plot manipulation functions

- To label the y-axis, use

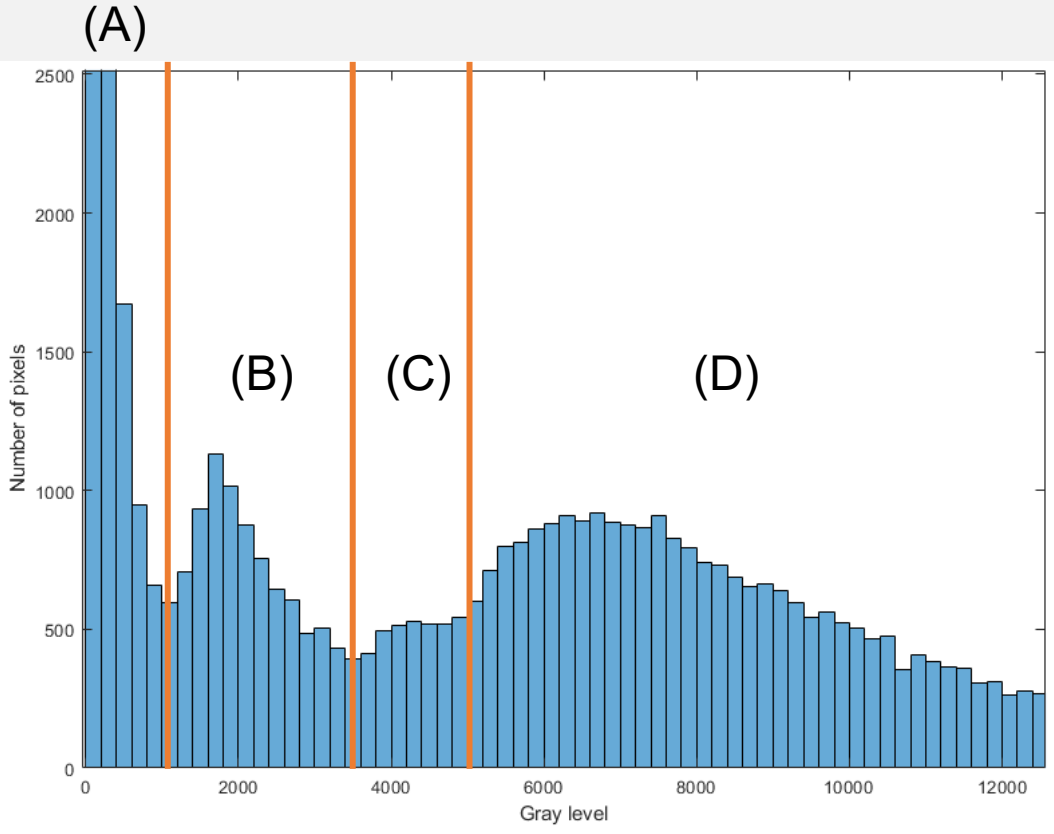
```
ylabel('Text')
```

Example:

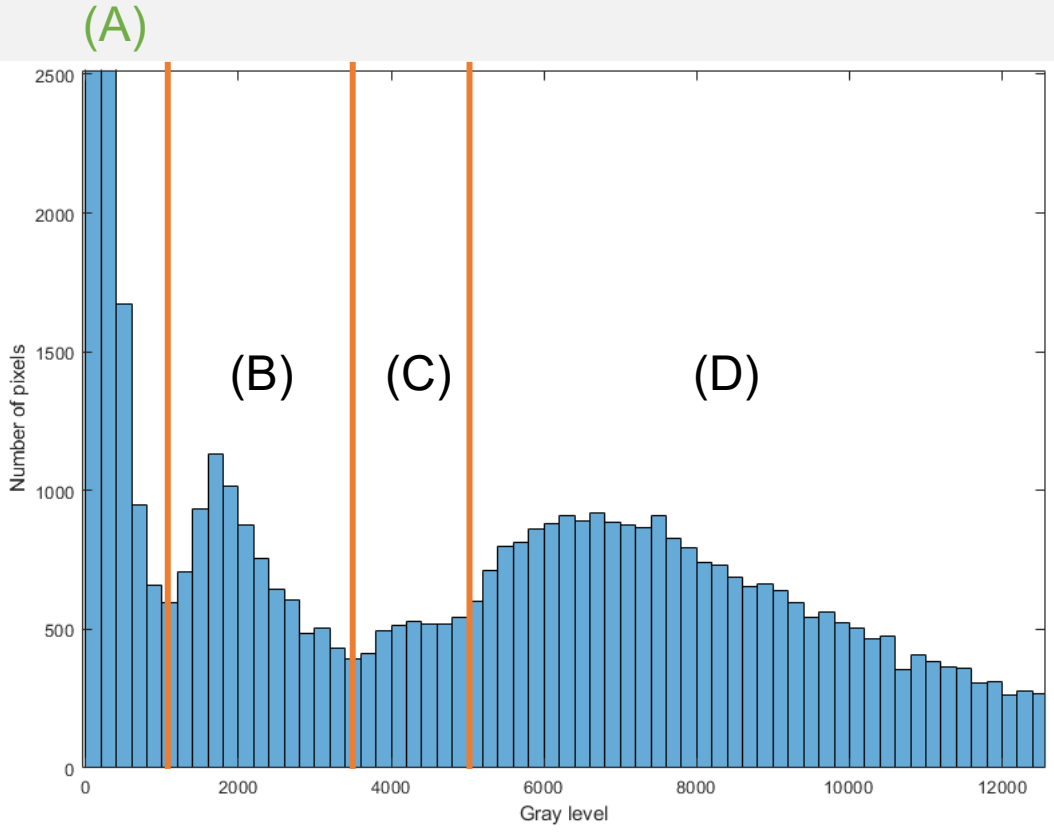
```
ylabel('Number of pixels')
```

Note: You can use the function `xlabel` to label the x-axis

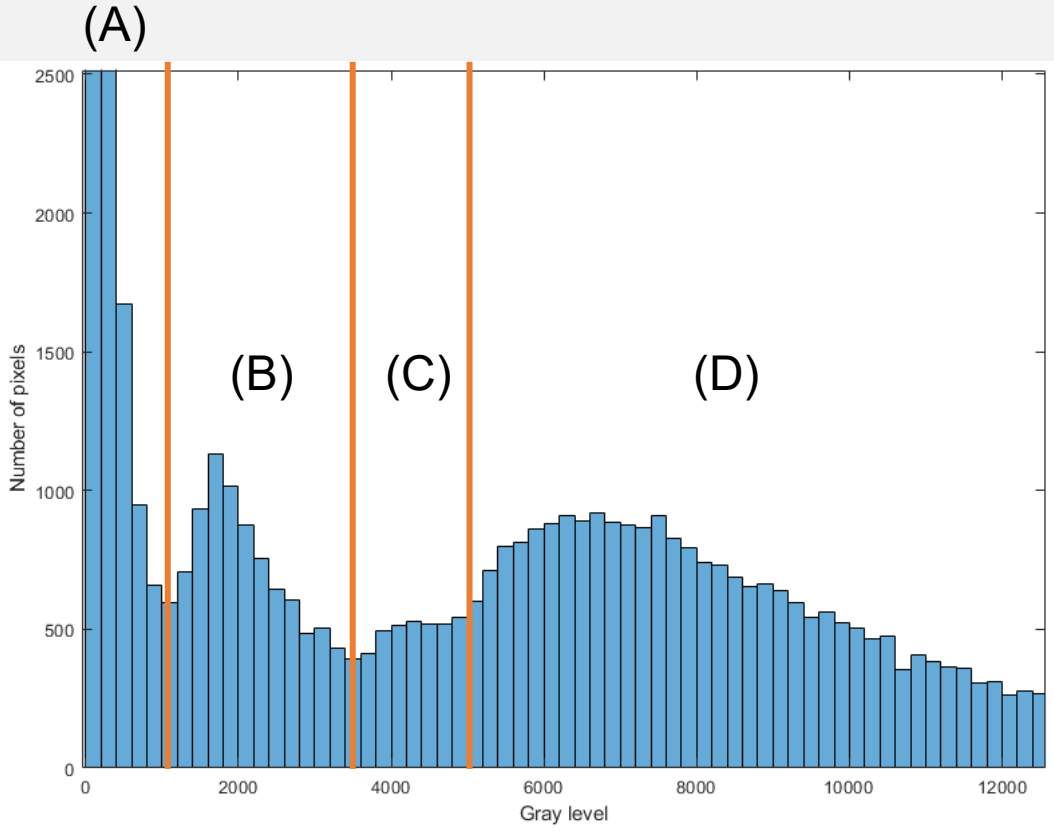
Which region of the histogram is the background?



Which region of the histogram is the background?

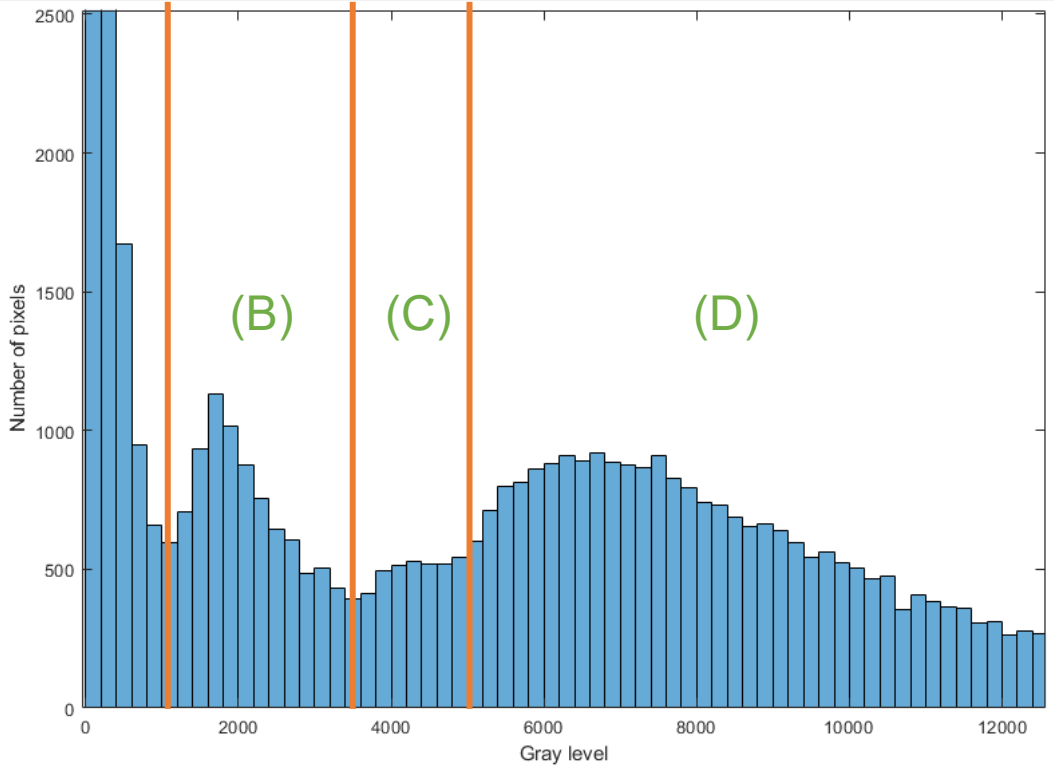


Which region(s) of the histogram are the cells?



Which region(s) of the histogram are the cells?

(A)



Otsu's method

- Otsu's method is a popular algorithm to automatically determine a threshold value

- The next few slides will walk you through the math, but you won't need to write the algorithm yourself

- Instead, think about what limitations the algorithm will have

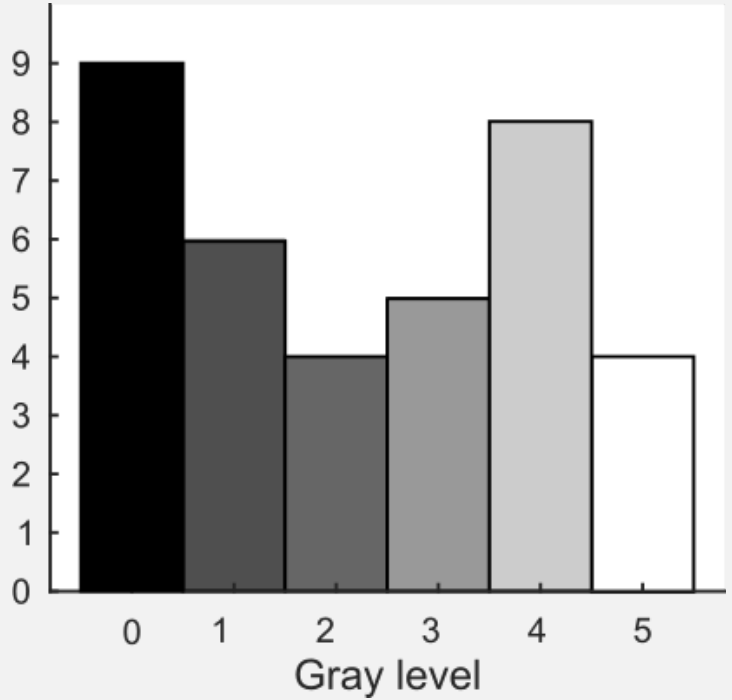
Otsu's method

- Otsu's method uses the image histogram to exhaustively search for a threshold that maximizes a metric called the between class variance

Example image and histogram

0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

Image

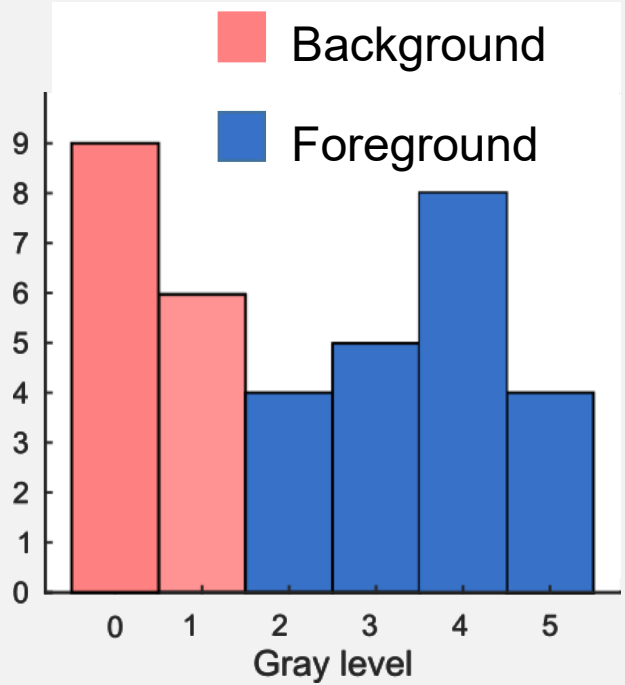


Intensity histogram

Otsu's algorithm attempts to divide the image into two classes: "background" and "foreground"

0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

Image



Intensity histogram

Between class variance σ_B^2

$$\sigma_B^2 = W_b(t)W_f(t)\big(\mu_b(t) - \mu_f(t)\big)^2$$

W_b and W_f are the weights given by

$$W_b(t) = \frac{N_b}{N_b + N_f}$$

Note: The subscript b stands for "background" and f stands for "foreground"

where N_b is the number of pixels in the "background" class and N_f is the number of pixels in the "foreground" class

Between class variance σ_B^2

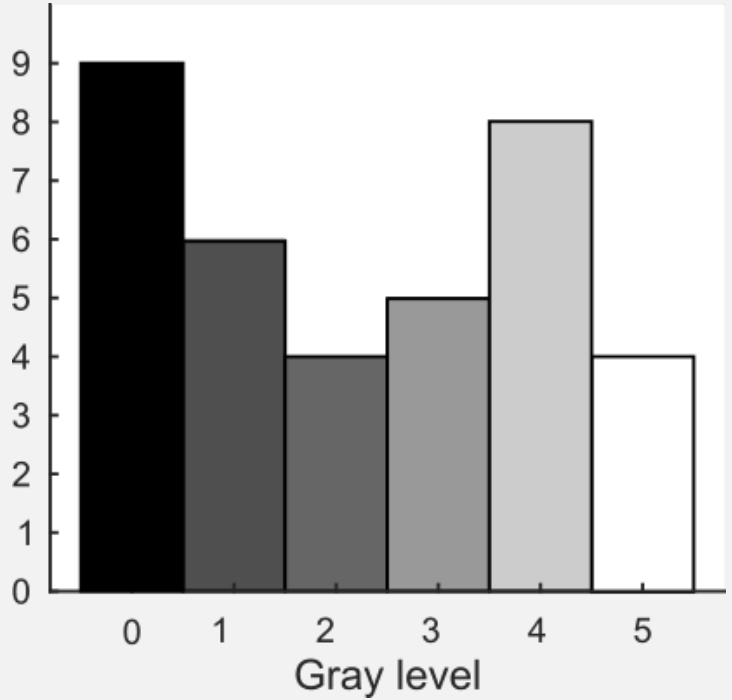
$$\sigma_B^2 = W_b(t)W_f(t)\big(\mu_b(t) - \mu_f(t)\big)^2$$

μ_b and μ_f are the average intensities of the background and foreground classes

Example of a single calculation

0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

Image

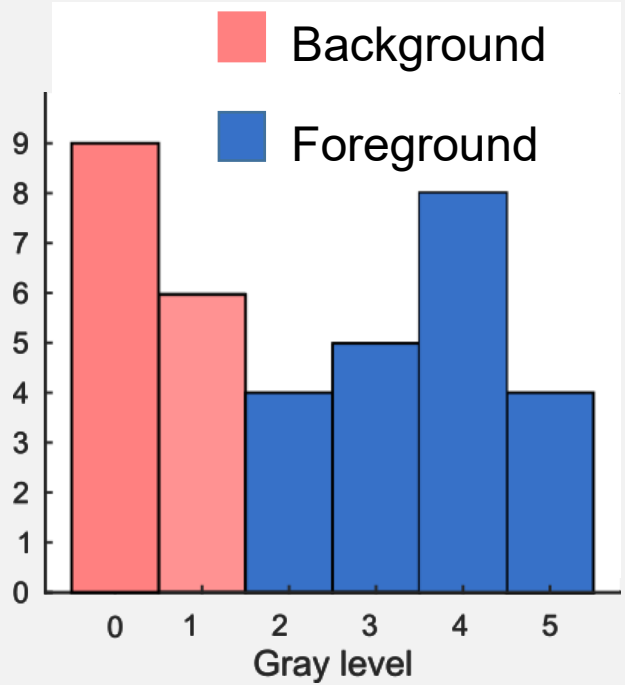


Intensity histogram

Algorithm picks a threshold value and divides the pixels

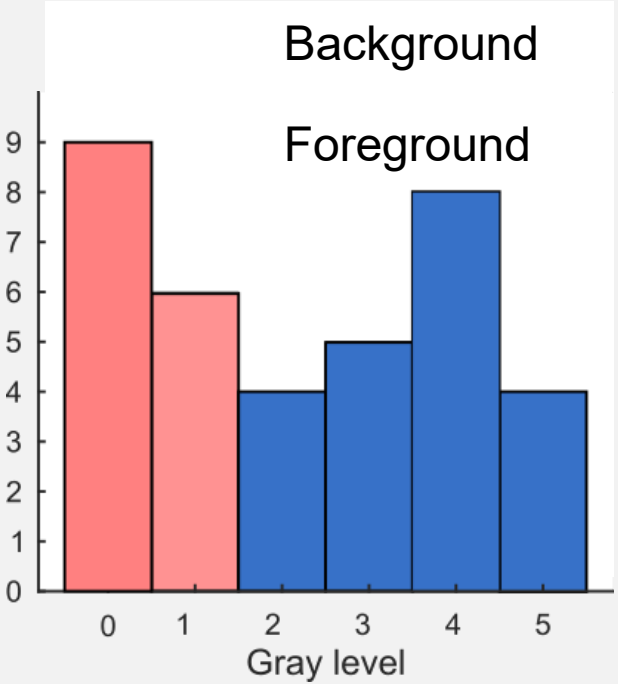
0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

Image



Intensity histogram

Compute weights

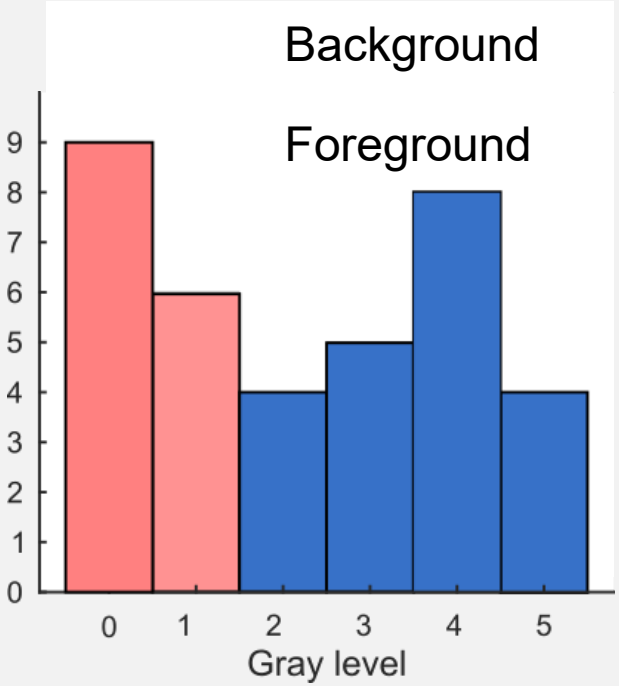


$$W_{\{b,f\}} = \frac{N_{b,f}}{N_b + N_f}$$

$$W_b = \frac{9+6}{36} = 0.42$$

$$W_f = \frac{4+5+8+4}{36} = 0.58$$

Compute mean intensities



$$\mu_{\{b,f\}} = \frac{\sum_i p_i x_i}{N_{b,f}}$$

$$\mu_b = \frac{(9 \times 0) + (6 \times 1)}{9 + 6} = 0.4$$

$$\mu_f = \frac{(4 \times 2) + (5 \times 3) + (8 \times 4) + (4 \times 4)}{4 + 5 + 8 + 4} = 3.57$$

Compute the between class variance σ_B^2

$$\sigma_B^2 = W_b W_f \left(\mu_b - \mu_f\right)^2$$

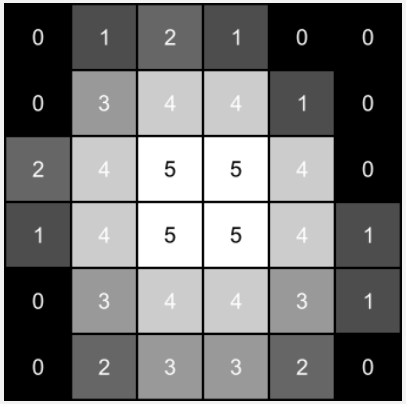
$$= (0.42 \times 0.58)(0.4 - 3.57)^2$$

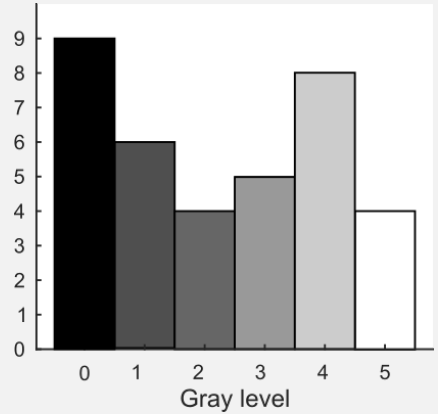
$$= 2.44$$

Otsu's method

- Otsu's method uses the image histogram to exhaustively search for a threshold that maximizes a metric called the between class variance

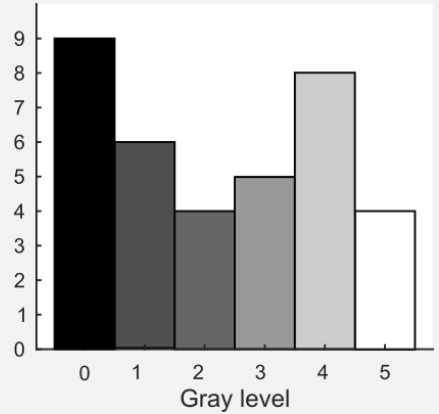
- **Exhaustive search:** The algorithm does this calculation for every possible value for the threshold





Grayscale value →	0	1	2	3	4	5
W_b	0	0.25	0.42	0.53	0.67	0.89
μ_b	0	0	0.40	0.74	1.21	1.91
W_f	1	0.75	0.58	0.47	0.33	0.11
μ_f	2.25	3.00	3.57	3.94	4.33	5.00
σ_b^2	0	1.69	2.44	2.56	2.17	0.95

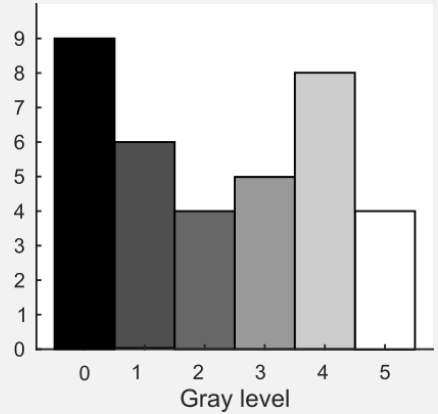
0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0



Find grayscale value that gives the highest value of σ_B

Grayscale value →	**0**	**1**	**2**	**3**	**4**	**5**
W_b	0	0.25	0.42	0.53	0.67	0.89
μ_b	0	0	0.40	0.74	1.21	1.91
W_f	1	0.75	0.58	0.47	0.33	0.11
μ_f	2.25	3.00	3.57	3.94	4.33	5.00
σ_b^2	0	1.69	2.44	2.56	2.17	0.95

0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

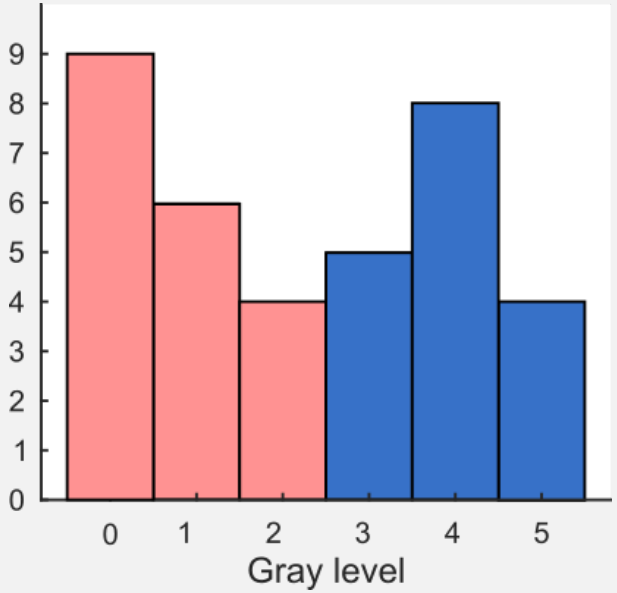


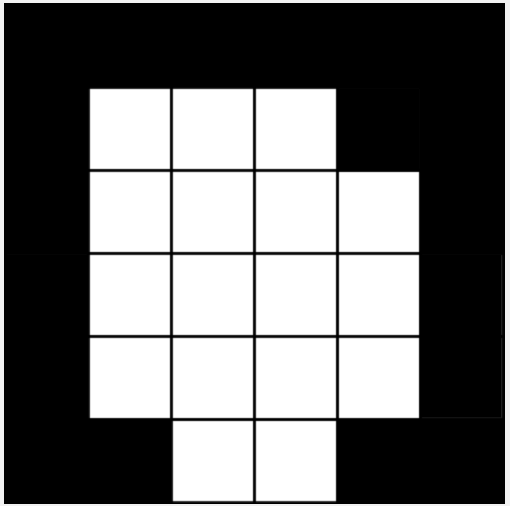
Find grayscale value that gives the highest value of σ_B

Grayscale value →	**0**	**1**	**2**	**3**	**4**	**5**
W_b	0	0.25	0.42	0.53	0.67	0.89
μ_b	0	0	0.40	0.74	1.21	1.91
W_f	1	0.75	0.58	0.47	0.33	0.11
μ_f	2.25	3.00	3.57	3.94	4.33	5.00
σ_b^2	0	1.69	2.44	2.56	2.17	0.95

Otsu's method returns a threshold intensity of 3

0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

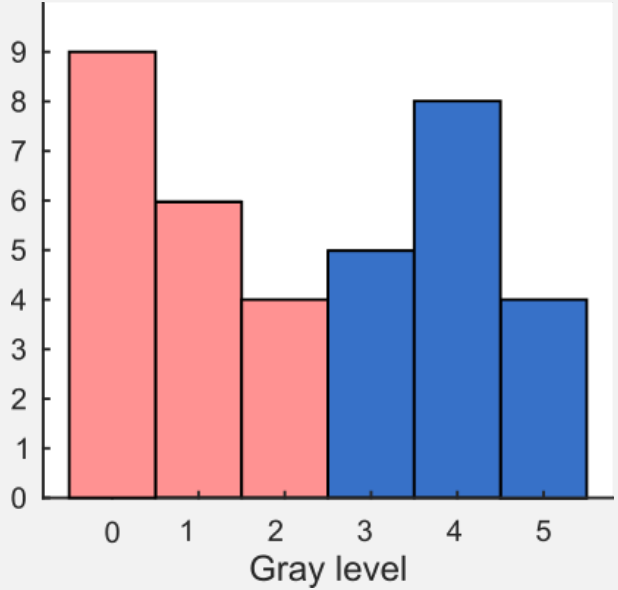




Resulting mask

What are the limitations of Otsu's method?

- Main assumption: there are **two distinct** intensity classes in the image



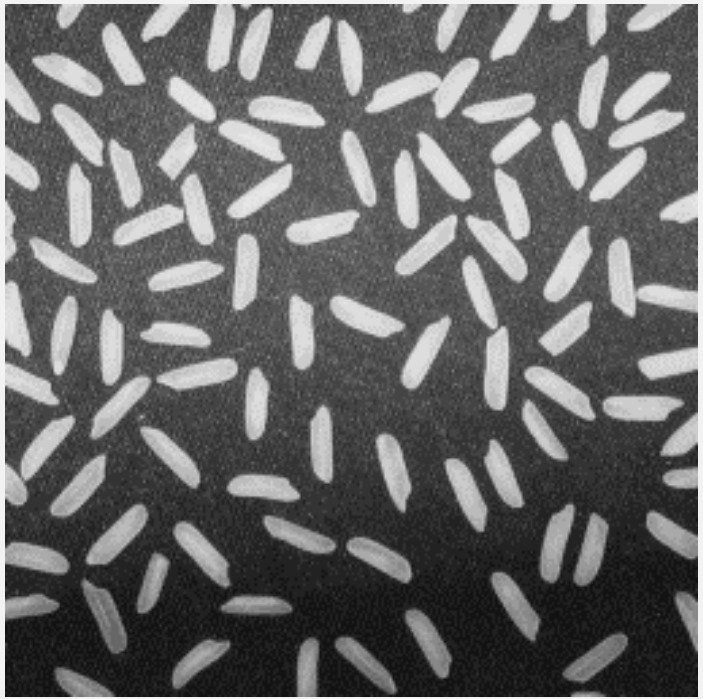
Only background and foreground

Distinct: Intensity distributions are **separated by a valley** in the intensity histogram

What are the limitations of Otsu's method?

- Global thresholding algorithm
- Global means that it uses the intensity of the entire image
- Does not work well if image has uneven illumination

Example of uneven illumination



Questions?

Using Otsu's method in MATLAB

Make a mask using Otsu's method using the function `imbinarize`

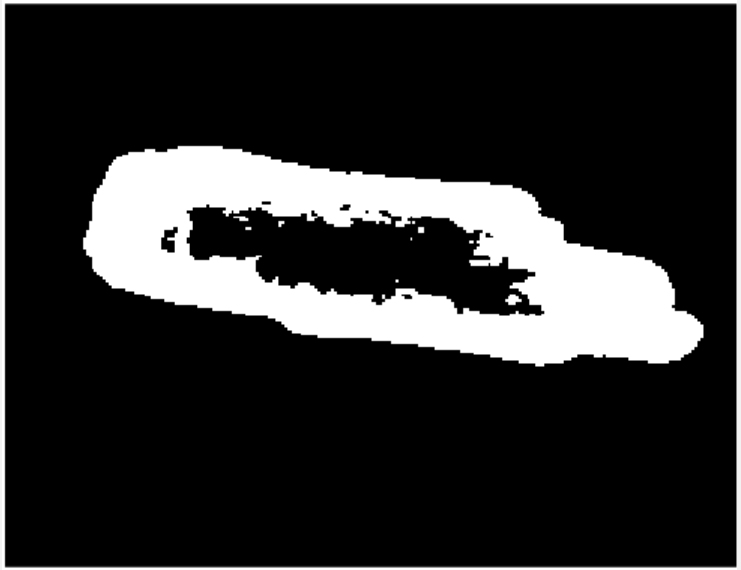
```
mask = imbinarize(I)
```

Can you explain the results using the concepts we've learned today?

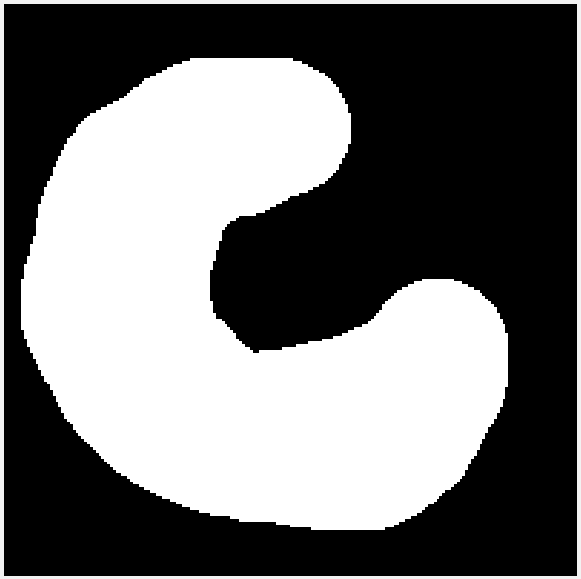
To fill in holes in a mask

```
mask = imfill(mask, 'holes')
```

`imfill` defines "holes" as a region of `false` pixels completely surrounded by `true` pixels



Hole



Not a hole

Questions?