Introduction to Image Processing Workshop

Image Package Built-in functions
Semester 2

Romuald Jolivot - romuald.j@bu.ac.th

Reminder

Don’t forget to load the “*image*” package

- Load the package
    - “pkg load image”

Using existing functions

- Know already use of double loop to check/process single pixels
- Know how to create your own function
- Still need to learn image processing concept
- But:
  - Process images faster using built-in function
  - Use optimized algorithm

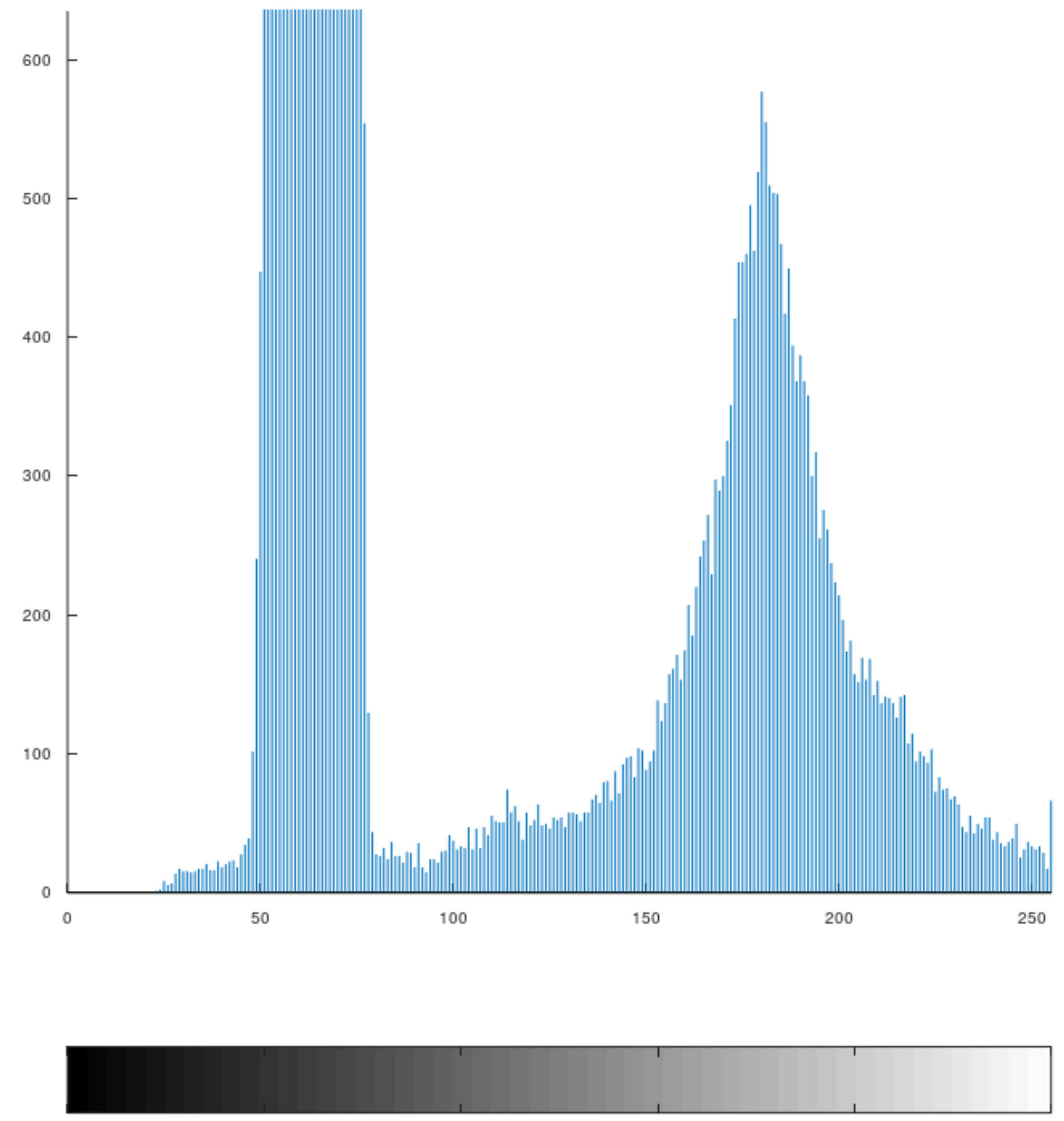
List of functions

Function name	Description
imhist	Output the histogram of the image
im2bw	Convert image to binary
histeq	Equalize histogram
imfill	Fill holes or regions in image
imfindcircles	Find circles in image
viscircles	Draw circles
regionprops	Properties of connected components
bwboundaries	Output boundary around detected group of pixel
imfilter	Apply a filter to an image
imclearborder	Clear pixel along the border

imhist

- Output the histogram of an image
- function:
  - h = imhist(I) with I the loaded image, save the histogram into a variable
  - imhist(I) plot directly the histogram

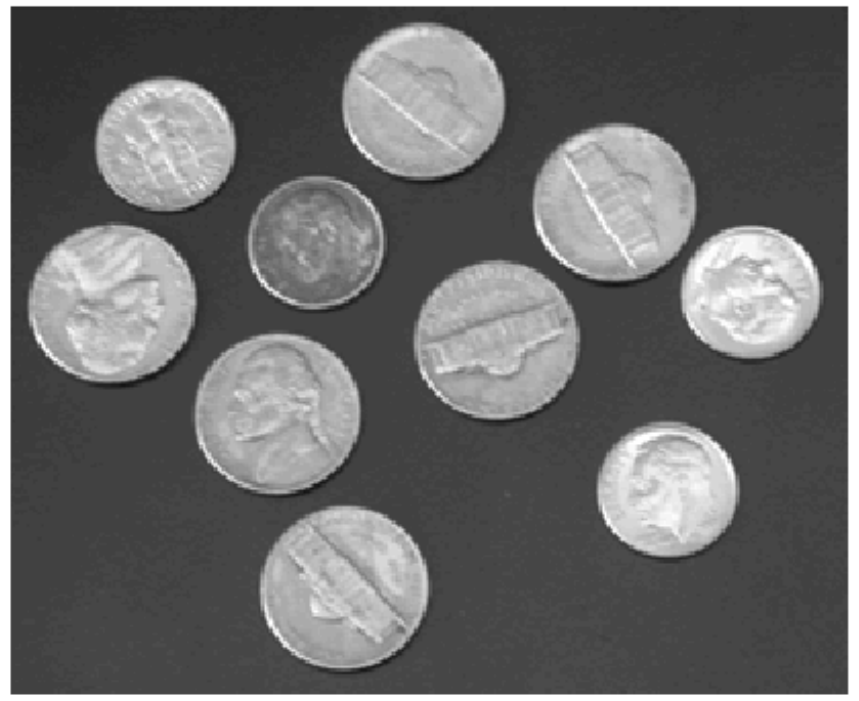
Original image

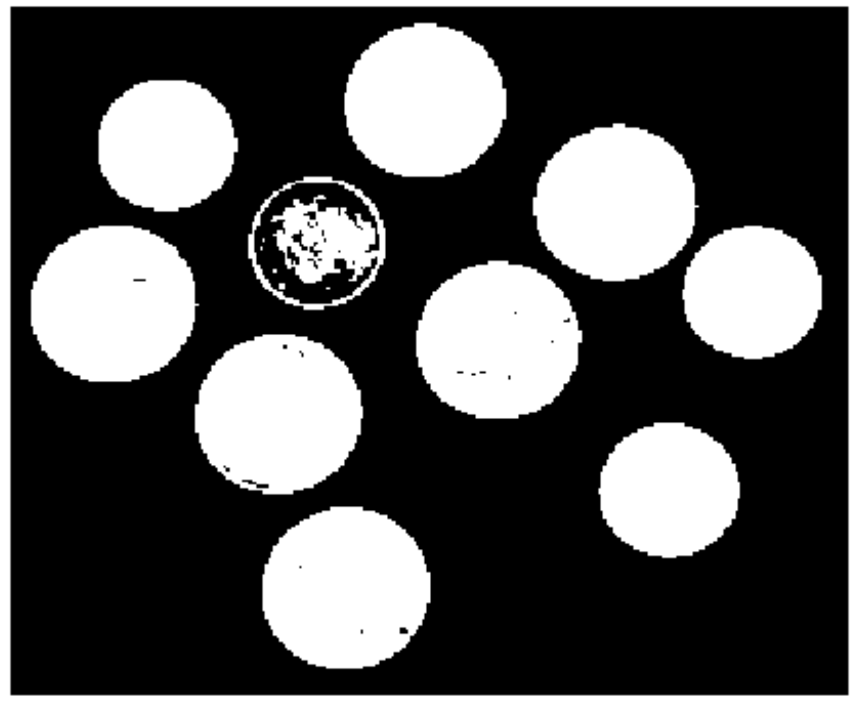


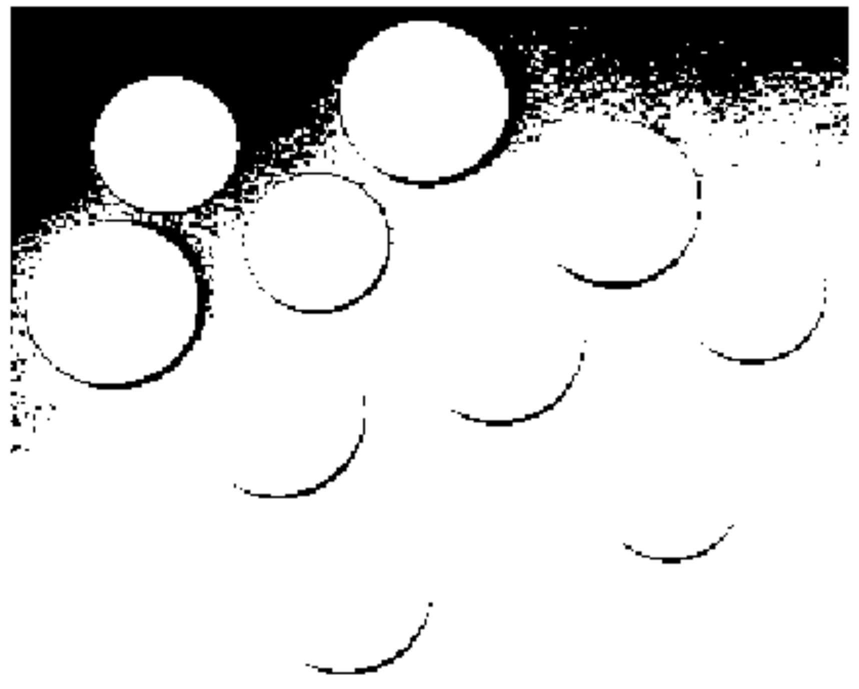
Histogram of original image

im2bw

- Convert color and grayscale image to binary
- function:
  - BW = im2bw(I)   with I the loaded image
  - BW = im2bw(I, thresh)   with thresh a defined threshold value

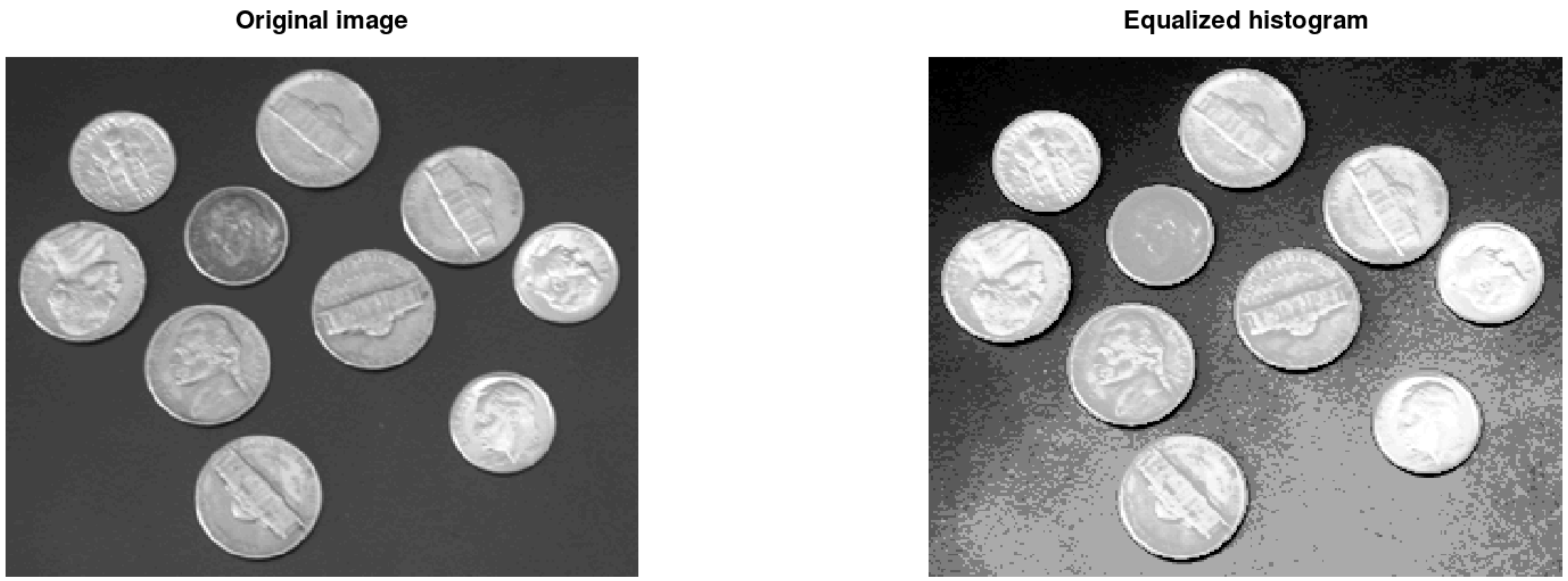

Original image


Basic binarization


Manual threshold

histeq

- Enhance contrast by histogram equalization
- function:
  - J = histeq(I)   with I the loaded image



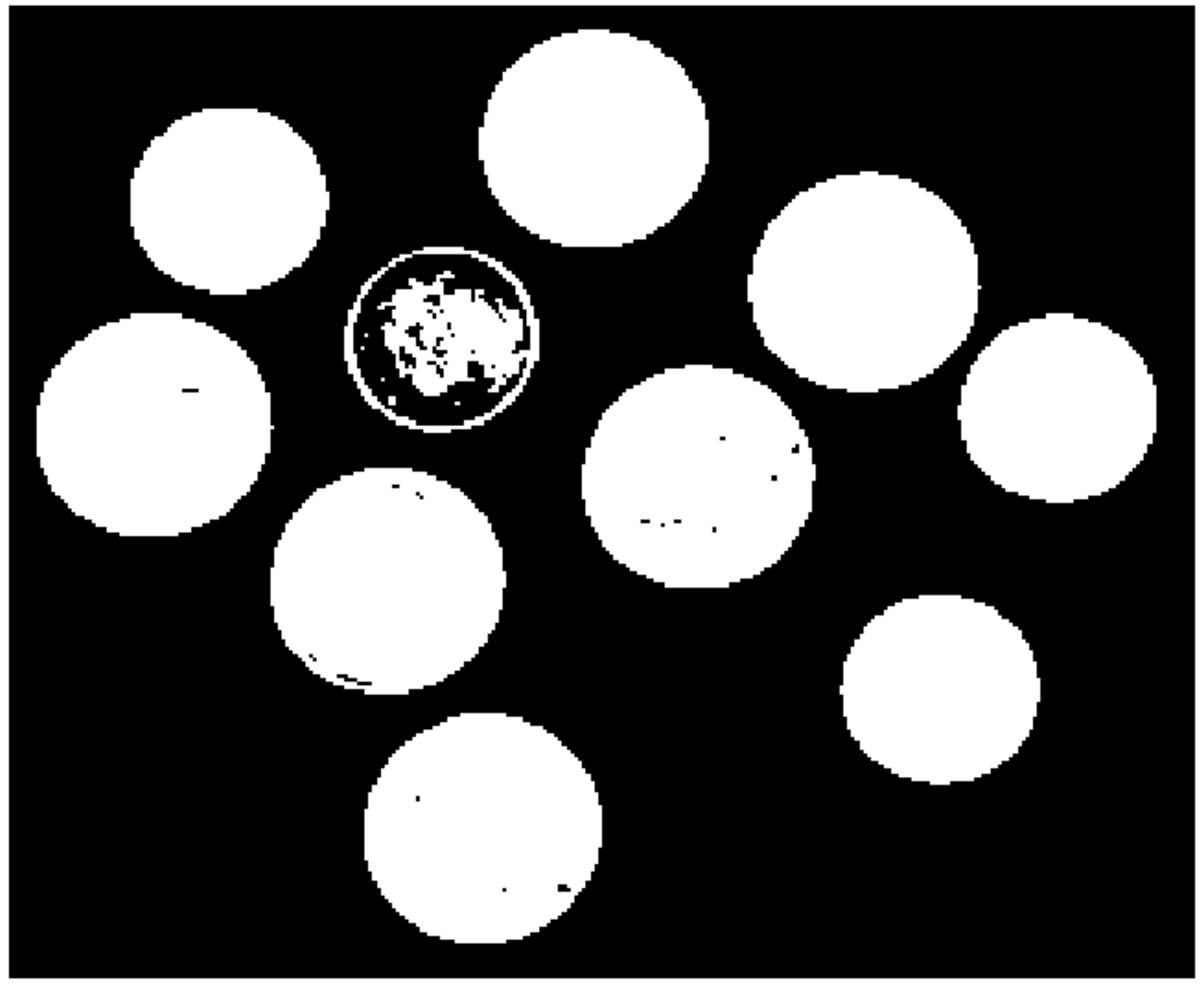
Original image

Equalized histogram

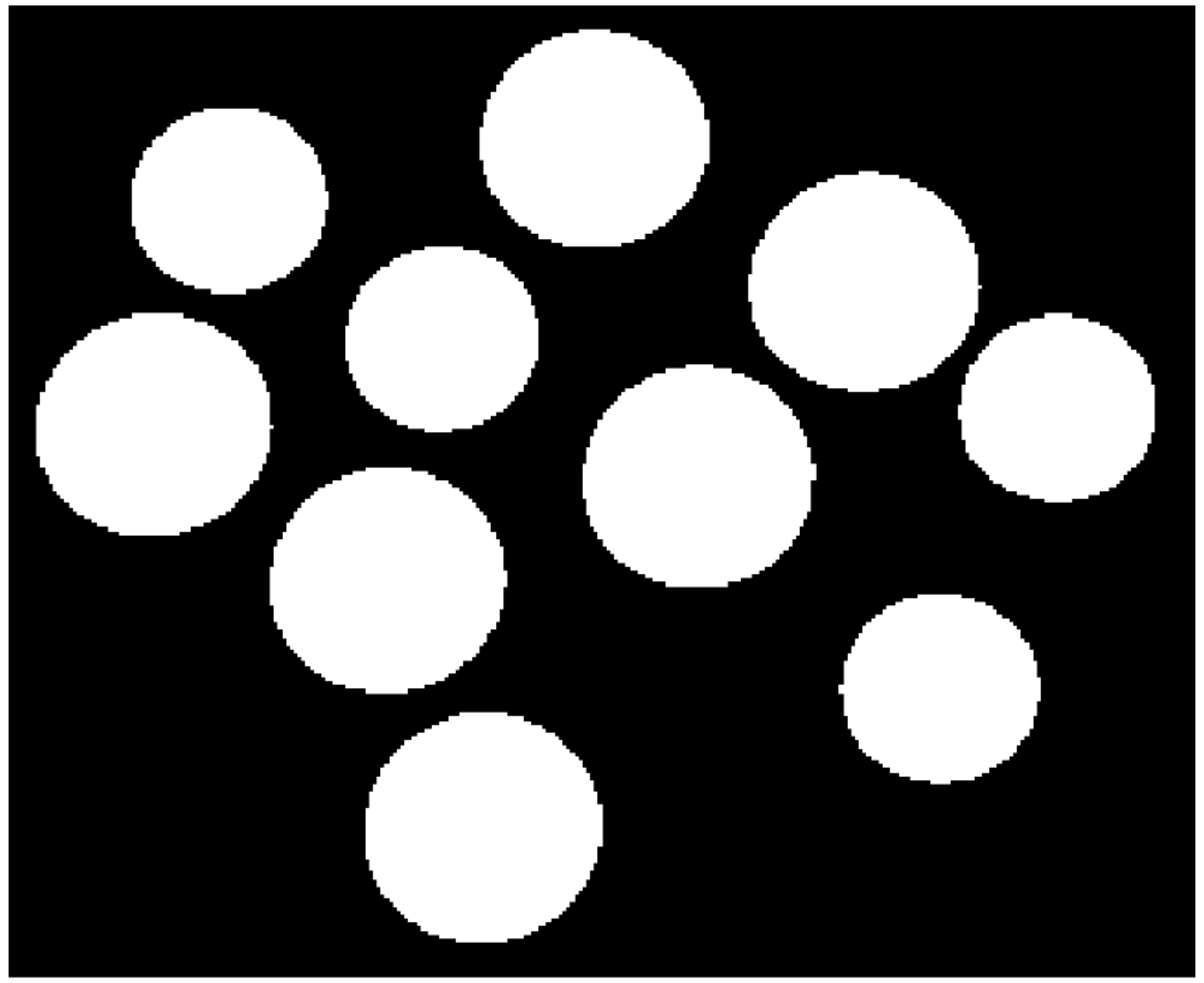
imfill

- Fill group of pixels with holes in it
- function:
  - J = imfill(BW)   with BW the binarized image

Basic binarization



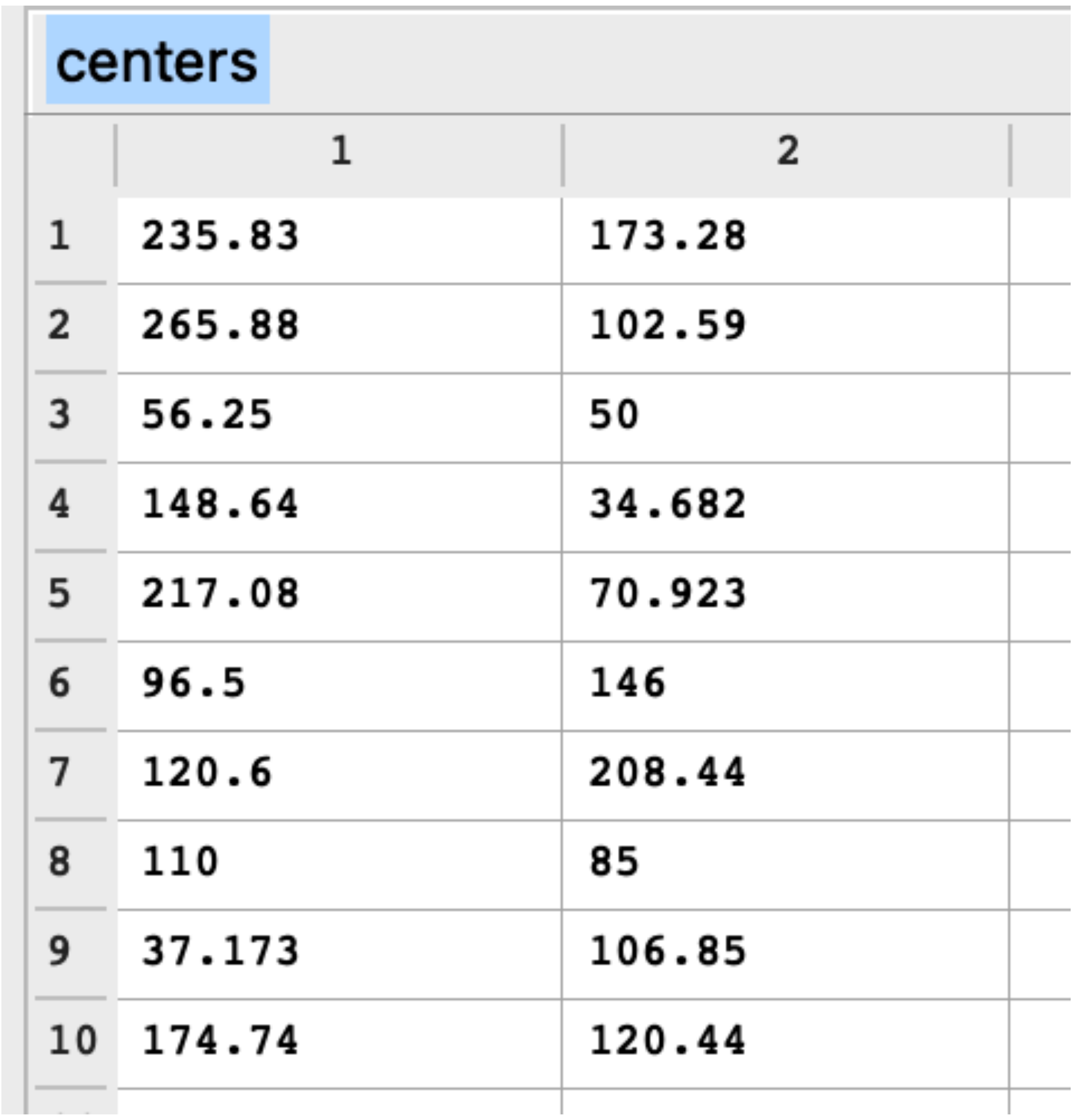
Hole filled



imfindcircles

- Find circles in an image
- function:
  - [centers,radii] = imfindcircles(I, [Range radius])   with I the originial image

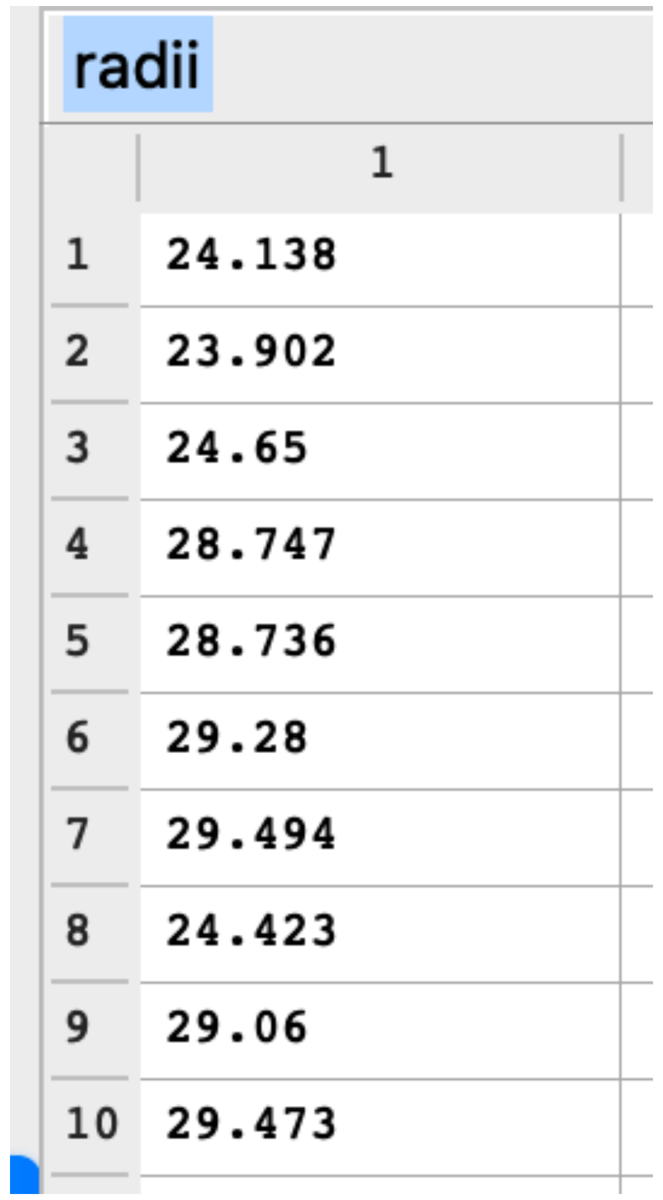
- centers: x and y coordinates of the centers



centers

	1	2
1	235.83	173.28
2	265.88	102.59
3	56.25	50
4	148.64	34.682
5	217.08	70.923
6	96.5	146
7	120.6	208.44
8	110	85
9	37.173	106.85
10	174.74	120.44

- radii: radius of the circles

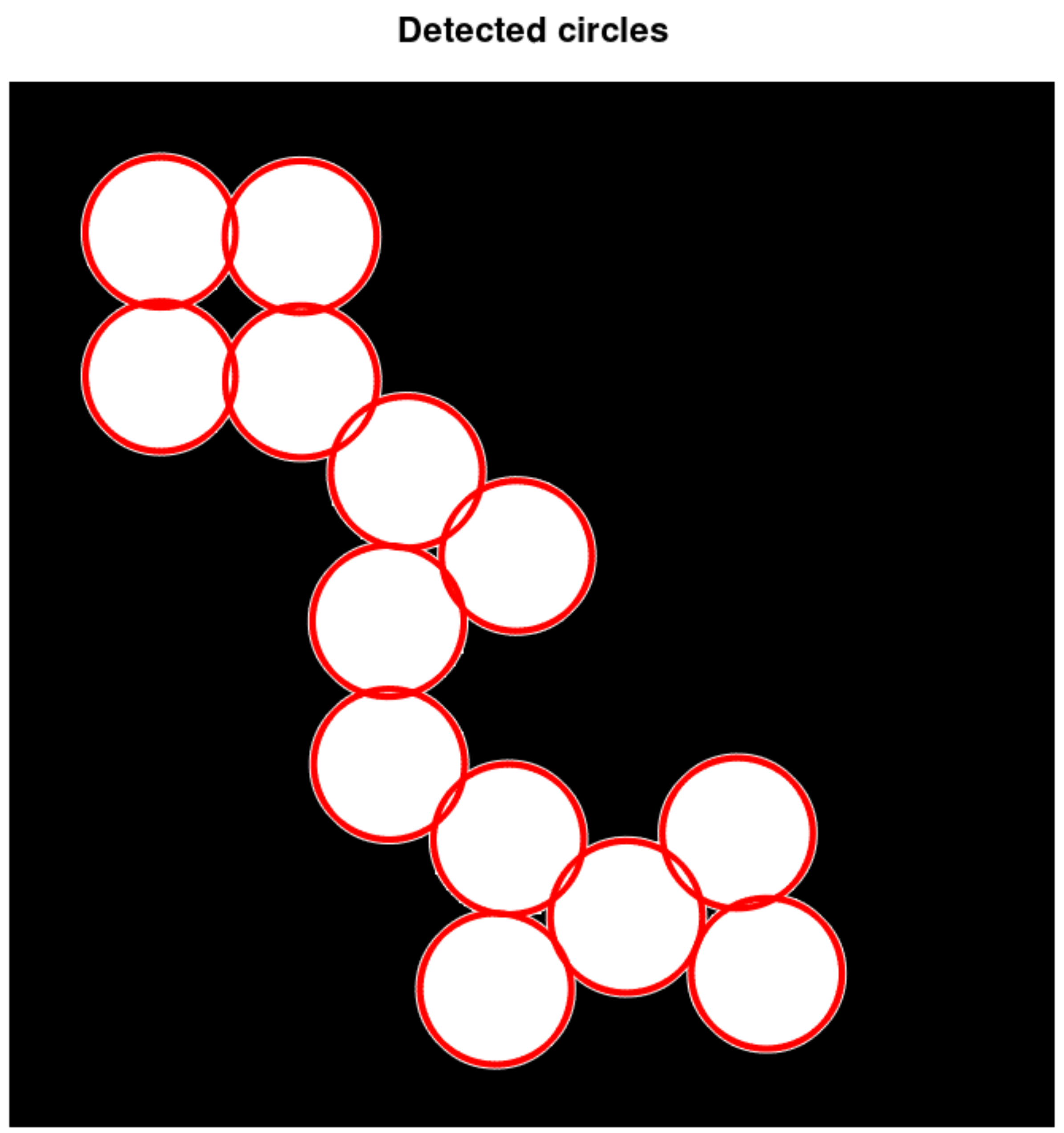


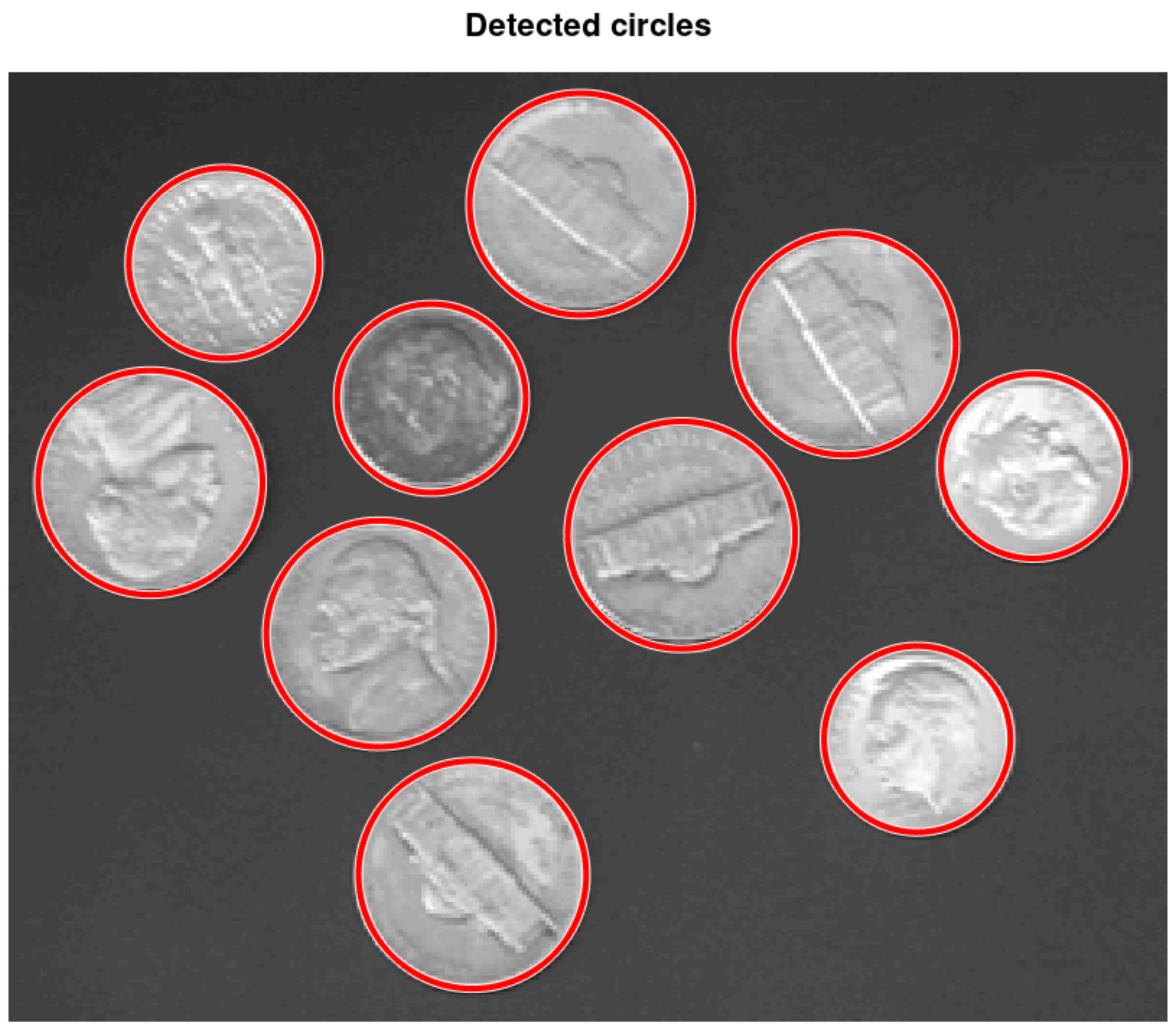
radii

	1
1	24.138
2	23.902
3	24.65
4	28.747
5	28.736
6	29.28
7	29.494
8	24.423
9	29.06
10	29.473

viscircles

- Draw circles with specified centers and radius
- Function:
  - viscircles(centers, radii)   with centers and radii obtained from imfindcircles





regionprops

- Measure properties of connected pixel in an image
- Function:
  - cc = regionprops(BW) with BW, a binary image
    - without specification, return “area”, “centroid” and “boundingbox”
  - cc=regionprops(BW, ‘all’)
    - Returns shape and pixel measurement for all group of pixels:
  - Access elements using the following structure (area example)
    - cc.Centroid
    - centroid = cat(1,cc.Centroid)
    - Region properties are capital sensitive
    - cc.centroid won’t work

regionprops

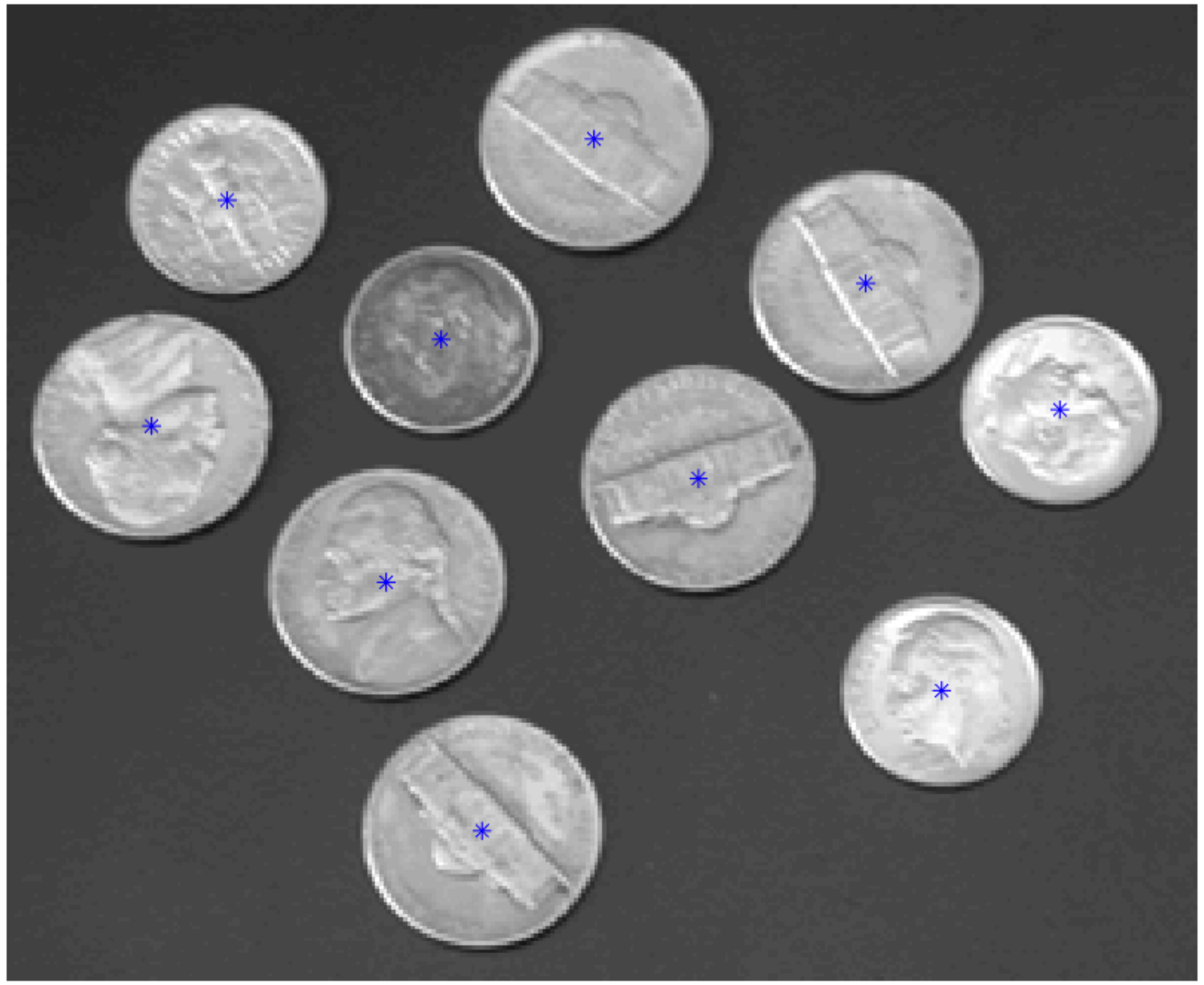
- Measure properties of connected pixel in an image
- Properties:
  - Check help for detail of each properties (*help regionprops*)

Area	BoundingBox	Centroid	ConvexArea	ConvexHull
ConvexImage	Eccentricity	EquivDiameter	EulerNumber	Extent
Extrema	FilledArea	FilledImage	Image	MajorAxisLength
MaxIntensity	MeanIntensity	MinorAxisLength	Orientation	Perimeter
PixelIdXList	PixelList	PixelValues	Solidity	SubarrayIdx
WeigthCentroid				

regionprops

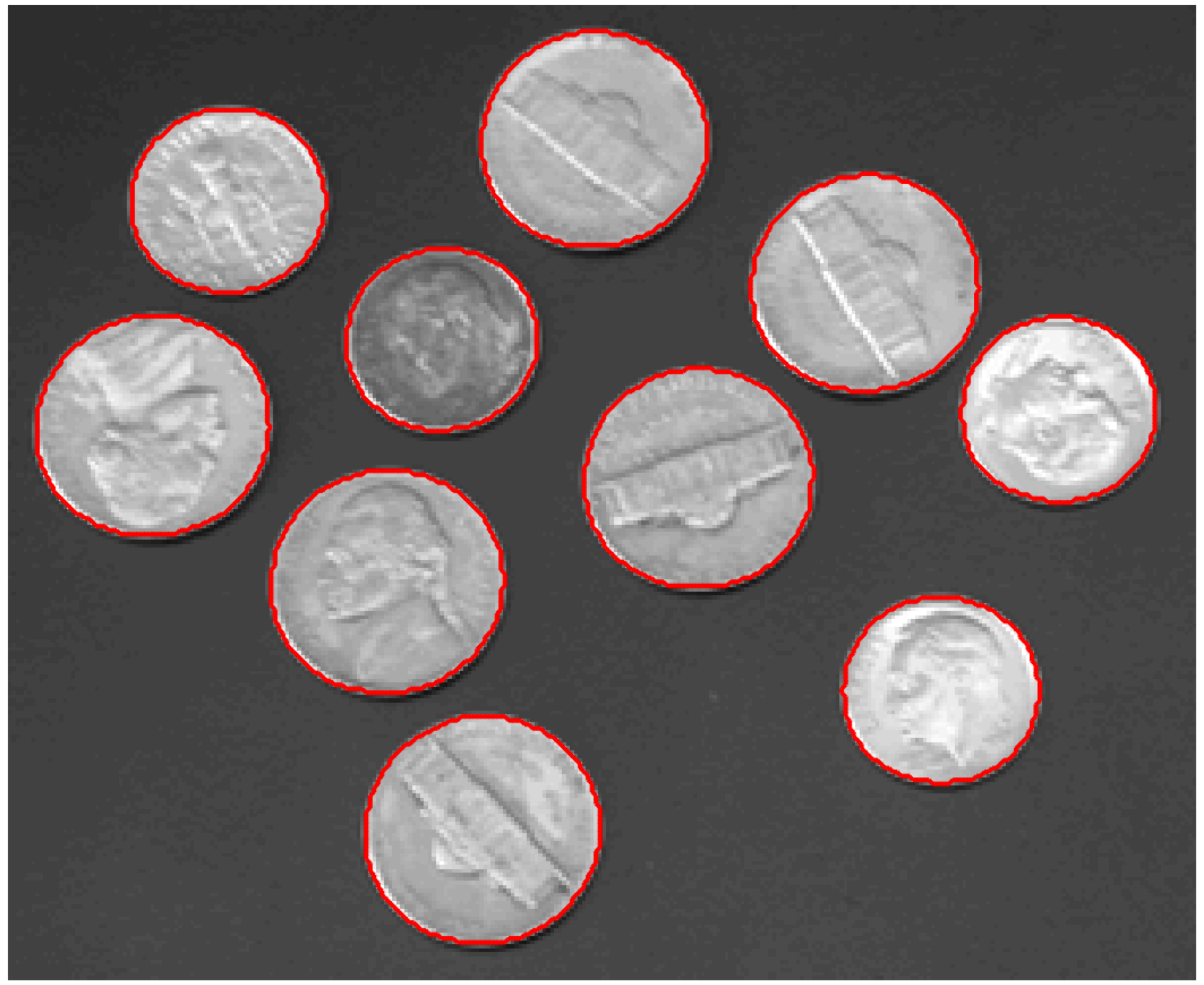
- Example of centroids displayed

Centroids displayed



bwboundaries

- Trace boundaries around group of pixels
- function:
  - J = bwboundaries(BW) with BW the binarized image



Boundaries displayed

fspecial

- Create a predefined filter
- function:
  - h = fspecial(‘type’, size)

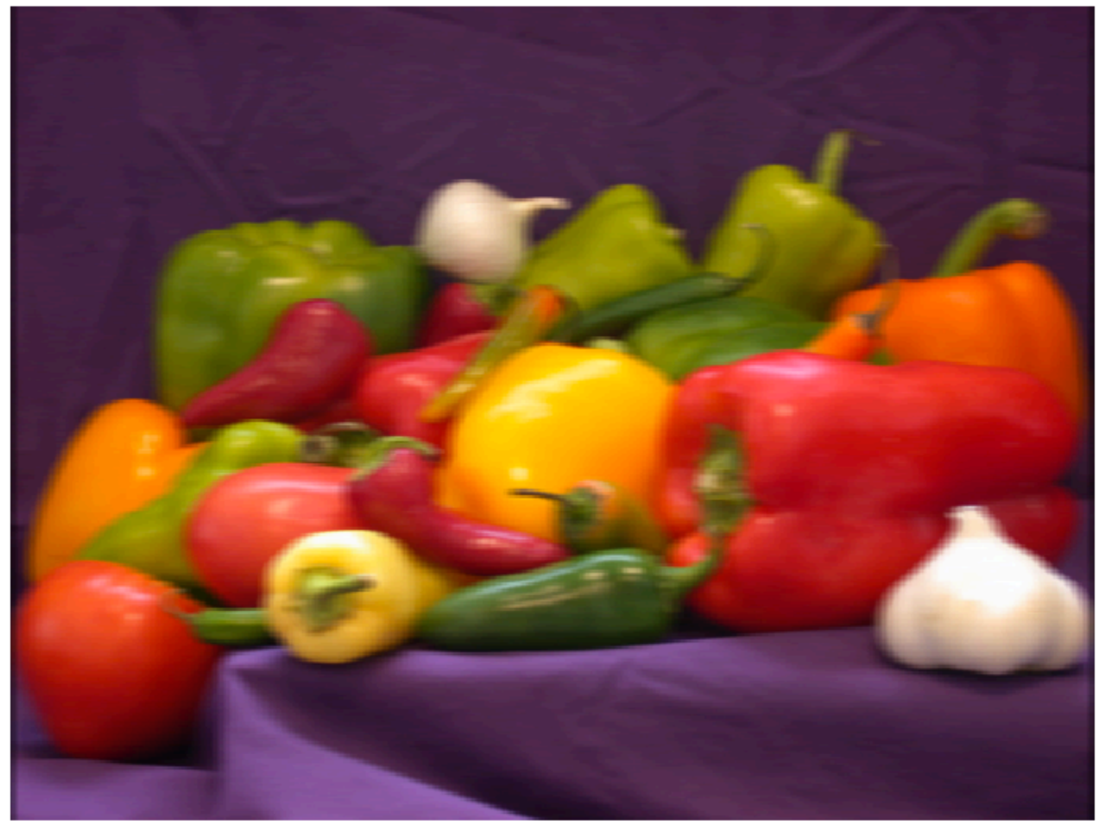
(“average”) or (“average”, lengths (N or NxM))	(“disk”) or (“disk”, radius)
(“guassian”) or (“gaussian”, lengths)	(“log”) or (“log”, length)
(“laplacian”) or (“laplacian”, length)	(“unsharp”) or (“unsharp”, alpha)
(“motion”) or (“motion”, length)	(“sobel”)
(“prewire”)	(“kirsch”)

imfilter

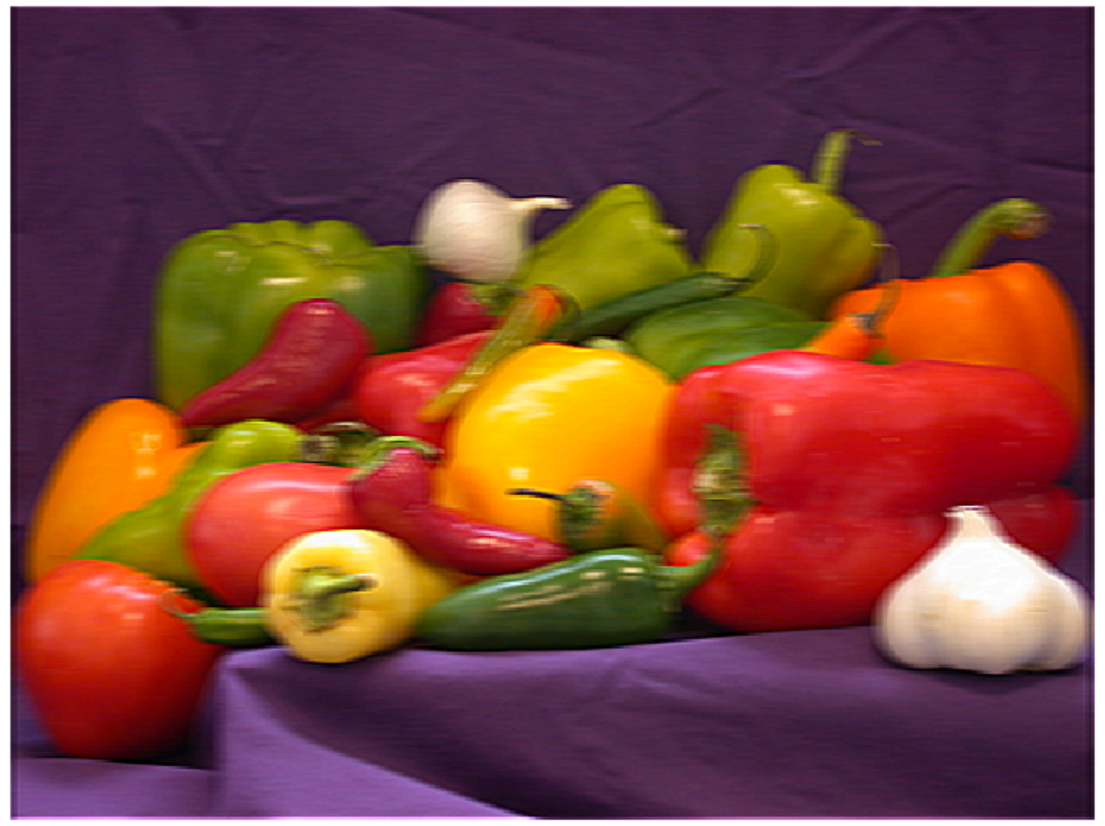
- Apply a filter to an image
- function:
  - J = imfilter(I, h)   I, the input image to be filter, h, the filter to be applied

Original image

Motioned image

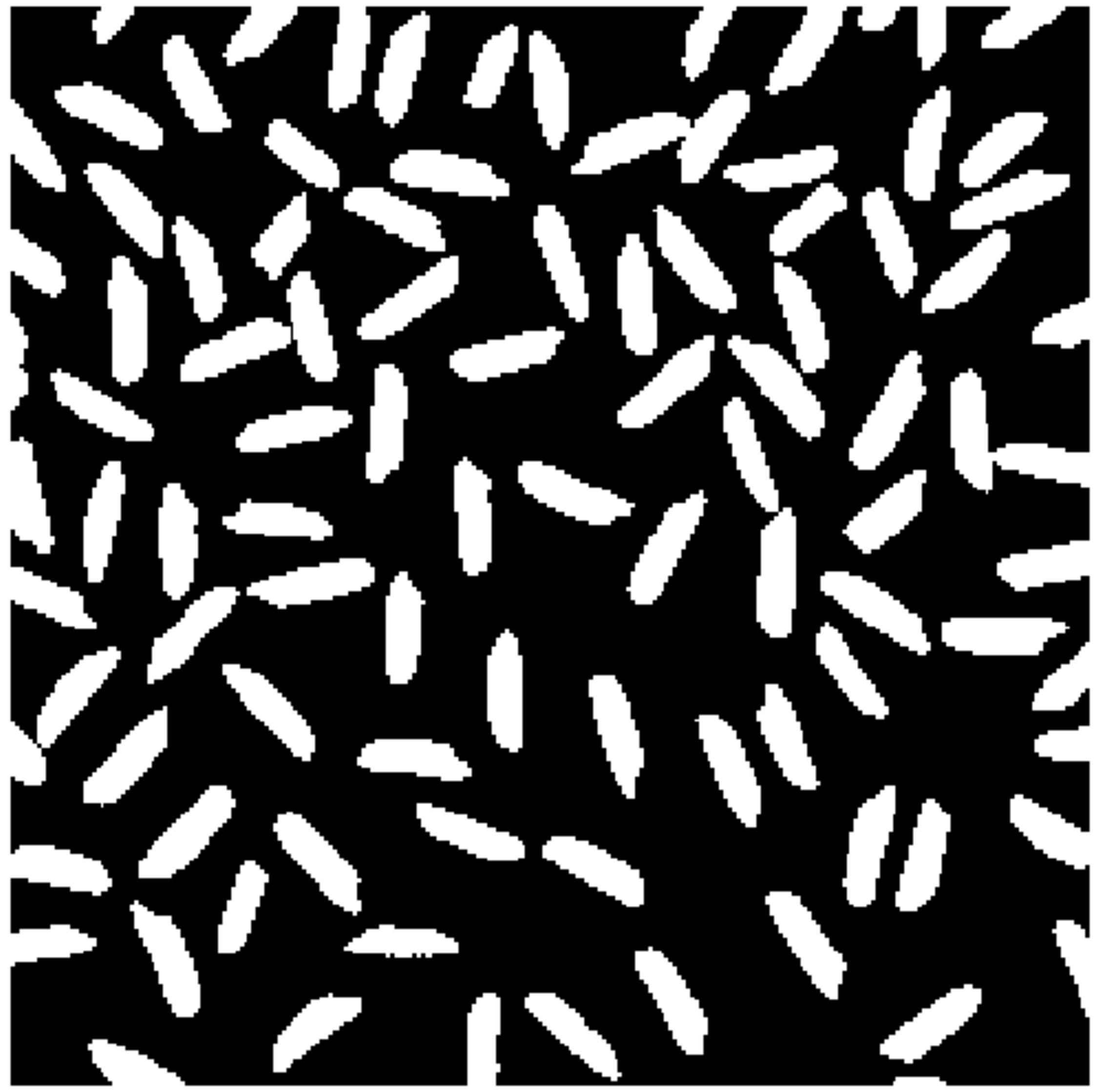


Unsharped image

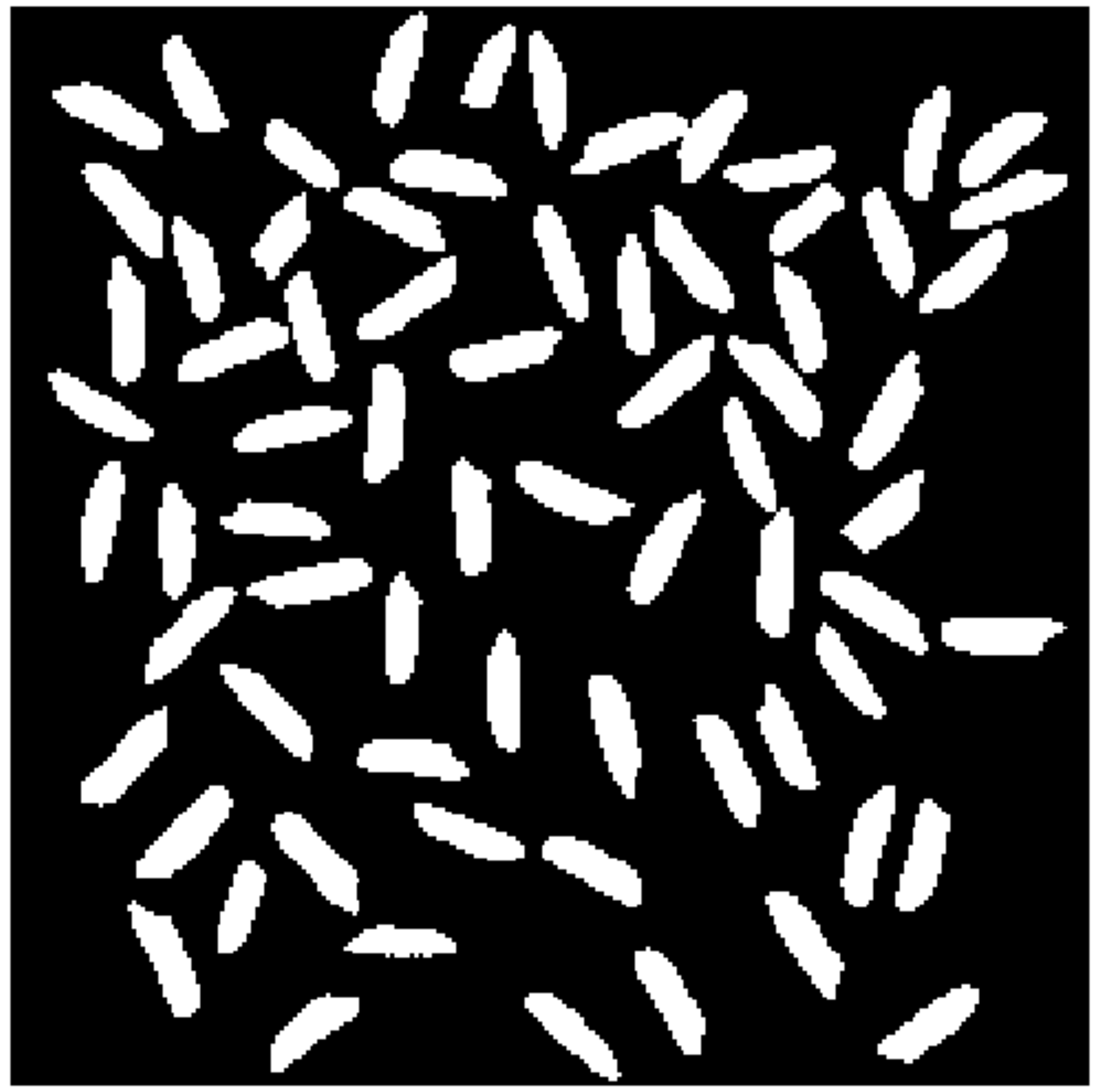


imclearborder

- Remove pixel group around the border of the image
- function:
  - J = imclearborder(BW)   with BW the binarized image



Original image



Bordered pixels removed