Summer Training I
MATLAB for Engineers

Benha University

Computer Systems Engineering
Electrical Engineering Department

Faculty of Engineering
(at Shoubra)

Lab 05

Getting Started – Example 1

1. Start MATLAB
2. Read an image into the workspace, using the imread command.
```
≫I = imread('cameraman.tif');
```
3. Display the image, using the `imshow` function.
```
≫imshow(I)
```
4. You can also view an image in the Image Viewer app.
```
≫ imtool(I)
```
5. View the distribution of image pixel intensities.
```
≫figure
   imhist(I)
```
6. Improve the contrast in an image, using the `histeq` function. Histogram equalization spreads the intensity values over the full range of the image.
```
≫I2 = histeq(I);
Figure
imshow(I2)
```
7. Write the newly adjusted image I2 to a disk file, using the imwrite function.
```
≫imwrite (I2, ' cameraman2.png');
```
8. you can view an image info using `imfinfo` function which returns the information about the image in the file, such as its format, size, width, and height.
```
≫imfinfo(' cameraman2.png')
```

Example2 – find the circles

1. Read an image into the workspace, using the imread command.
```
≫rgb = imread('coloredChips.png');
≫imshow(rgb)
```
2. convert the image to greyscale, look at the grayscale version of this image to view whether the objects are brighter or darker than the background.
```
≫gray_image = rgb2gray(rgb);
≫imshow(gray_image)
```
3. The background is quite bright and most of the chips are darker than the background. But, by default, `imfindcircles` finds circular objects that are brighter than the background. So, set the parameter 'ObjectPolarity' to 'dark' in `imfindcircles` to search for dark circles.
```
≫[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark')
```
Note that the outputs `centers` and `radii` are empty, which means that no circles were found. This happens frequently because `imfindcircles` is a circle *detector*, and similar to most detectors, `imfindcircles` has an internal *detection threshold* that determines its sensitivity. In simple terms it means that the detector's confidence in a certain (circle) detection has to be greater than a certain

level before it is considered a *valid* detection. `imfindcircles` has a parameter 'Sensitivity' which can be used to control this internal threshold, and consequently, the sensitivity of the algorithm. A higher 'Sensitivity' value sets the detection threshold lower and leads to detecting more circles. This is similar to the sensitivity control on the motion detectors used in home security systems.

4. Increase detection sensitivity → it is possible that at the default sensitivity level all the circles are lower than the internal threshold, which is why no circles were detected. By default, 'Sensitivity', which is a number between 0 and 1, is set to 0.85. Increase 'Sensitivity' to 0.9.

```
≫[centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...
         'Sensitivity',0.9)
≫imshow(rgb)
  h = viscircles(centers,radii);
```

5. increase the sensitivity to get more circles

```
≫  h = viscircles(centers,radii);
   [centers,radii] = imfindcircles(rgb,[20 25],'ObjectPolarity','dark', ...
     'Sensitivity',0.97);
   imshow(rgb)
   h = viscircles(centers,radii);
```

6. The yellow chips are almost the same intensity, maybe even brighter, as compared to the background. Therefore, to detect the yellow chips, change 'ObjectPolarity' to 'bright'.

```
≫  [centersBright,radiiBright] = imfindcircles(rgb,[20 25], ...
    'ObjectPolarity','bright','Sensitivity',0.92);
≫  imshow(rgb)
≫  hBright = viscircles(centersBright, radiiBright);
```

7. if you increase the sensitivity it will find other circles than the yellow ones so we will edit the edgethreshold value. The 'EdgeThreshold' parameter controls how *high* the gradient value at a pixel has to be before it is considered an edge pixel and included in computation. A high value (closer to 1) for this parameter will allow only the strong edges (higher gradient values) to be included, whereas a low value (closer to 0) is more permissive and includes even the weaker edges (lower gradient values) in computation. In case of the missing yellow chip, since the contrast is low, some of the boundary pixels (on the circumference of the chip) are expected to have low gradient values. Therefore, lower the 'EdgeThreshold' parameter to ensure that the most of the edge pixels for the yellow chip are included in computation.

```
≫[centersBright,radiiBright,metricBright] = imfindcircles(rgb,[20 25], ...
    'ObjectPolarity','bright','Sensitivity',0.92,'EdgeThreshold',0.1);
≫hBright = viscircles(centersBright, radiiBright);
```

8. Draw both dark and bright circles

```
≫h = viscircles(centers,radii);
```