CSE 573: Computer Vision and Image Processing
Homework 2_ Scale-space blob detection

Name: Charushi Nanwani
UBIT ID: charushi
Person #: 50248736

1.

- Output of all images with blob detection

The outputs(blobs) for all the images are obtained considering the below values

Table 1

Sigma	2
N	15
K	2^0.25
Threshold	0.005

➢ **Image 1:** *'../data/butterfly.jpg'*

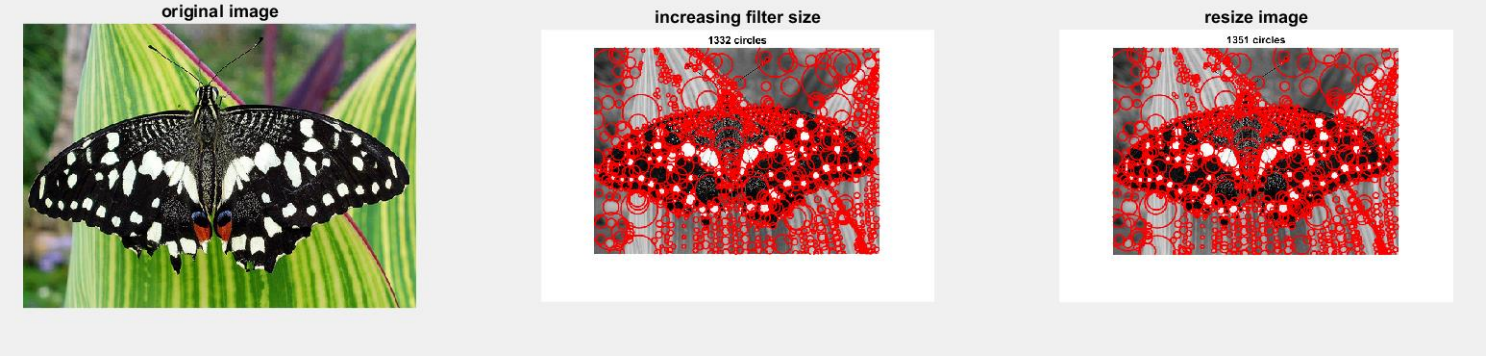

Figure 1.1

➢ **Image2:** *'../data/einstein.jpg'*

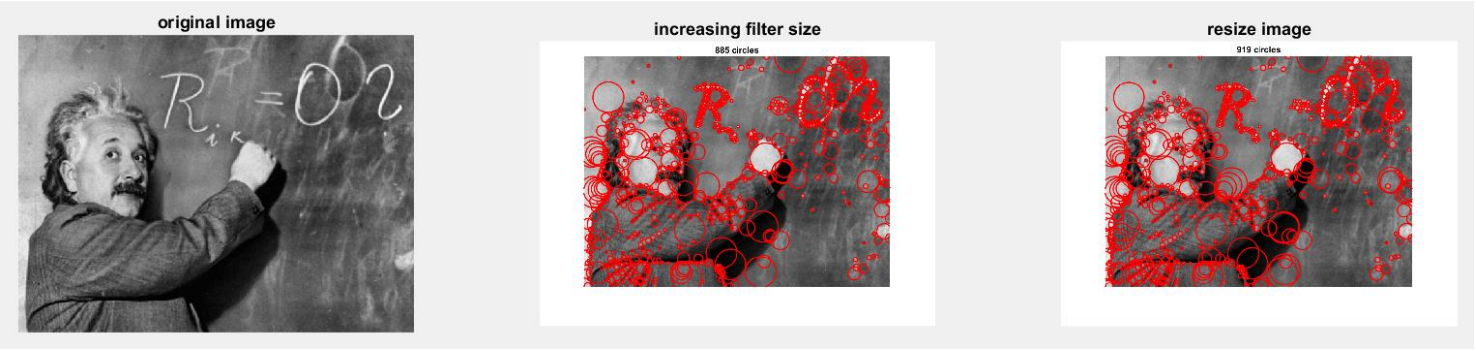

Figure 1.2

- **Image3**: *'../data/fishes.jpg'*

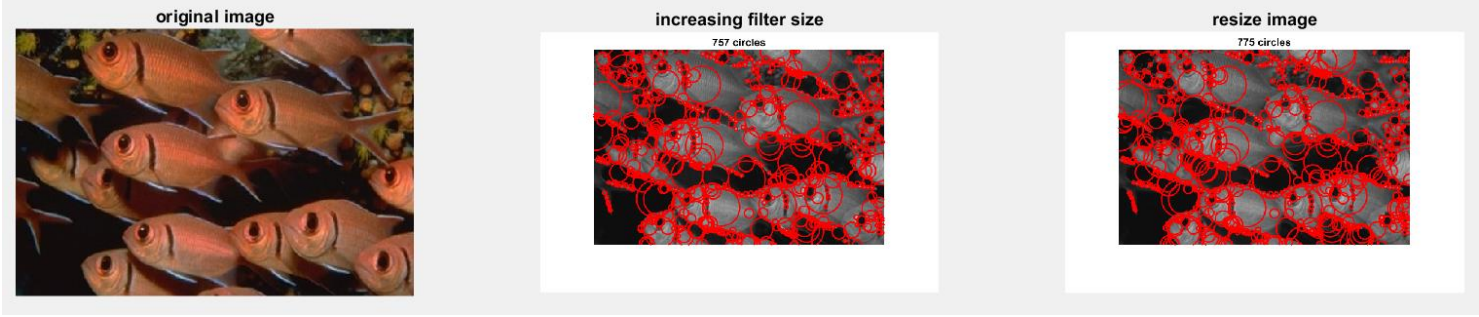

Figure 1.3

- **Image4:** *'../data/sunflowers.jpg'*

Figure 1.4

- **Image5:** *'../data/img1.jpg'*

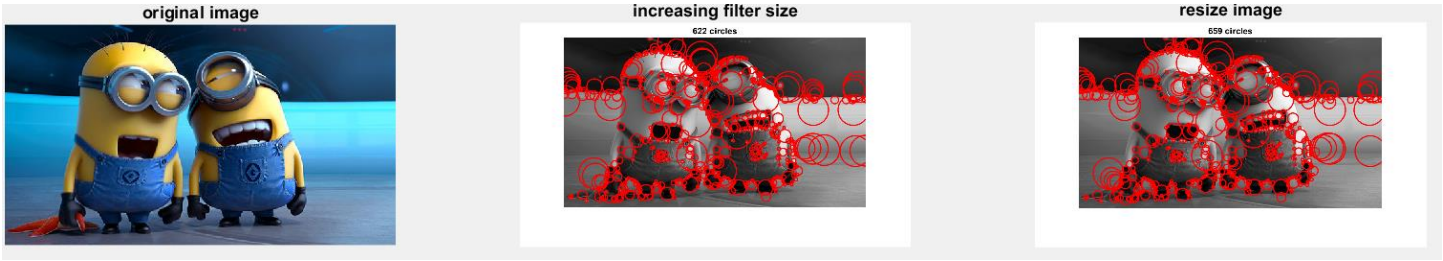

Figure 1.5

- **Image6:** *'../data/bob.jpg'*

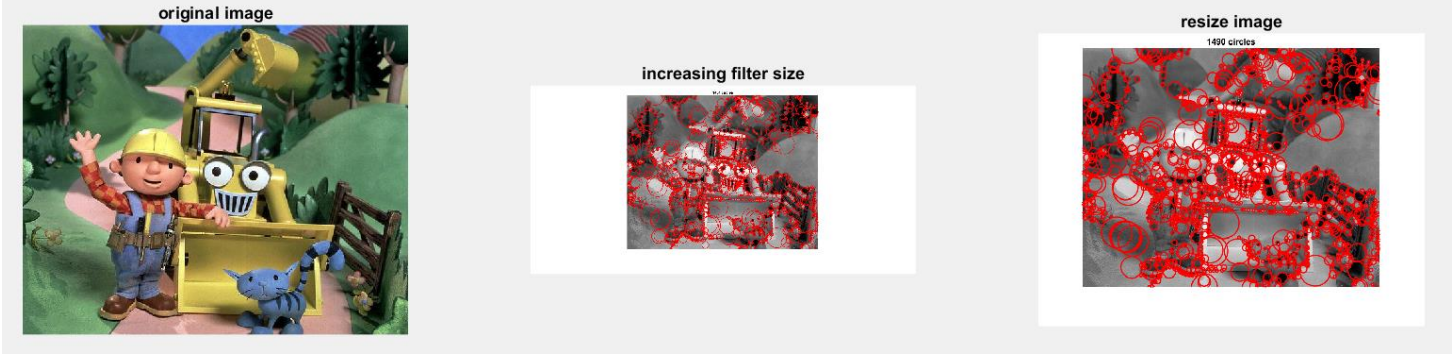

Figure 1.6

- **Image7:** *'../data/img5.jpg'*

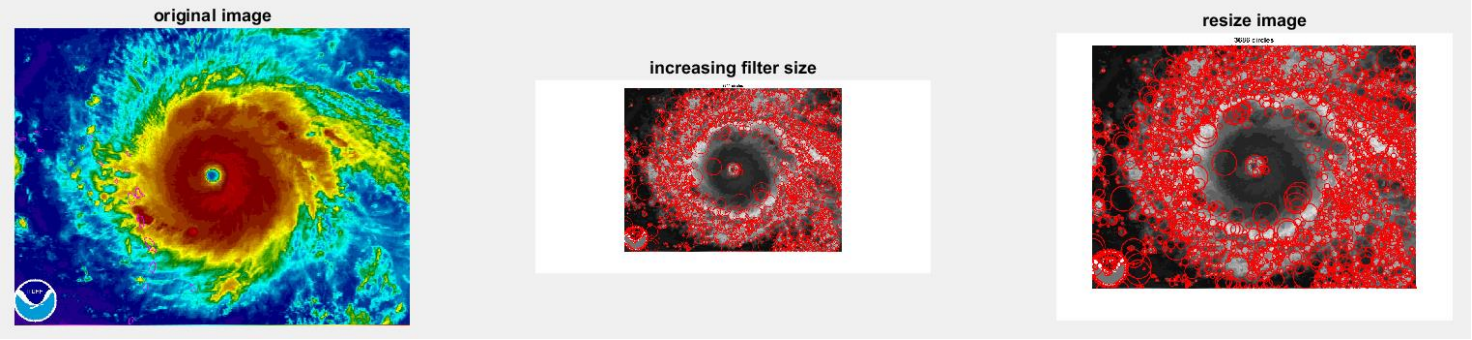

Figure 1.7

- **Image8:** *'../data/img9.jpg'*

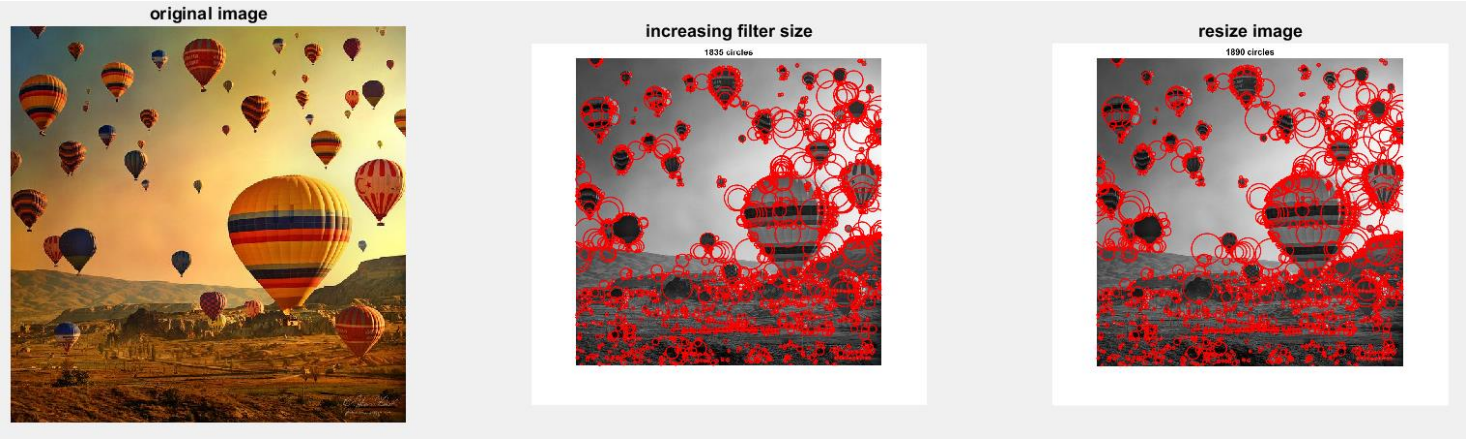

Figure 1.8

- Comparison of running times for both the methods

Table 2

	Running Time (of the whole program) in seconds		Running Time (for getting the LoG filtered scale space) in seconds	
Image	Method 1 – Increasing filter size **(Inefficient)**	Method 2- Resizing image **(Efficient)**	Method1	Method2
'../data/butterfly.jpg'	2.3012	1.1454	**1.213936**	**0.115909**
'../data/einstein.jpg'	3.0780	1.5416	**2.009072**	**0.149951**
'../data/fishes.jpg'	1.8260	0.7936	**1.254074**	**0.110516**
'../data/sunflowers.jpg'	1.8595	1.0680	**0.975982**	**0.089000**
'../data/img1.jpg'	2.0641	0.8051	**1.598446**	**0.147358**
'../data/bob.jpg'	3.5092	1.5460	**2.035849**	**0.140352**
'../data/img5.jpg'	7.1524	4.3772	**3.223047**	**0.191624**

'../data/img9.jpg'	5.1977	2.9016	**2.526978**	**0.181454**

As we can see from the running time for both the implementations, if we obtain the LoG filtered response by downsampling the image at each level and then performing non maximum suppression on the up-sampled images, we are reducing the running time as compared to when we are increasing the filter size at each level. Thus, the downsampling of image approach is **efficient** as compared to increasing filter size (**inefficient**).

2. Different Implementation choices

a. When applying non maximum suppression on each 2D slice (at each level) separately, and then looking for maximum across all levels, we get a problem that many adjacent circles are being overlapped and this ends up displaying redundant blobs across the image.

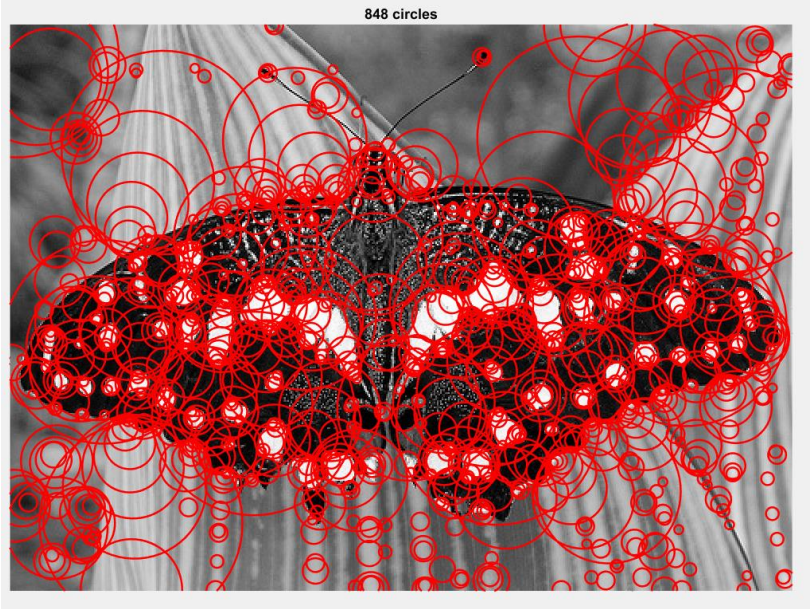

Figure 2.1

However, if we perform the non-maximum suppression across the 3D scale space (:,:,N), we get a cleaner output which displays the blobs accordingly.

b. Performing max before applying non maximum suppression
Performing non maximum suppression on a 3D array is complex. Also, we would get sort of similar output as performing max and then taking non suppression maximum instead of performing non maximum suppression and then taking maximum values across scale space.

3. Different Parameters used

The values across which I found the output to be optimal is specified in the ***'Table1'*** above.
I tried different values of k (2^0.1, 0.25) and found that for these values I am getting too many blobs which are almost of same size. This might be because, for these values sigma is not getting affected that much at each level and we are getting circles of almost similar radiis.
I also tried different values of sigma and threshold
Sigma values tried: 0.25, 3, 5
Threshold values: 0.001, 0.01, 0.002, 0.005, 0.1

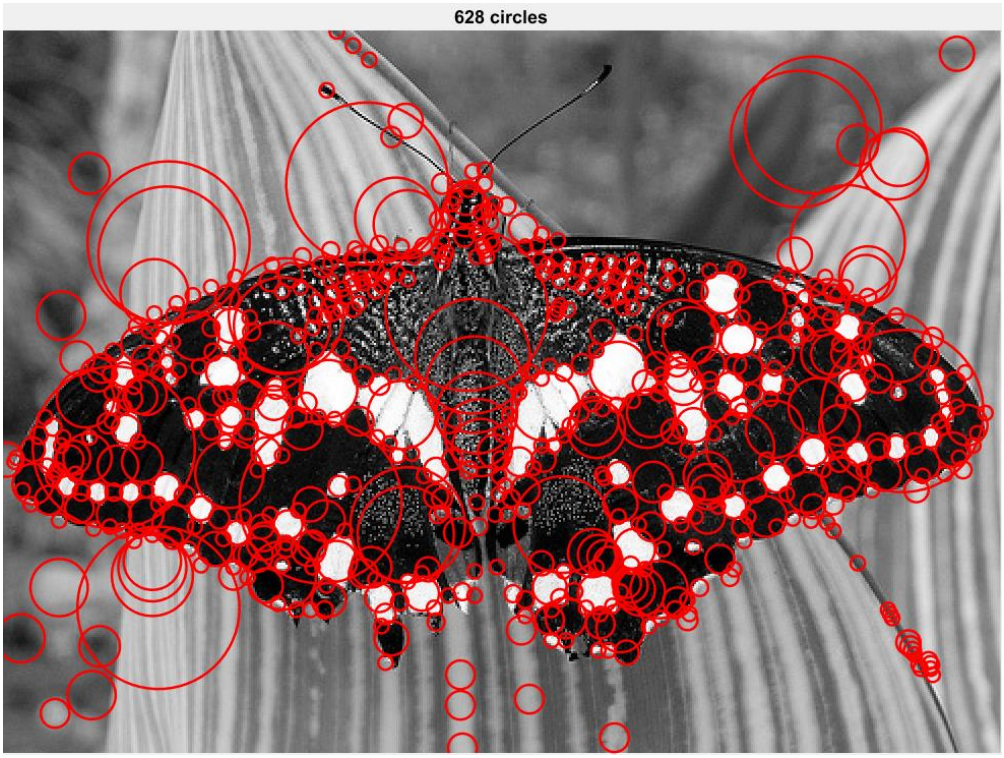

Figure 3.1- Sigma: 2.5, k: 2^0.25, and threshold: 0.02

Since the threshold is big, we are not getting many blobs in the background (gray areas)

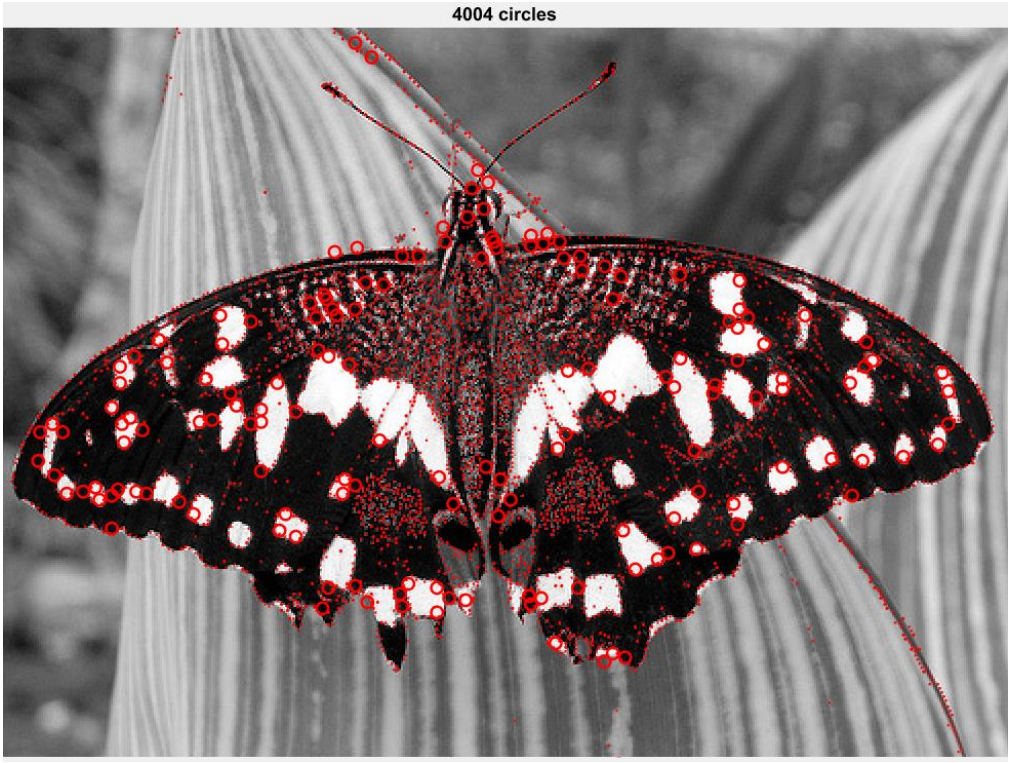

Figure 3.2- Sigma: 2, k: 0.25, and threshold: 0.02

Again, blobs are not being displayed in the background properly, and the circles are almost of similar sizes.

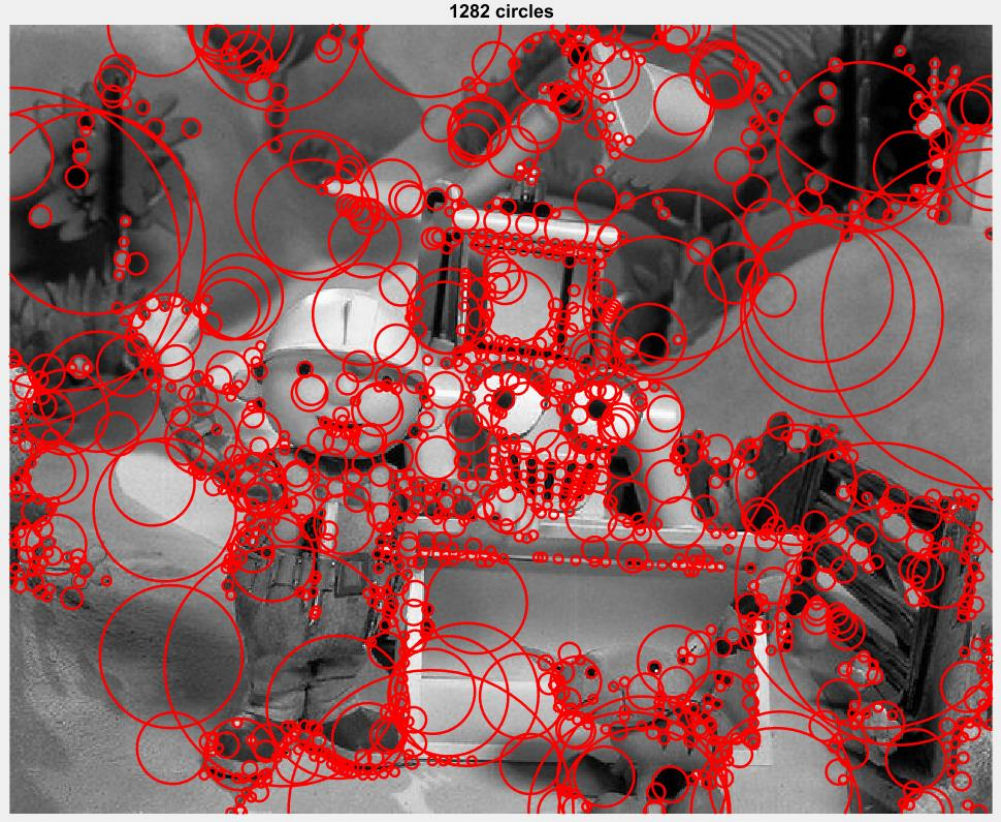

Figure 3.3- Sigma: 2, k: 2^0.25, threshold: 0.005 and N:30

If we are increasing N, the sigma values at the higher levels will be large and hence we are getting circles or blobs with bigger radii.

- To Run the program

Call main function (in 'main.m' file')

The parameters to be passed to the main function are:

- Image path – relative path to the to the image from the current location
- Sigma
- N
- Threshold
- K
- Method (1 – increasing filter size, 2 – downsampling image)

- ***main('../output/bob.jpg',2,15,0.005,2^0.25,1)***

- Scripts in the '**matlab'** directory

- ***main.m*** – main function to call
- ***increase_filt_size.m***- returns the LoG filtered scale space by increasing filter size at each level, and also returns the array of different sigma values at each level
- ***downsample_image.m***- returns the LoG filtered scale space by resizing the image (continuously multiplying by (1/k)) at each level, and also returns the array of different sigma values
- ***nonMaximum.m*** – performs non maximum suppression using ordfilt2 (for a window size of 3) and returns the output matrix
- ***harris.m*** – sample Harris detector (already given)

- ***show_all_circles.m*** – to show blobs on the image (already given)

References:

- https://courses.cs.washington.edu/courses/cse576/06sp/notes/Interest2.pdf
- http://www.cs.utah.edu/~manasi/coursework/cs7960/p1/project1.html