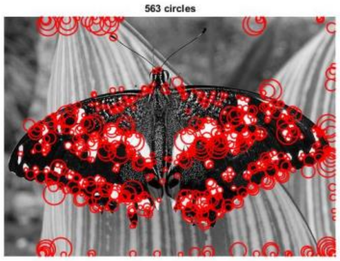
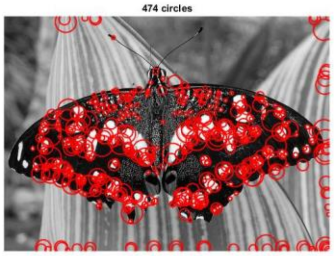
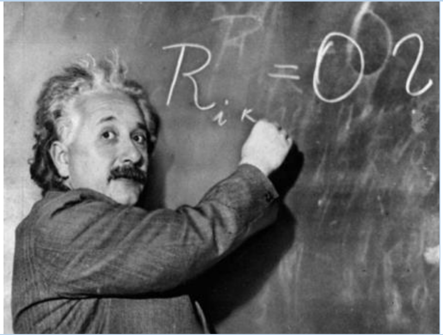
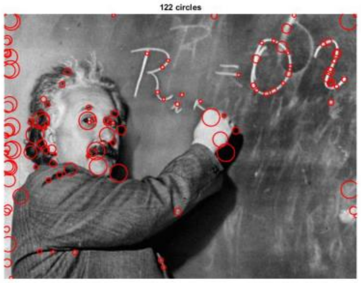
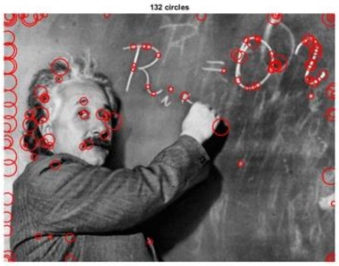
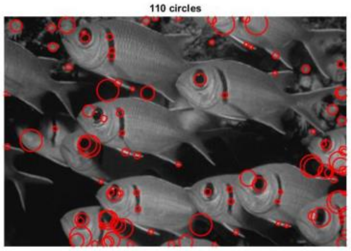
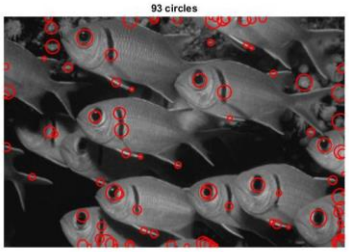
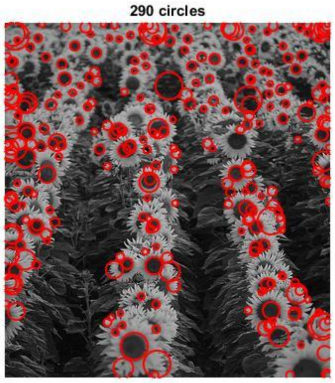
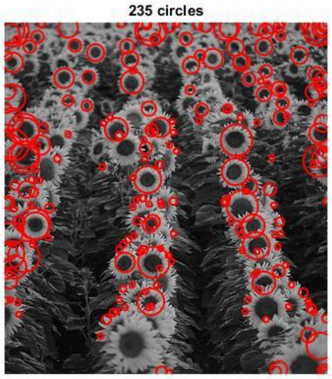
CSE 573: Computer Vision an Image Processing
Homework 2

Submitted by: Ketan Shah
Person number: 50247131

1. **Following are the blobs detected on the sample images provided:**

Original Image	**Upsampled approach**	**Downsampling approach**
		
		
		
		

CSE 573: Computer Vision an Image Processing
Homework 2

Submitted by: Ketan Shah
Person number: 50247131

2. **Following is the timing comparison :**

Picture name	Time taken for up sample approach	Time taken for down sample approach
Butterfly	0.3920	0.0653
Einstein	0.6606	0.0945
Fishes	0.3825	0.0720
Sunflowers	0.2835	0.0581

It is worth noting that the time taken for finding filter responses in case of the up sampling approach is almost 4 times more than the down sampling approach. This shows that increasing scale is not a very efficient method. Since we had very few levels and smaller images there wasn't much computing cost penalty but if we have larger images and more levels in the scale space this difference will impact the performance a lot. Suppose we had to run the code for 1000 images. The small time difference in each iteration would add up to a huge difference.

3. **Parameters I used**

   For scaling factor k I found it better to keep it in the range 1.1 to 1.2 since k is a factor that affects the size at every level.
   If we take too large value of k each level in the up sampling approach will take larger time to execute since the last few level size will start increasing exponentially.

   The threshold value totally depends on the image and I've kept it as a parameter to be passed to the function for calculating blobs.

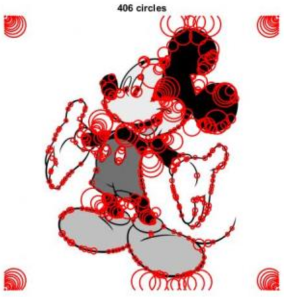
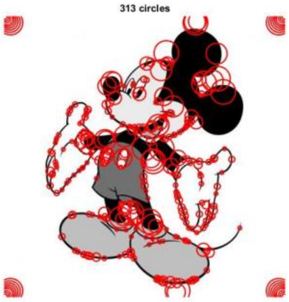
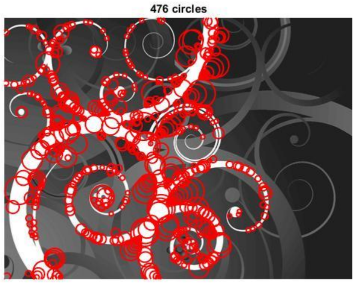
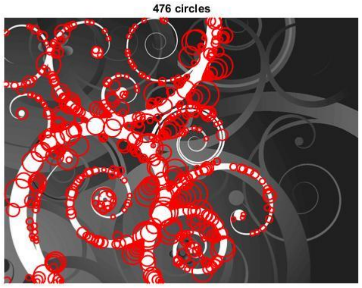
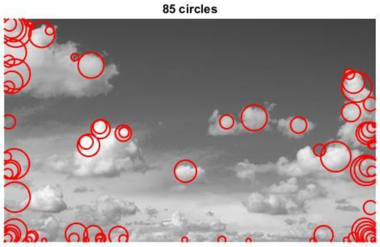
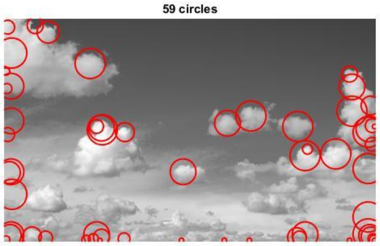
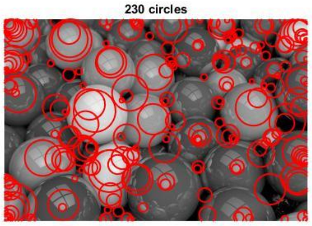
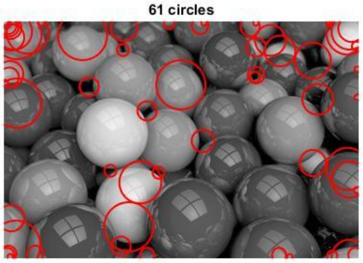
   The number of levels I found optimal for these kinds of images were 12-13. Taking too much large value is of no use since in the later layers the response comes very close to black. Also I noticed that the computation cost in the last few layers increases really fast since the size of the kernel starts getting bigger.

   The sigma value I started with 2 and increased it by scaling factor k every turn.

CSE 573: Computer Vision an Image Processing
Homework 2

Submitted by: Ketan Shah
Person number: 50247131

4. Following are the blobs detected on four random images taken from the internet.

Original image	Up sampling approach	Down sampling approach
		
		
		
		

CSE 573: Computer Vision an Image Processing
Homework 2

Submitted by: Ketan Shah
Person number: 50247131

5. **Following are the timing comparisons of images I used from the internet**

Picture name	Time taken for up sample approach	Time taken for down sample approach
Mickey	0.6349	0.0732
Design	0.6520	0.0966
Sky	0.4226	0.0561
Balls	0.6240	0.0510

Here the timing observation is similar to what we had in the first set of sample images.

6. **Instructions for running the code.**
Two main functions in the code directory are used for detecting the blobs.

A. The function **blobDetect** is used to detect the blobs using the inefficient method

```
function blobDetect = blobDetect(imgpath, k,  numLevels, threshold)
```

The parameters to be passed to this function are:

a. Imgpath = The path to the image in a string format
b. k = scaling factor(double or int)
c. numLevels = number of levels you want in the scale space (integer value)
d. threshold = the threshold you want to set for blobs

B. The function **blobDetectDownSample** is used to detect the blobs using the inefficient method

```
function blobDetectDownSample = blobDetect(imgpath, k,  numLevels, threshold)
```

The parameters to be passed to this function are:

e. Imgpath = The path to the image in a string format
f. k = scaling factor(double or int)
g. numLevels = number of levels you want in the scale space (integer value)
h. threshold = the threshold you want to set for blobs

7. **Interesting observation and implementations**

a. The first observation was that the number of blobs in the up sample approach is higher than the downsample approach for similar threshold.
b. Many circles were turning up at the same spot with different sizes.
c. To reduce this number of circles I wrote a simple function called reducedRadiiCal which eliminates close points and keeps only one of them(usually the one with higher radius).
d. Also changing the size of the window for non maximum suppression affected the outcome too.