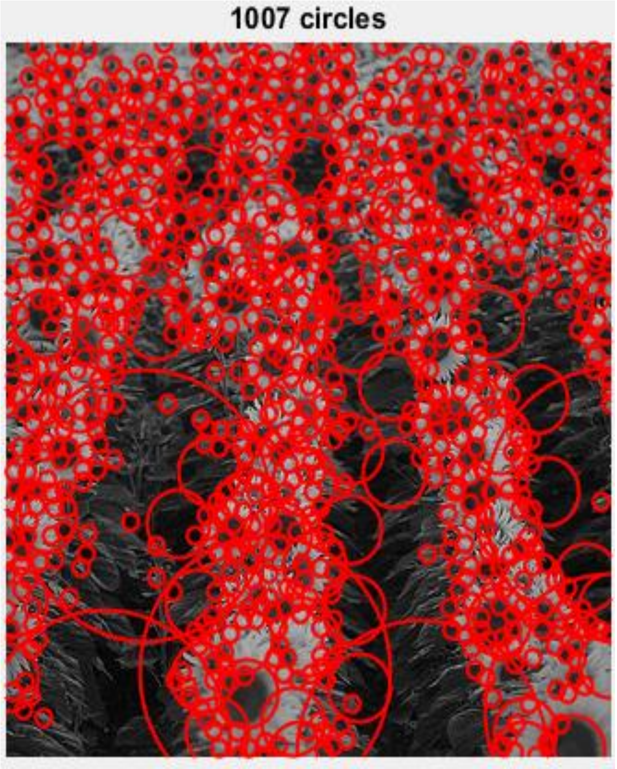
Homework 2

1)

Image 1: Sunflowers

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.007

Implementation 1, Running Time: 1.78 Seconds   Implementation 2, Running Time: 0.63 Seconds



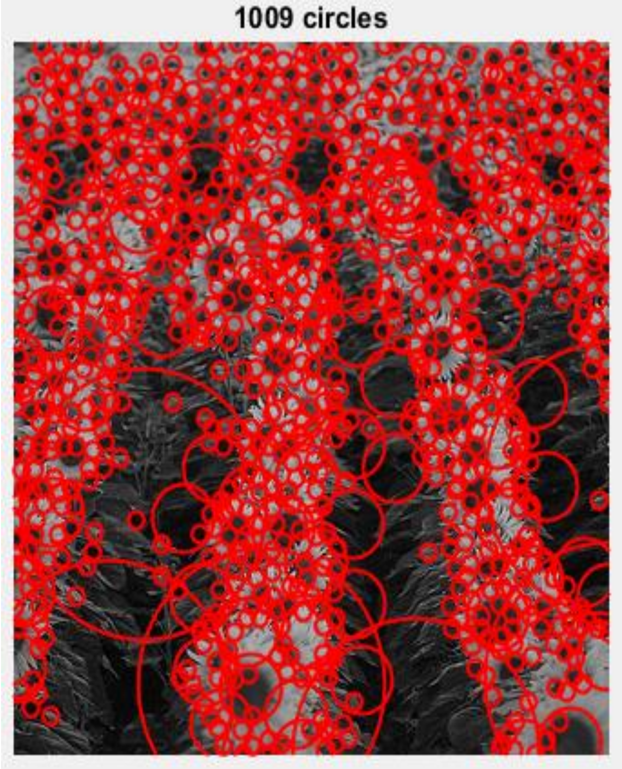
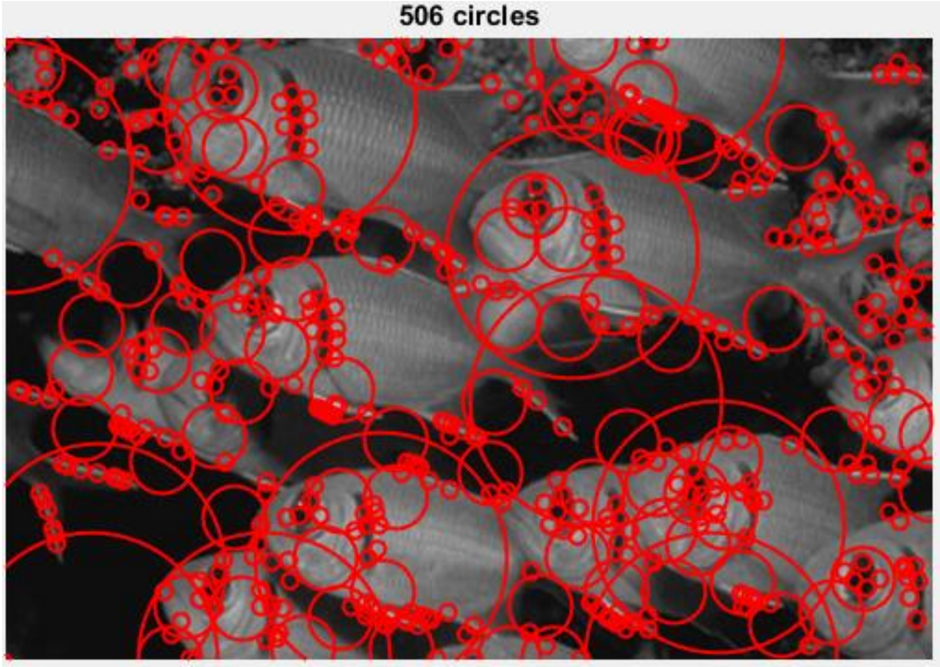


Image 2: Fishes

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.007

Implementation 1, Running Time: 2.05 Seconds



Implementation 2, Running Time: 0.43 Seconds

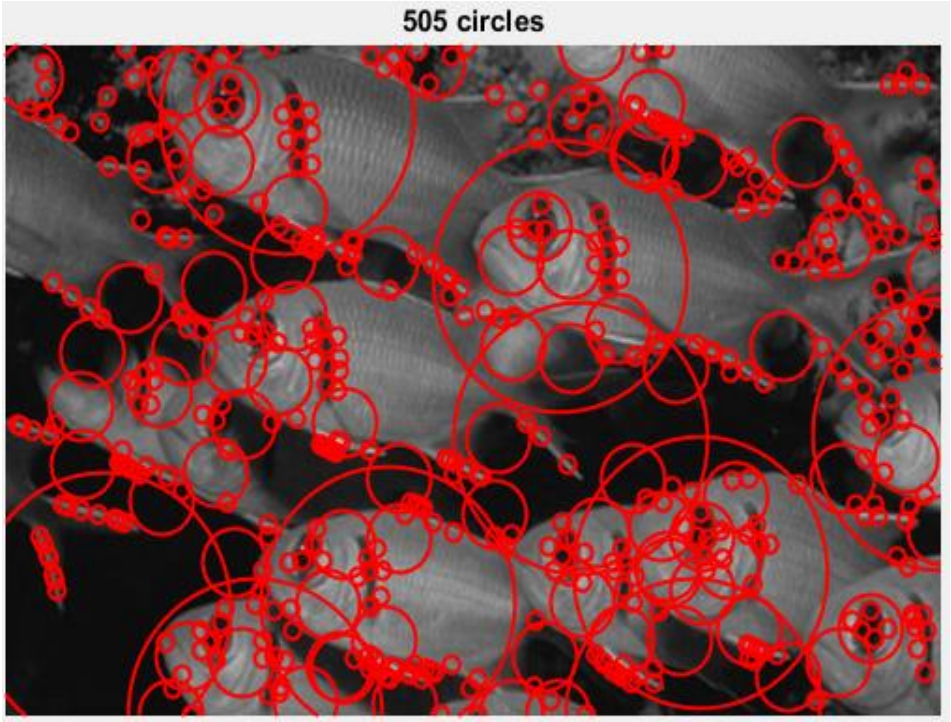
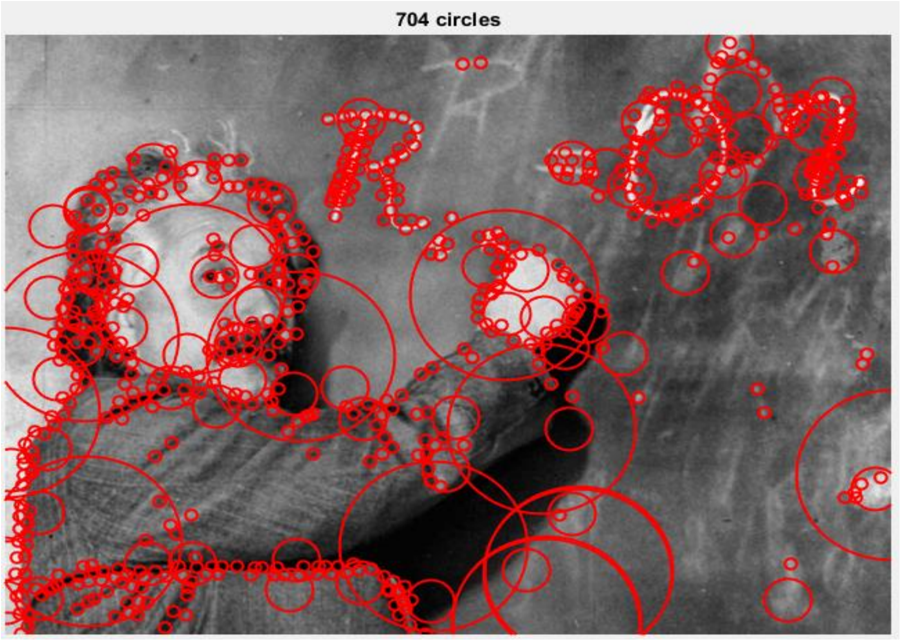


Image 3: Einstein

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.007

Implementation 1, Running Time: 3.5 Seconds



Implementation 2, Running Time: 0.57 Seconds

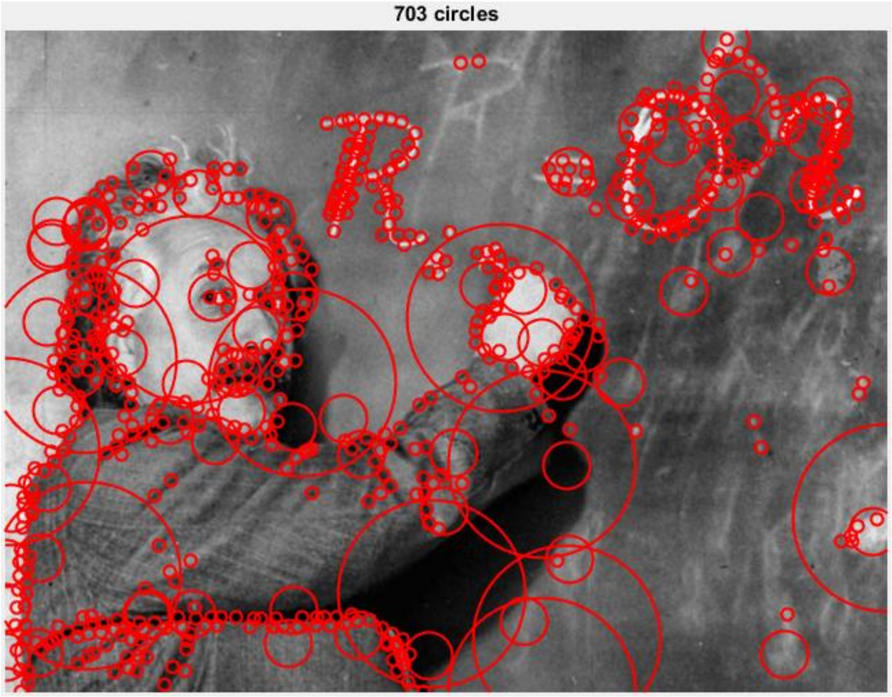
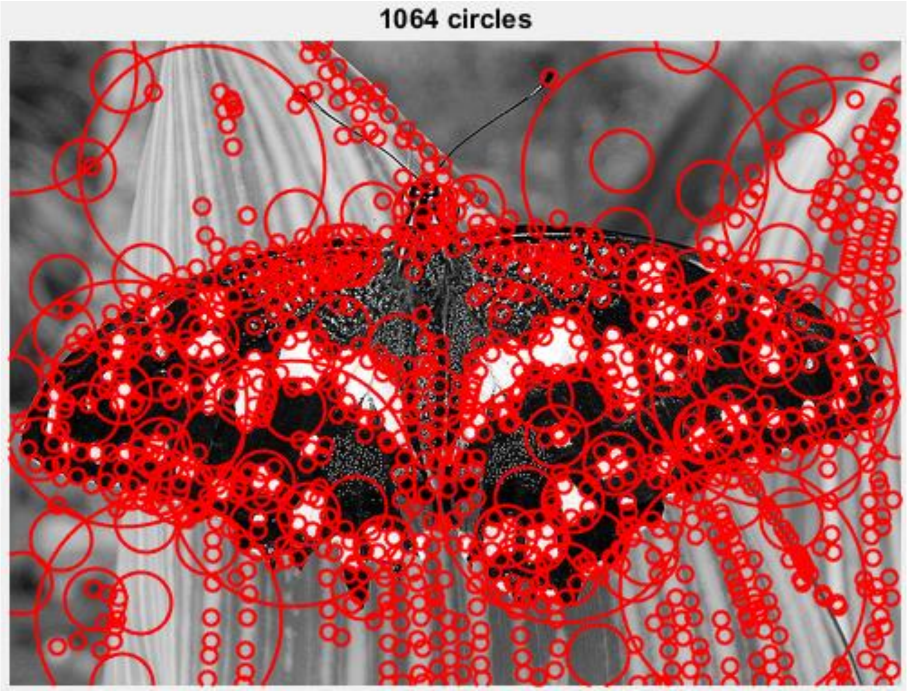


Image 4: Butterfly

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.007

Implementation 1, Running Time: 2.4 Seconds



Implementation 2, Running Time: 0.59 Seconds

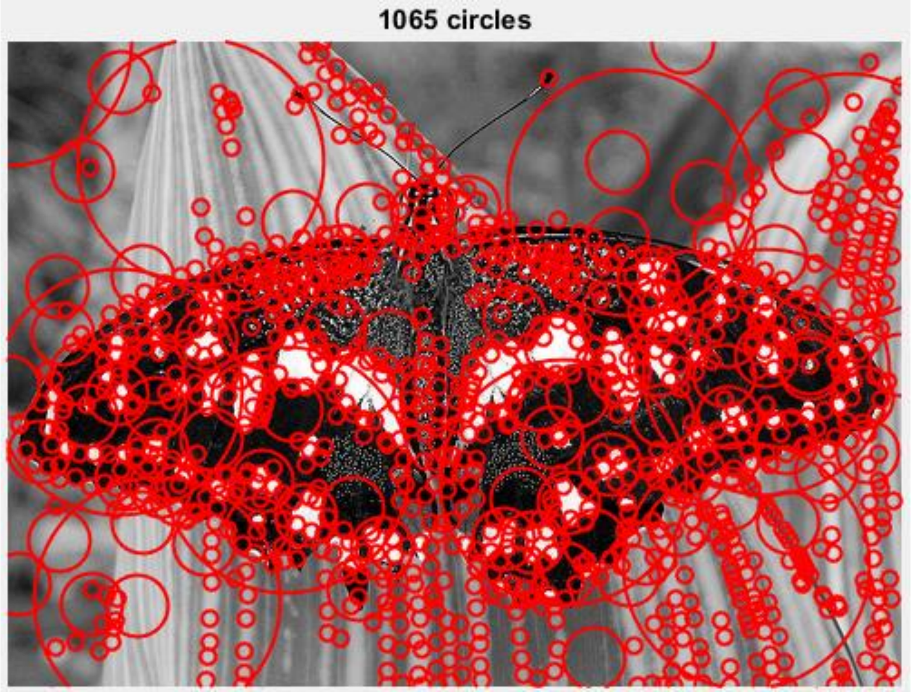
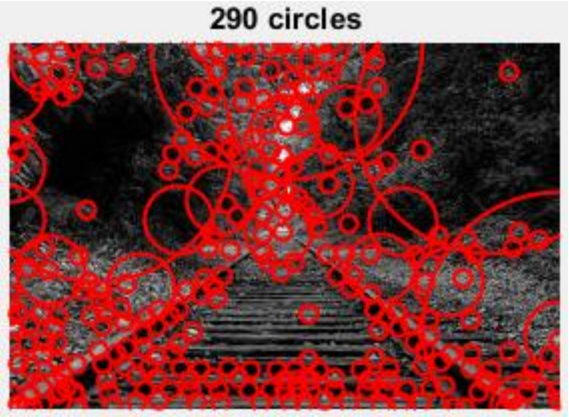


Image 5: Nature

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.005

Implementation 1, Running Time: 0.84 Seconds Implementation 2, Running Time: 0.33 Seconds



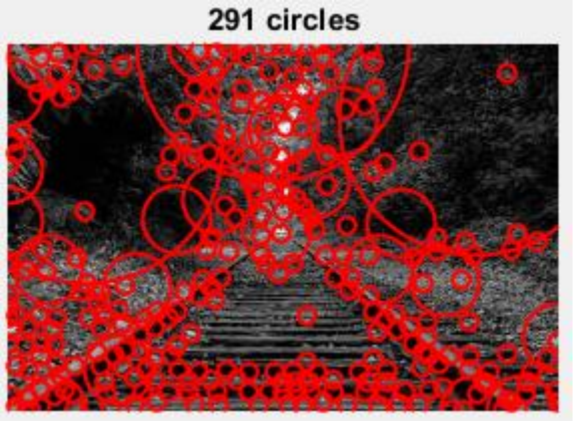
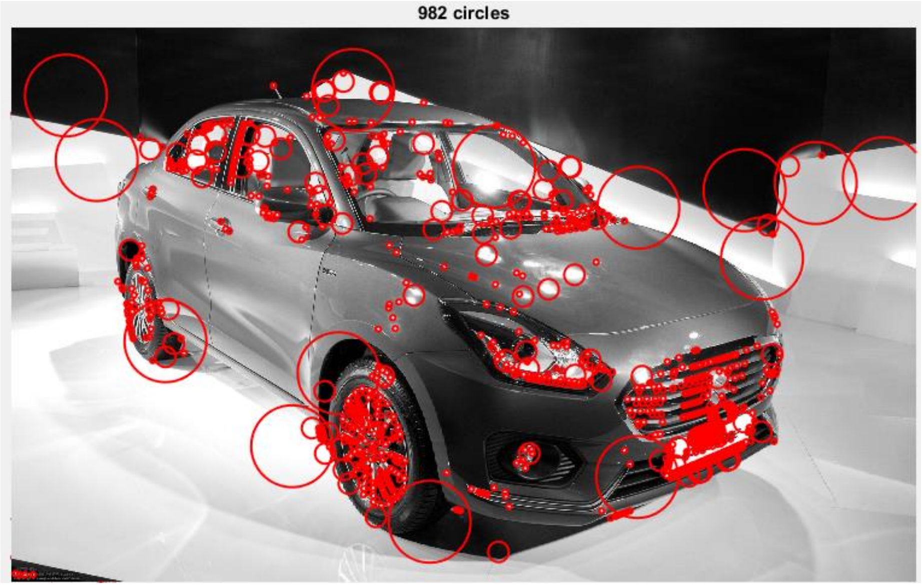


Image 6: Car

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.4

Implementation 1, Running Time: 12.47 Seconds



Implementation 2, Running Time: 0.99 Seconds

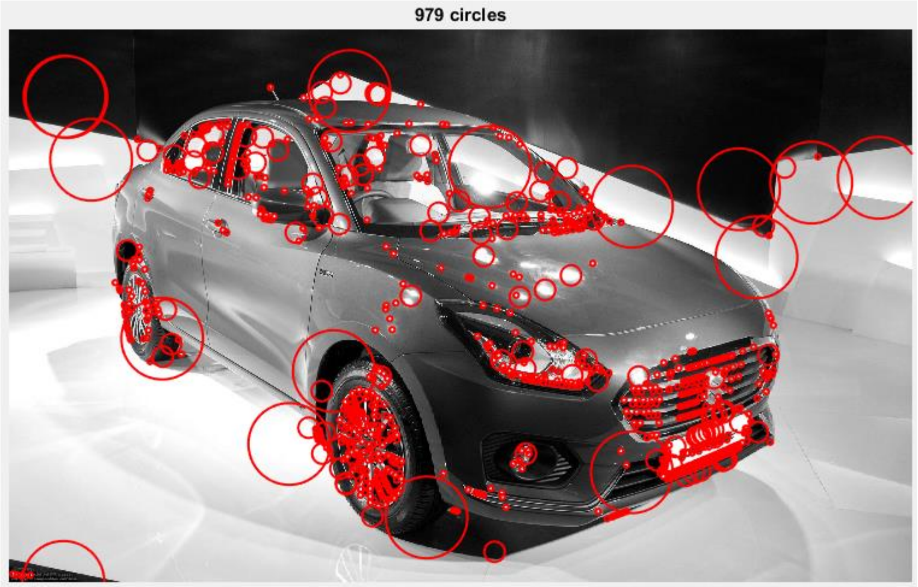
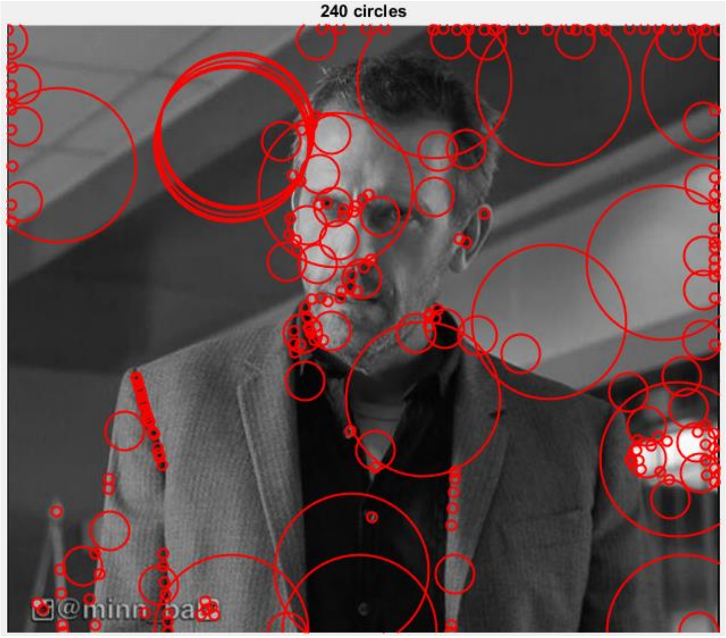


Image 7: Dr. House

Initial Sigma = 3, K factor = 4, Levels = 3, Threshold = 0.01

Implementation 1, Running Time: 3.35 Seconds



Implementation 2, Running Time: 0.3 Seconds

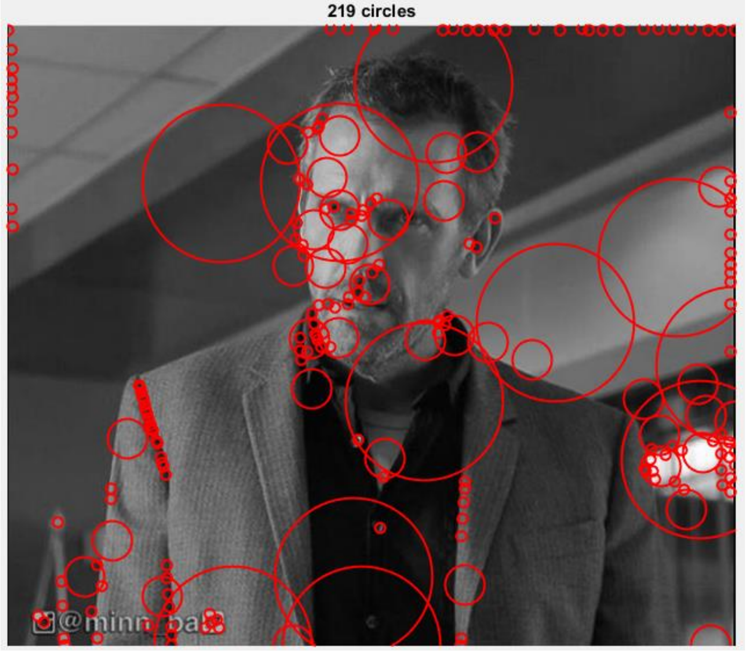
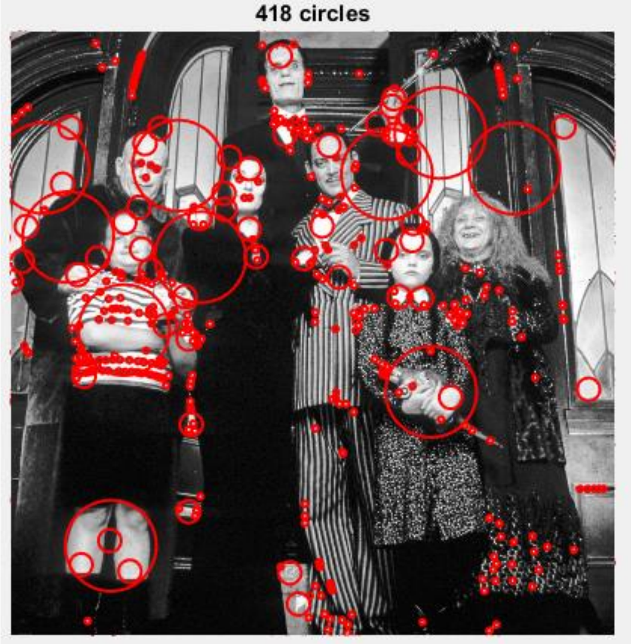


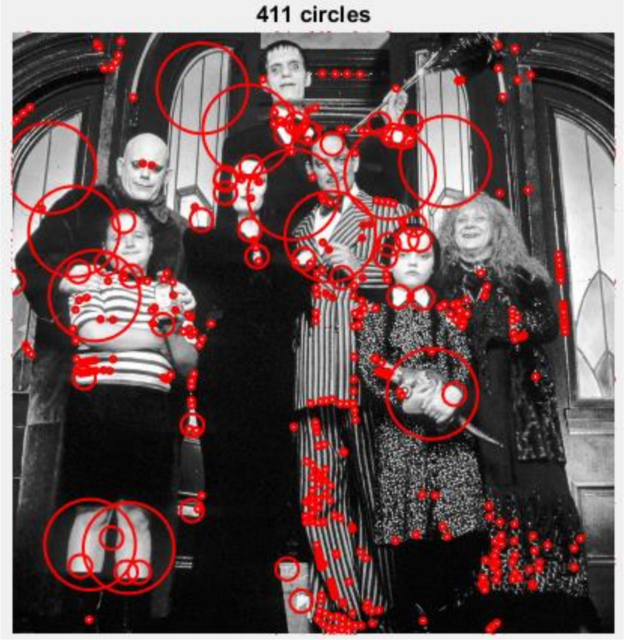
Image 8: The Addams

Initial Sigma = 4, K factor = 4, Levels = 3, Threshold = 0.0525

Implementation 1, Running Time: 24.54 Seconds



Implementation 2, Running Time: 0.87 Seconds



Comparison of Running Times: I can observe that second implementation (down sampling the image) is much faster than the first implementation (increasing sigma and filter size). However, the output for both method is almost similar for same threshold, sigma and k factor value.

2)

Implementation: I have used orffilt2 MATLAB method for non-maximum suppression implementation as I found out is relatively faster than colfilt method. For finding blobs, I used the find and max functions.

While applying LOG filter I used the ‘replicate’ property to deal with values near the border (avoided unnecessary blobs at the image border). I also used the ‘bicubic’ property during interpolation as I want to take the weighted average of neighborhood pixels.

3)

Parameter Values: I found that I was able to find adequate number and sizes of blobs for just 3 or 4 levels in the code. For more number of levels, I was getting huge number of blobs. In order to compensate that, I tried increasing threshold value, but then blobs with white background were not getting picked up. Hence I chose to have less number of levels. In fact, with more number of levels I could see that the filter size was even more than the size of image. Since I have less number of levels I chose a large k factor = 4 and initial sigma =3 to get the best output.