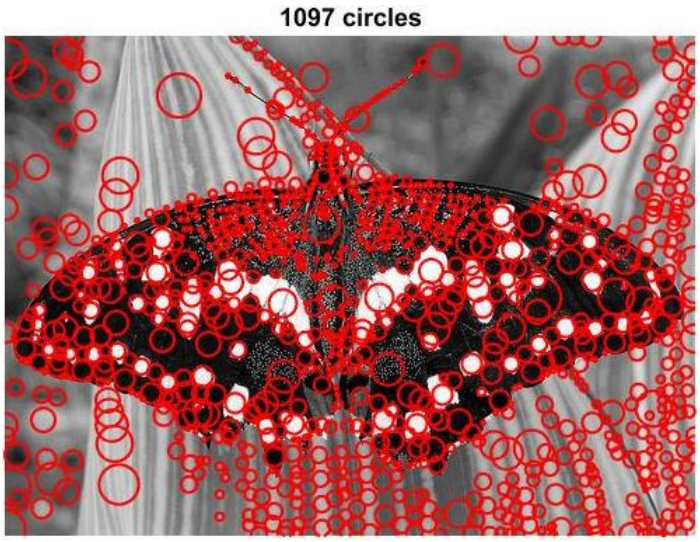
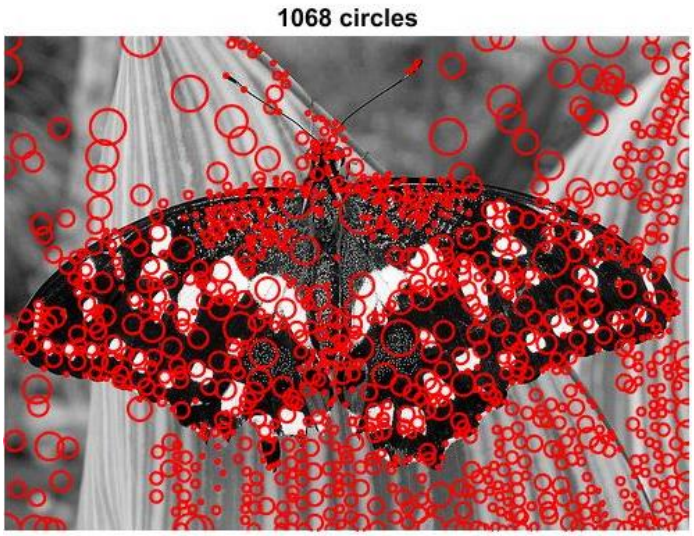
OUTPUTS OF BLOB DETECTOR

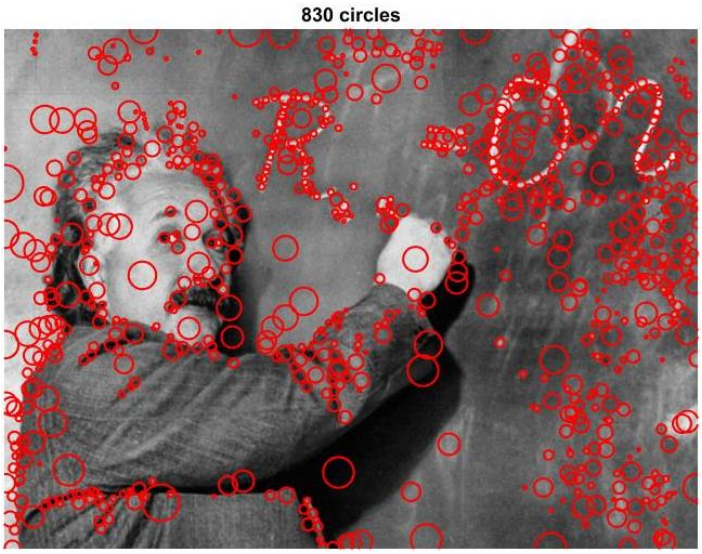
Following results were obtained for n = 12, k = 1.5, threshold = 0.001;



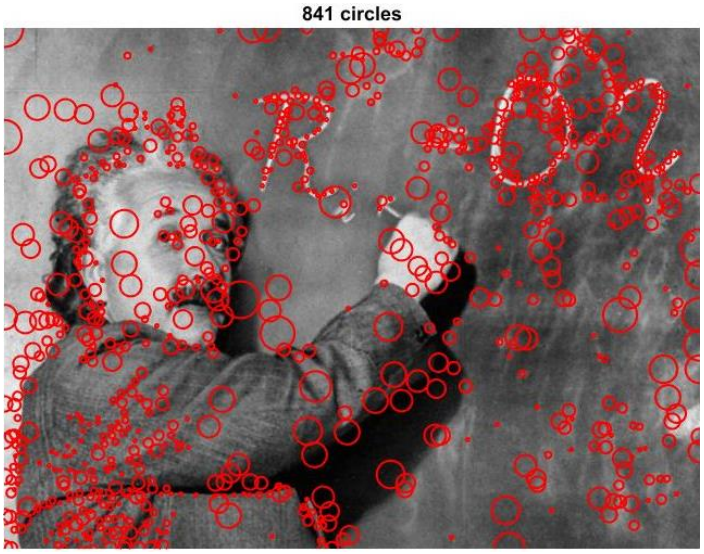
Elapsed time: 13.023967 seconds.



Elapsed time: 0.265587 seconds.

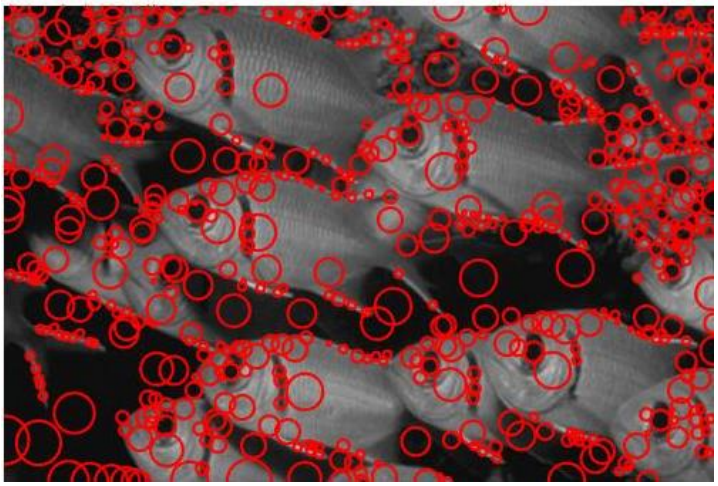


Elapsed Time: 20.918820 seconds.



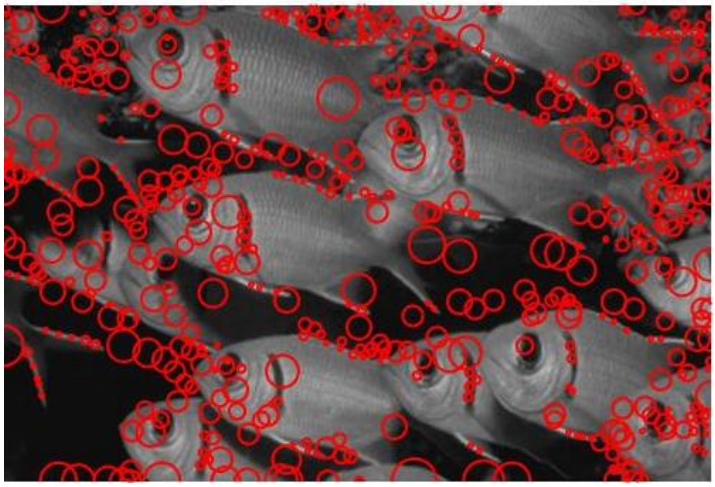
Elapsed Time: 0.069668 seconds.

650 circles



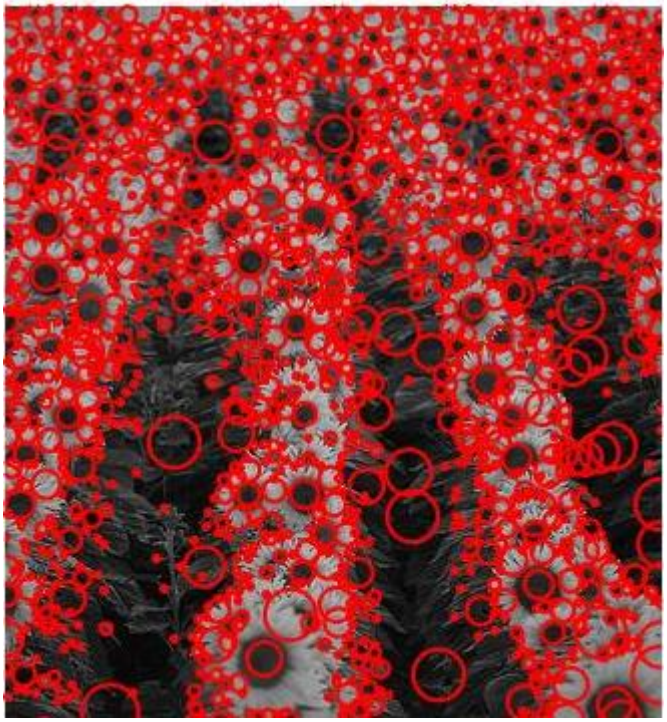
Elapsed Time: 8.746461 seconds

581 circles



Elapsed Time: 0.037100 seconds

1610 circles



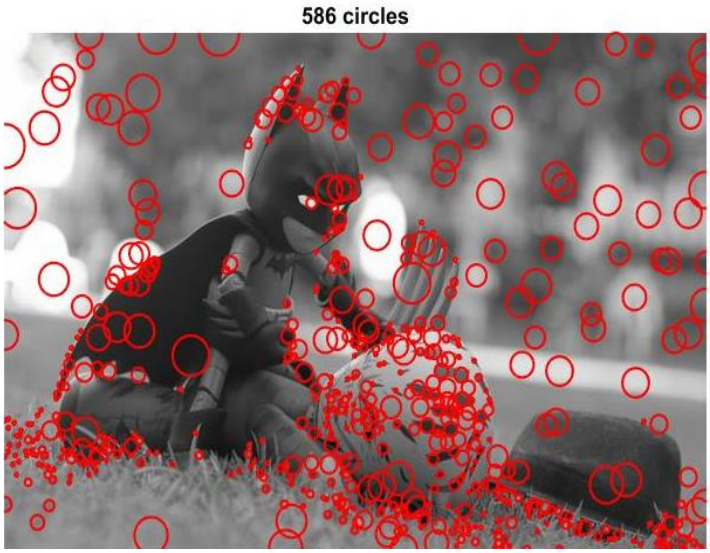
Elapsed Time: 12.441632 seconds.

1392 circles

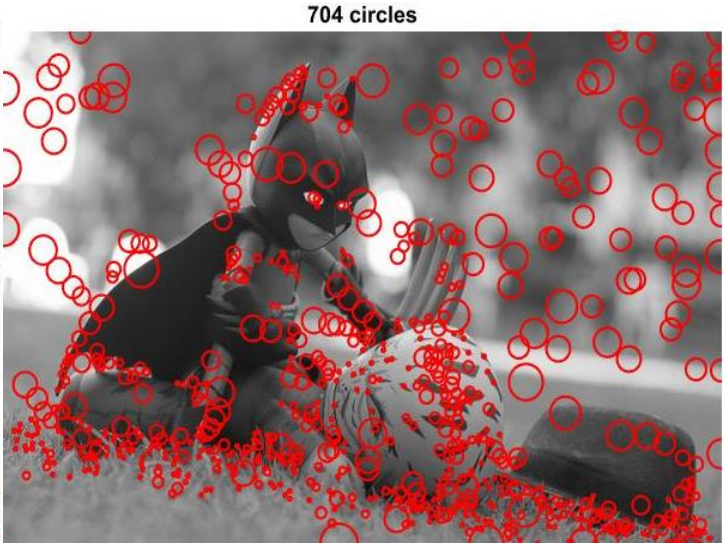
Elapsed Time: 0.049559 seconds

Elapsed time is 8.951003 seconds.

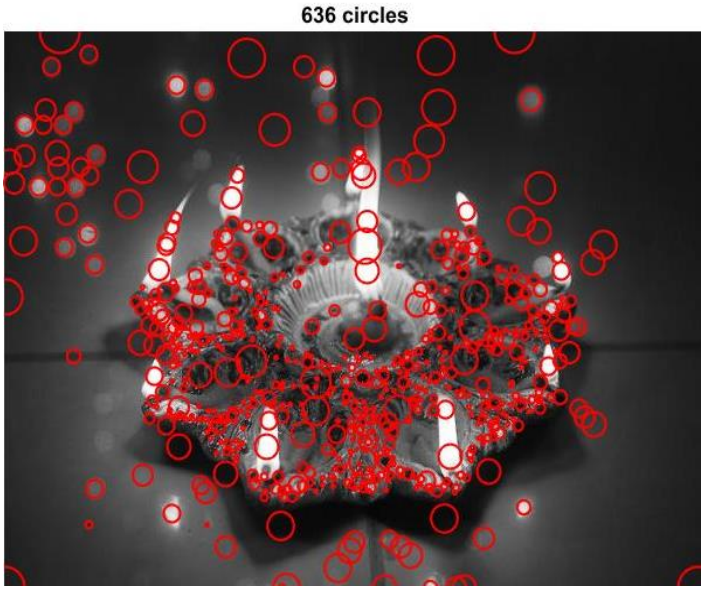
Elapsed time is 0.037473 seconds.



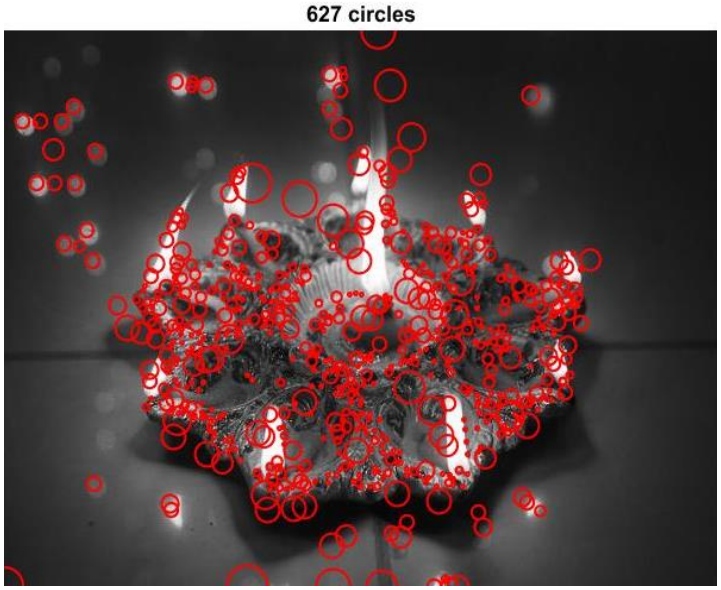
Elapsed time is 17.545752 seconds.



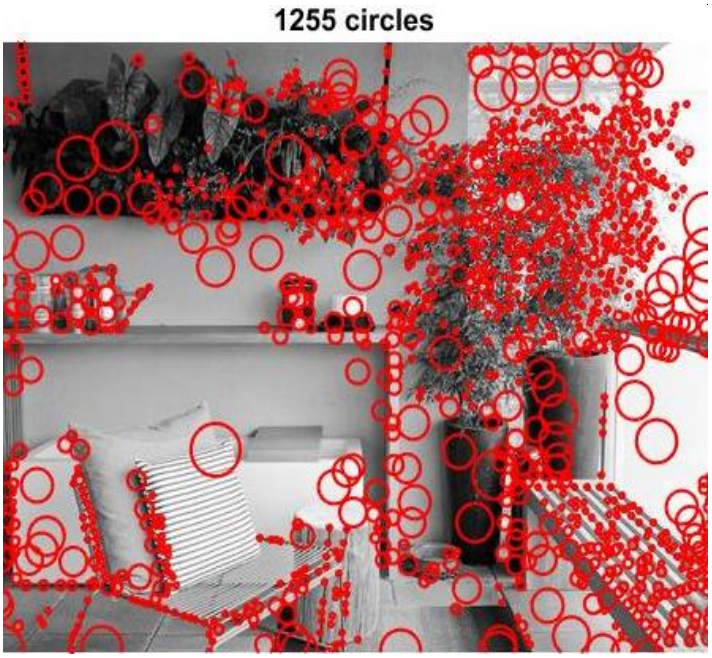
Elapsed time is 0.060780 seconds.



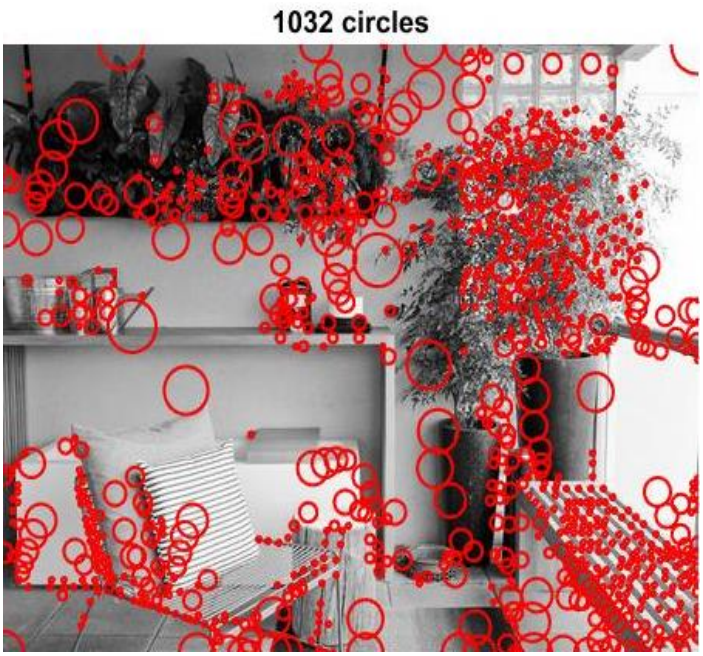
Elapsed time is 17.545752 seconds.



Elapsed time is 0.060780 seconds.



Elapsed time is 11.962951 seconds.



Elapsed time is 0.058122 seconds.

Thus the downscaling of image approach was much faster. There was difference in the plotted blobs in both cases because during downscaling and upscaling of image, there was information loss and some noise was introduced thereby. This is evident with shift in the regions of blobs and also it can be seen that there are some more smaller blobs plotted in the inefficient approach.

IMPLEMENTATION CHOICES AND PARAMETER VALUES

1) n = 12; k = 1.5; threshold = 0.001;

I found this to be a sweet spot for the 8 images used in this project.

- Increasing n led to larger scale, increasing running time

- Increasing threshold led to very few blob detections in images with low intensity in general like Tendulkar image. Decreasing it led to too many blobs being detected on well-lit images like the one of the butter fly.

- Increasing k led to ignoring of lower scales and smaller blobs and regions would not be detected and decreasing k would not allow to reach for larger regions or blobs.

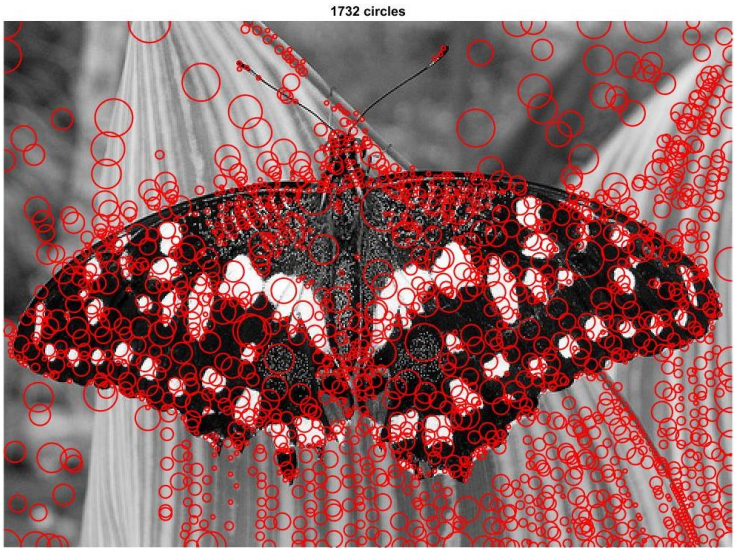
2) Use of ordfilt2 for bitmasking during nms

- Colfilt and ordfilt2 gave almost the same times and were interchangeably faster and slower but ordfilt2 was faster more often than not.
- Nlfilter was relatively the slowest

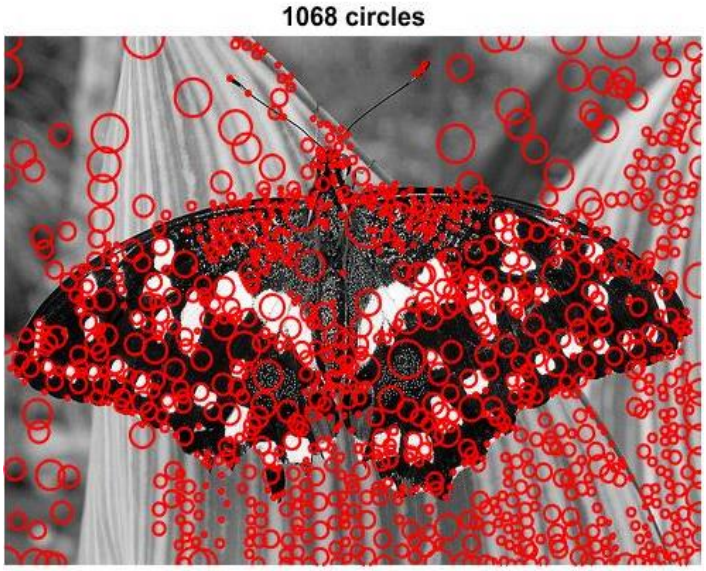
3) Use of bicubic interpolation

- The bilinear and nearest neighbor interpolation techniques produced too many blobs each time that were overlapping and redundant. The nearest neighbor technique performed the worst. Interpolation by bicubic method gave the smoothest result.

4) Use of [5 5] ordfilt window for nms response



[3 x 3] NMS window



[5 x 5] NMS window

Using the 5 x 5 window for nms gave the best result because it covered almost the same amount of region covered in the case with [3 x 3] for nms. There were a lot of intersections in the [3 x 3] case.
Similarly increasing the nmswindow to [7 x 7] or higher cause some of the regions to be missed out.

5) Radius = sqrt (2) * scale

Maximum response was observed at this value of radius. Increasing radius lead to intersection of blobs and unnecessary offsetting and decreasing radius too much from this factor had a reverse effect.