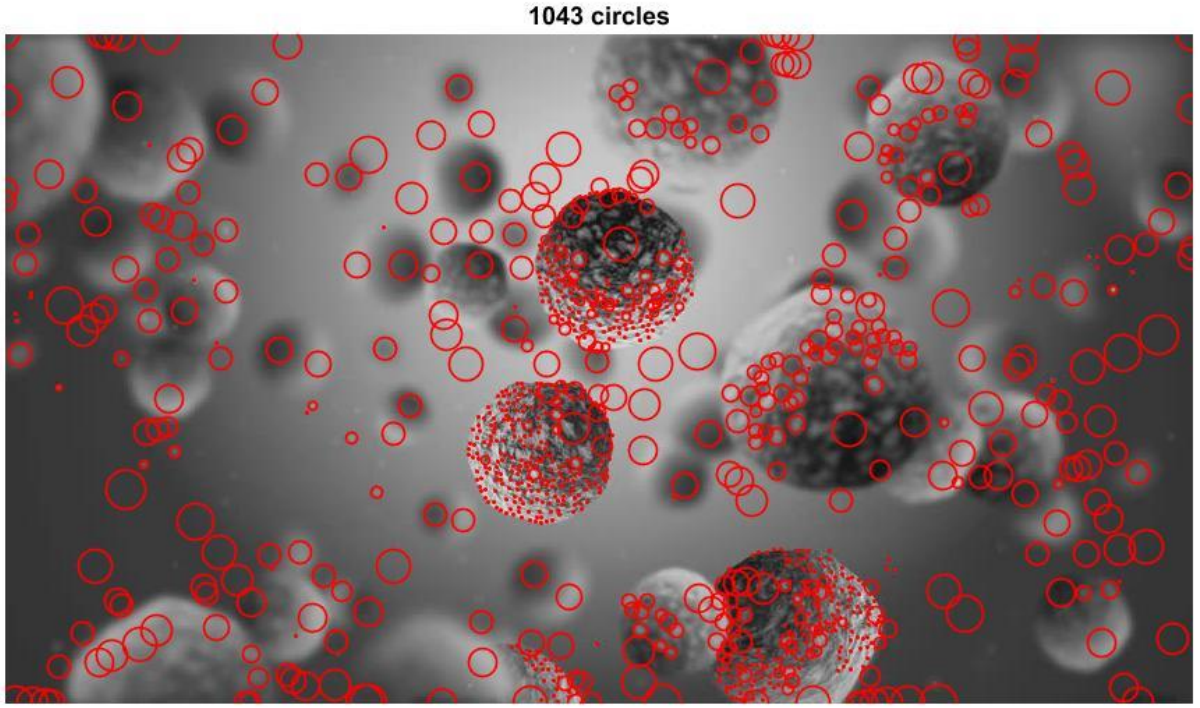
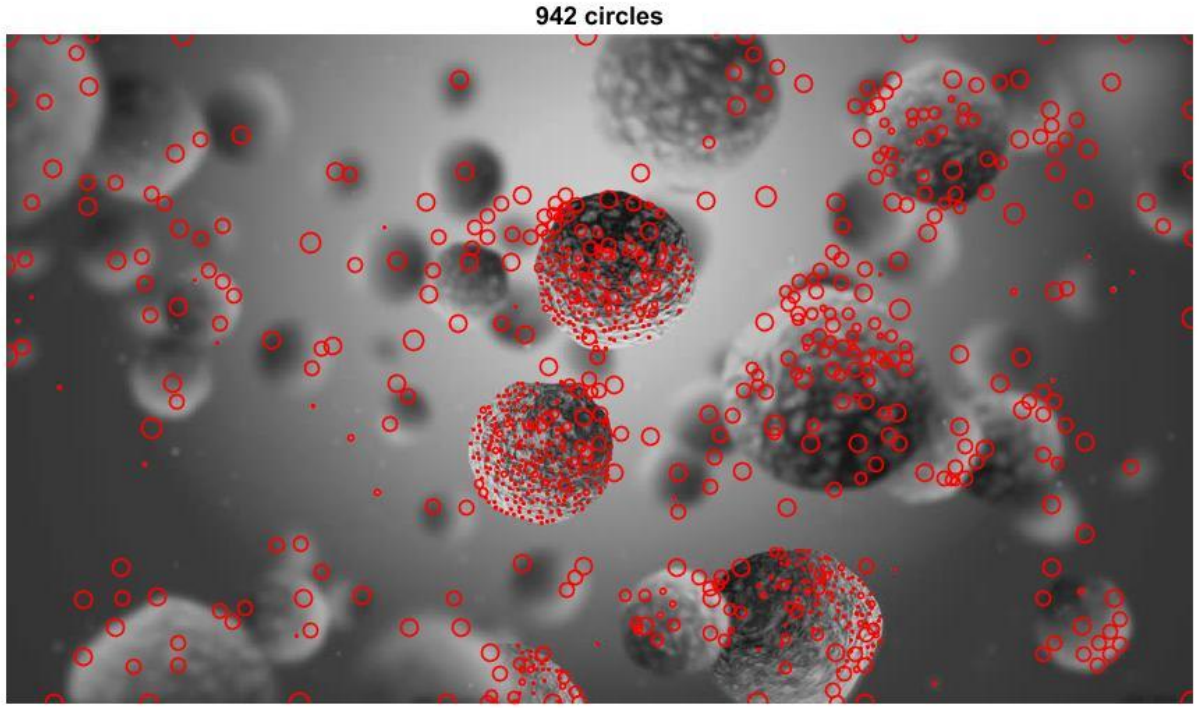




Increasing Kernel Size:
Running Time = 6.33s

Downsizing Image:
Running Time = 0.34s

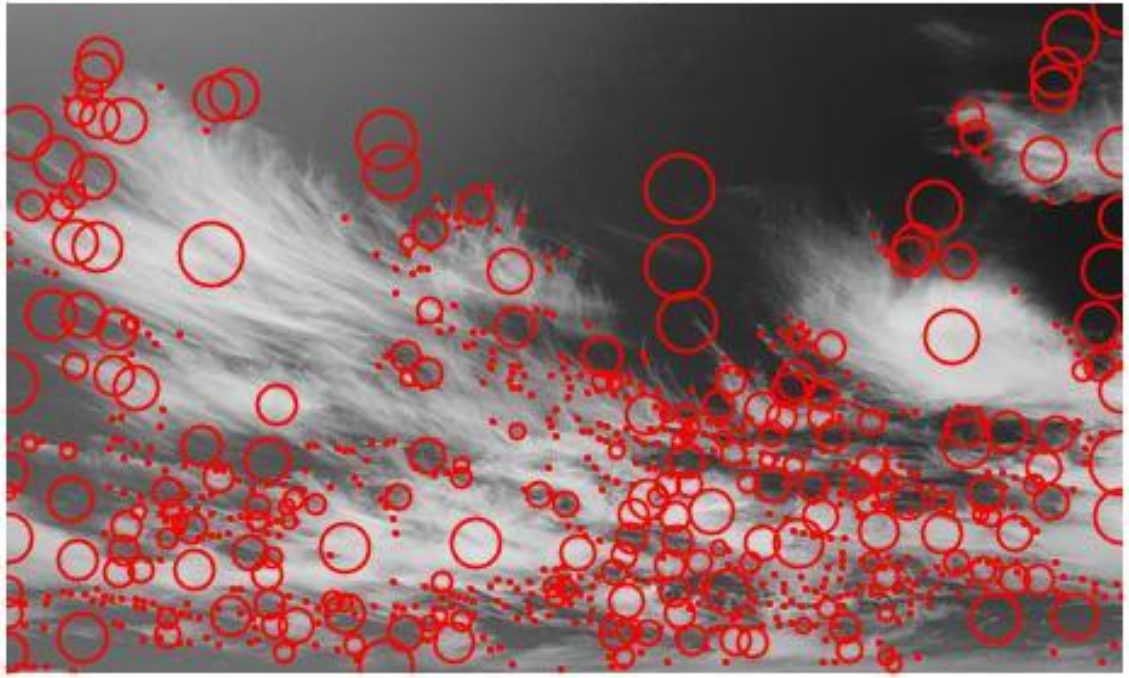
3666 circles

3418 circles

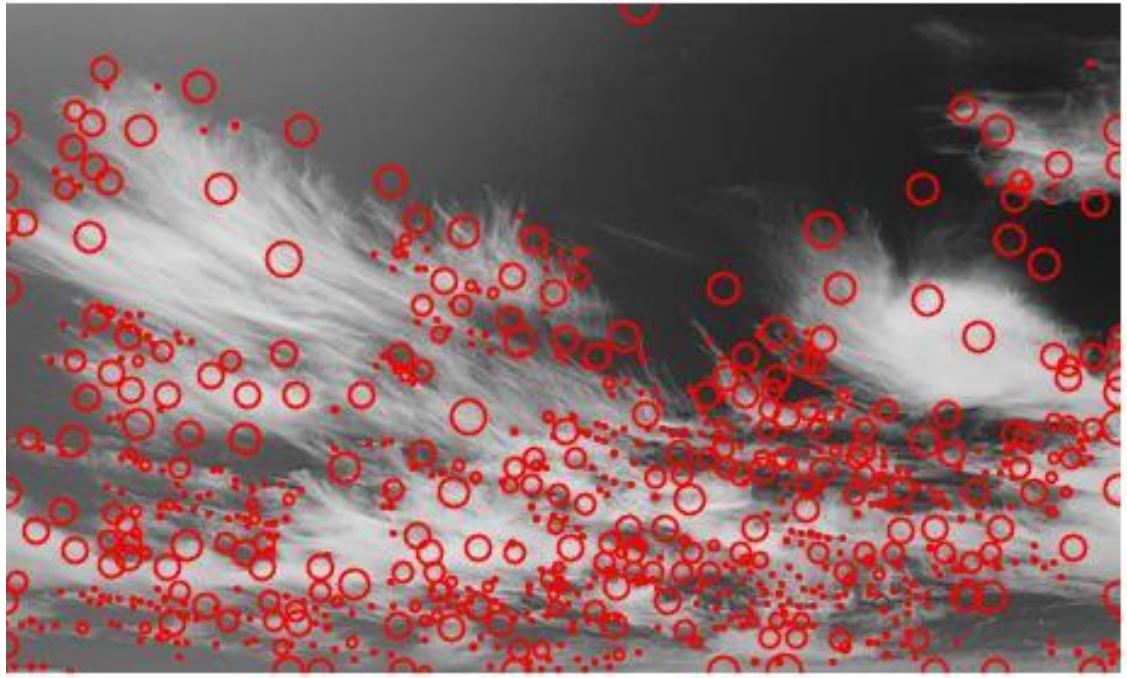
Increasing Kernel Size:
Running Time = 5.48s

Downsizing Image:
Running Time = 0.34s

805 circles

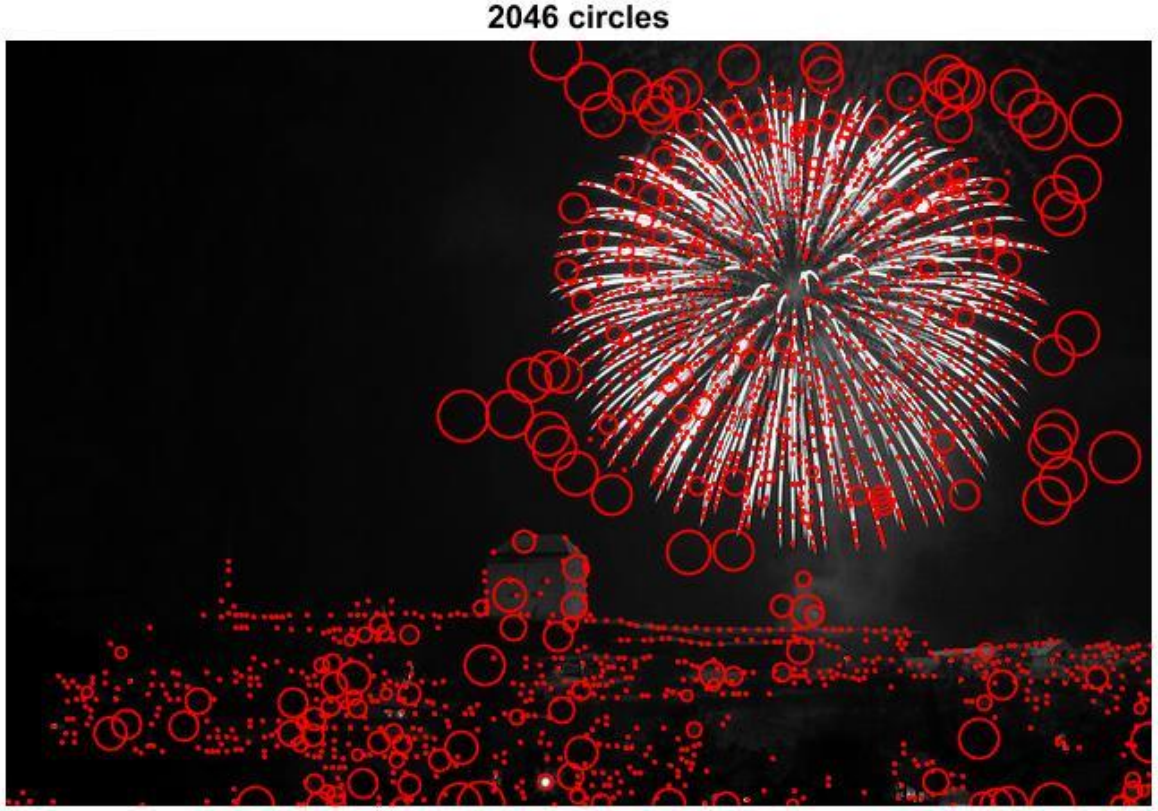


736 circles



Increasing Kernel Size:
Running Time = 2.29s

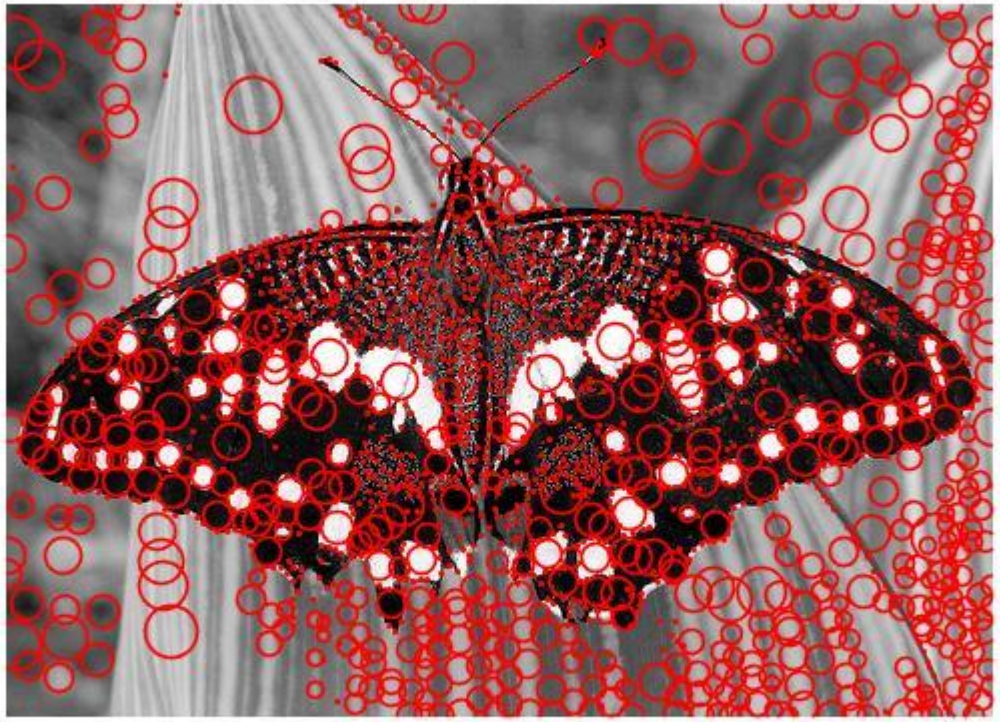
Downsizing Image:
Running Time = 0.23s



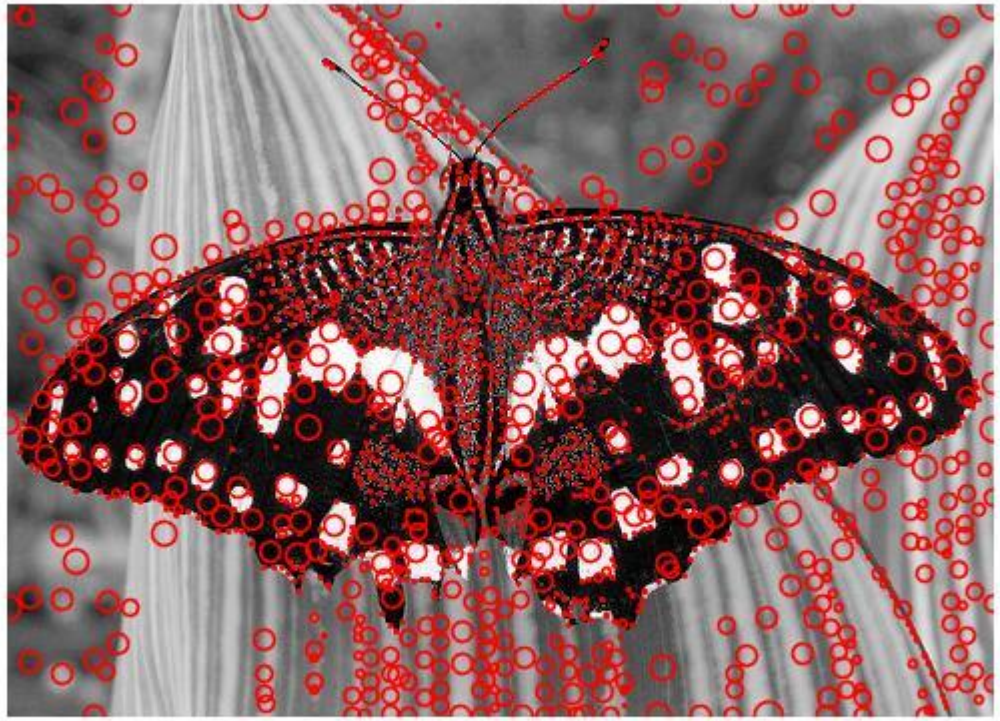
Increasing Kernel Size:
Running Time = 4.32s

Downsizing Image:
Running Time = 0.29s

1978 circles



1790 circles



Increasing Kernel Size:
Running Time = 2.86s

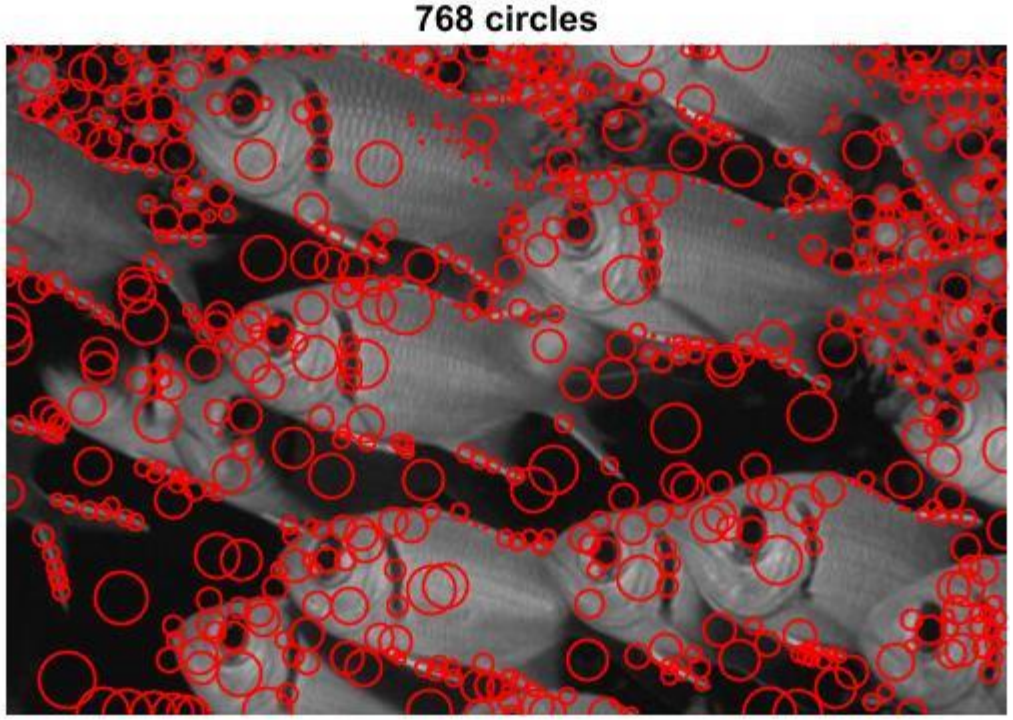
Downsizing Image:
Running Time = 0.33s

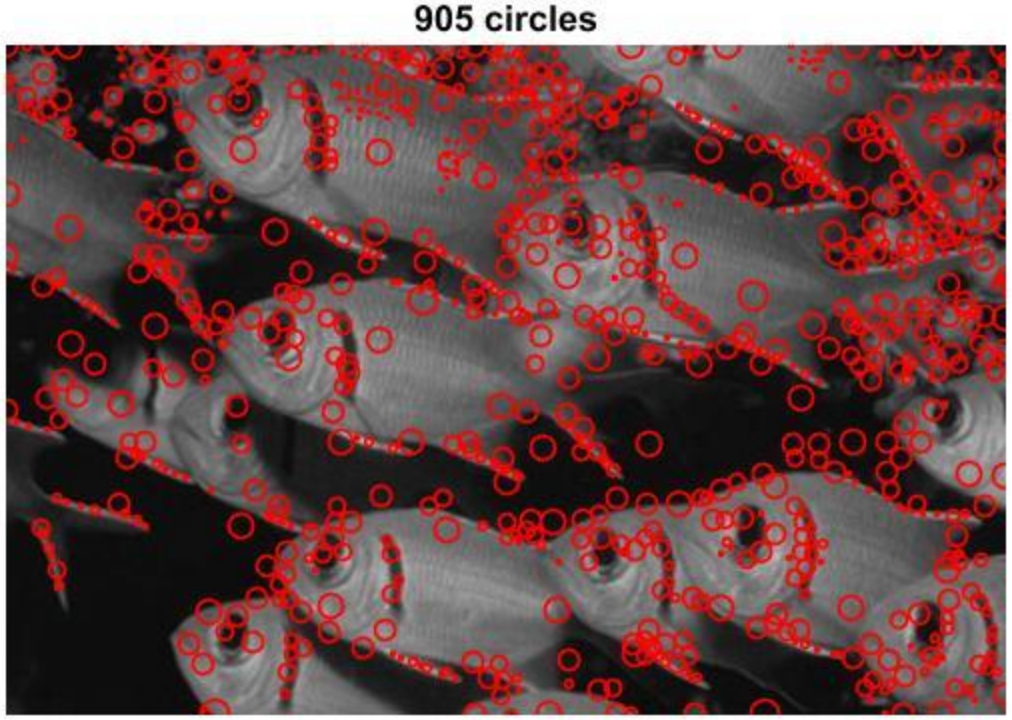
1624 circles

1571 circles

Increasing Kernel Size:
Running Time = 4.66s

Downsizing Image:
Running Time = 0.28s

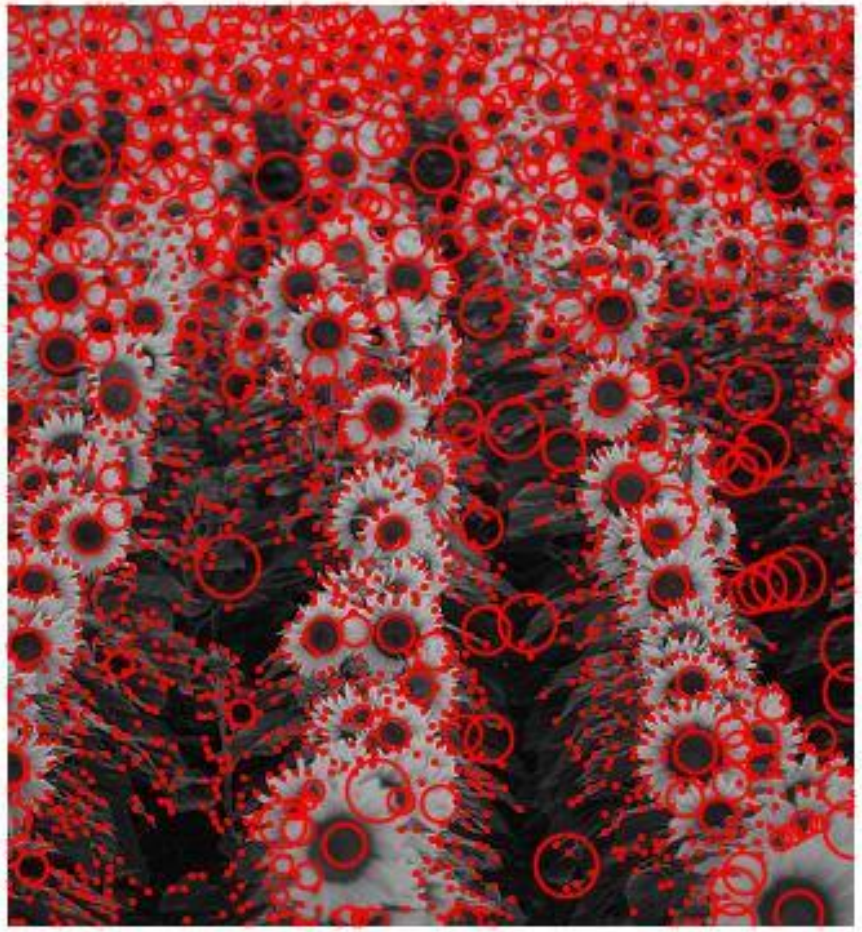




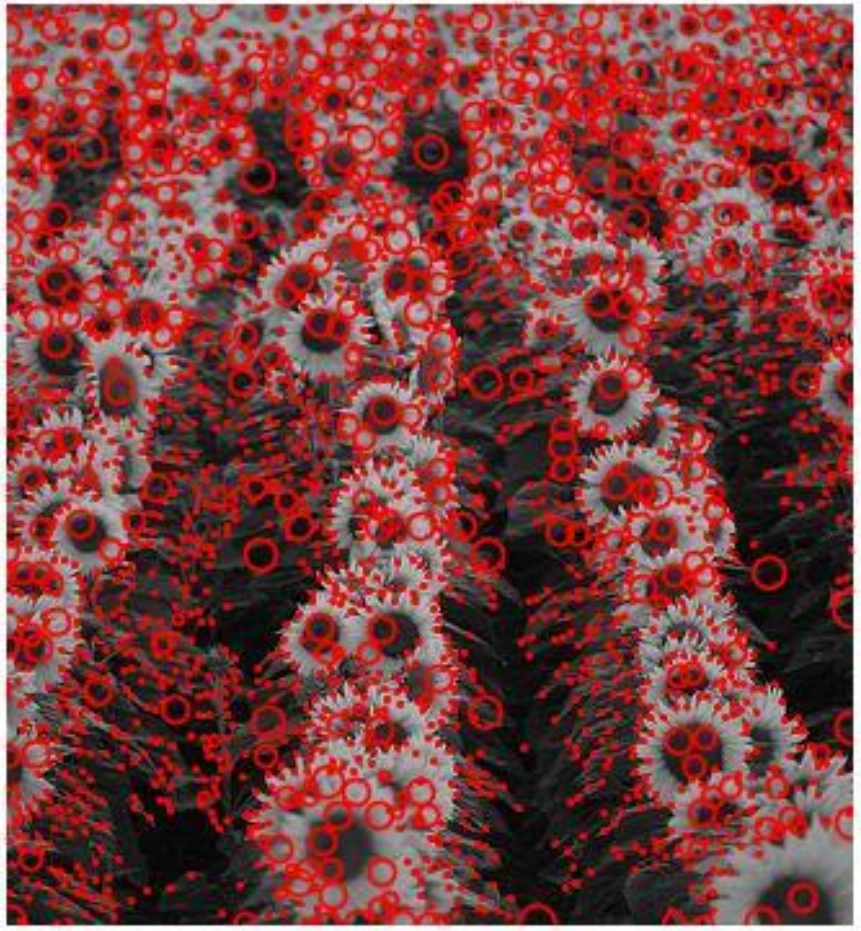
Increasing Kernel Size:
Running Time = 3.07s

Downsizing Image:
Running Time = 0.18s

2431 circles



2286 circles



Increasing Kernel Size:
Running Time = 2.11s

Downsizing Image:
Running Time = 0.13s

Comparison of running times

- As we can see from the running times stated that downsizing an image and then filtering it is much faster than increasing the kernel size.
- Therefore downsizing an image is more Downsizing Image.

Parameter choosing

- I played around with the values of sigma, threshold, K(multiplication factor) and number of levels in the code.
- I found sigma = 0.5, K = √2 and number of levels = 15 to give the best results for me in both the methods.
- Threshold = 0.01 gave me the best results in the Increasing kernel size method.
- Threshold = 0.03 gave me the best results in the Downsizing method.

- I have tried sigma = 1, threshold = 0.1, K =2 and No of levels = 10, but the above values gave me the best results.
- All the values were chosen based on the results I got by trial and error with multiple values.