3 Blob Detection Results

The results of applying the blob detector on two sets of images is presented in this section. The first set is the testing images given by the project description. The second set of images are chosen by the student to demonstrate the effectiveness and performance of the code. More details on computation time and parameters used are provided in table 1.

Four test images: These images include 'butterfly', 'Einstein', 'fishes', and 'sunflowers' images. The results are depicted in figures 4-7.

Student Images: Six images are tested in this section. Images are selected to cover both images of natural objects and scenes and the synthetic images created by computer graphics programs. These images include: 'rocks', 'Hunt' (library), 'ECE' (logo), 'circles' (synthetic), 'Hunt-2' (library), and 'solar system'. The results are depicted in figures 8-13.

3.1 Experiment

The blob detector is applied on each image with six different configurations. The parameters for each blob detector is as follows:

1. The initial scale (σ) = 2, Scale size = 10, threshold = 0.01, Filter type = DoG.
2. The initial scale (σ) = 2, Scale size = 15, threshold = 0.01, Filter type = DoG.
3. The initial scale (σ) = 2, Scale size = 10, threshold = 0.01, Filter type = LoG.
4. The initial scale (σ) = 2, Scale size = 15, threshold = 0.01, Filter type = LoG.
5. The initial scale (σ) = 5, Scale size = 10, threshold = 0.01, Filter type = DoG.
6. The initial scale (σ) = 5, Scale size = 10, threshold = 0.01, Filter type = LoG.

Note these values for initial scale and threshold are selected using try and error.

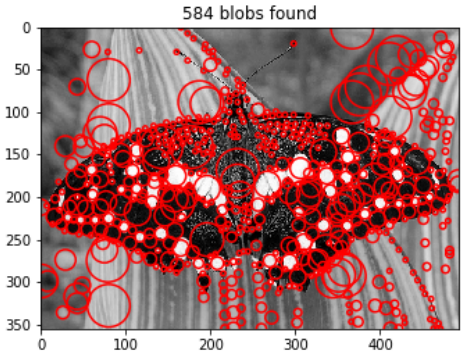
3.2 Example Command

In order to run the blob detector one need to have the 'Numpy' and 'Pillow' libraries installed. If they are not already installed they can be easily installed using the following command:

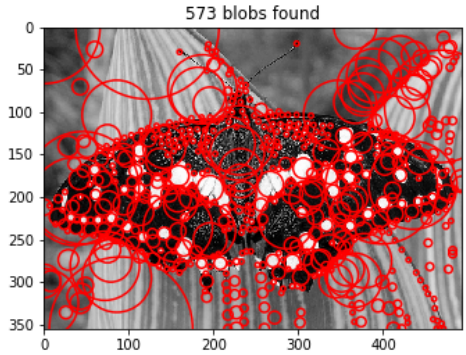
```
pip install pillow,   pip install Numpy
```

After installing the above libraries, one should run the 'afallah_blob.py' file. Then in order to apply it on an image the following command should be used:

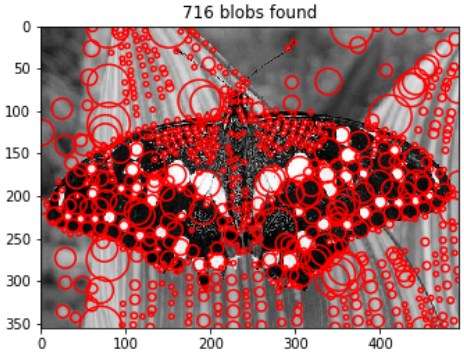
```
blob_detector('butterfly.jpg', 2, 10, 0.01, 'DOG', 'buttefly_blob')
```



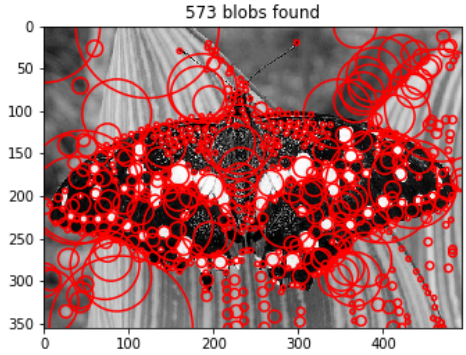
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



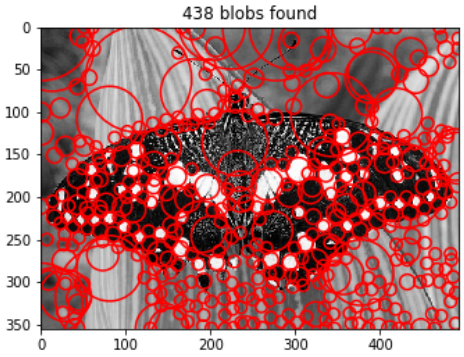
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



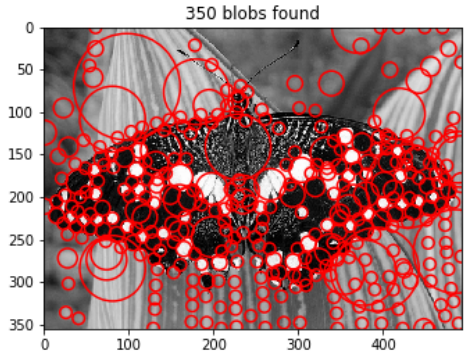
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



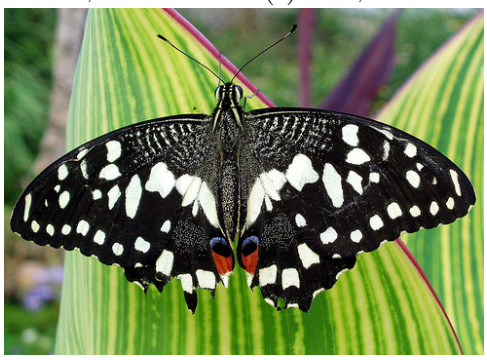
(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG

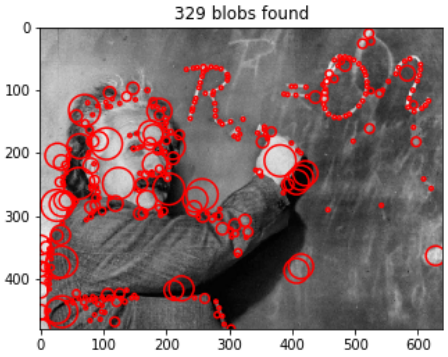


(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

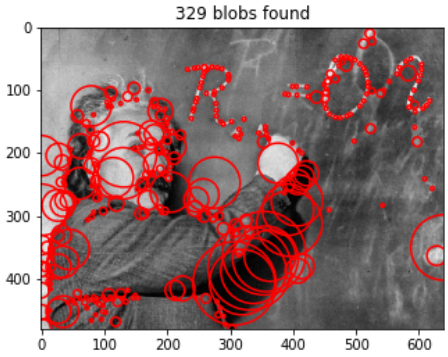


(g) Original Image

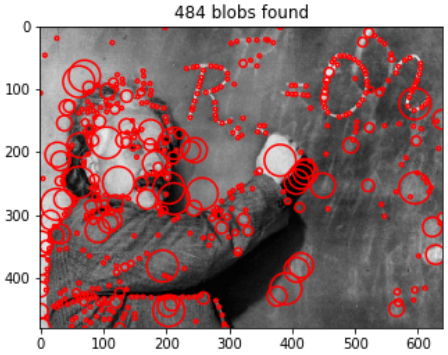
Figure 4: Test Image: 'butterfly'.



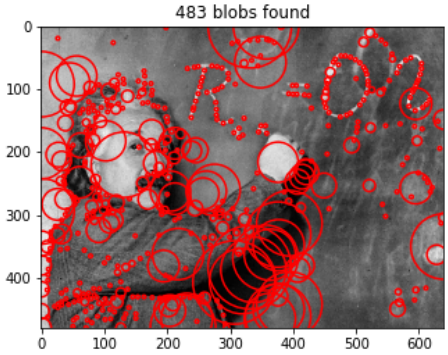
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



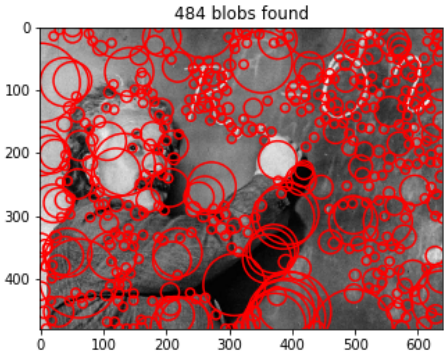
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



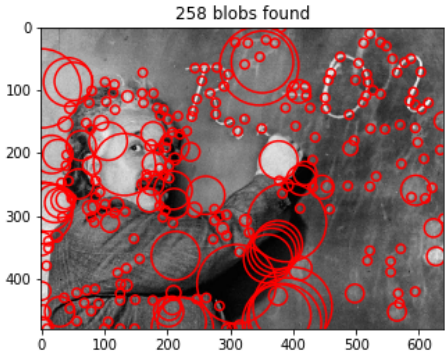
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



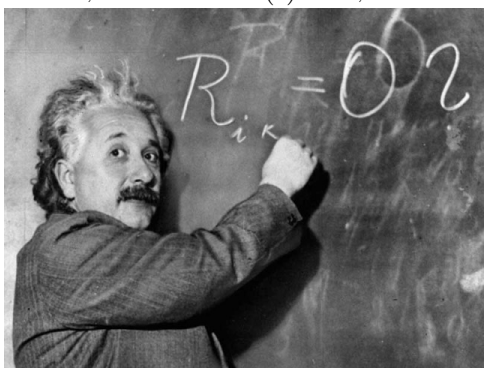
(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG

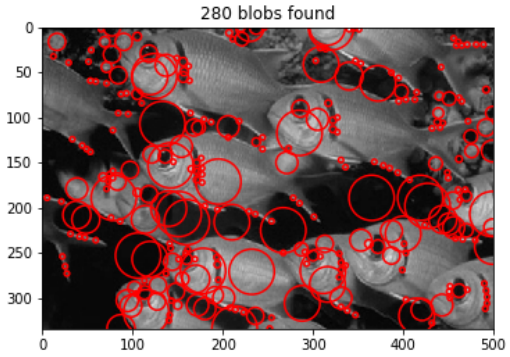


(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

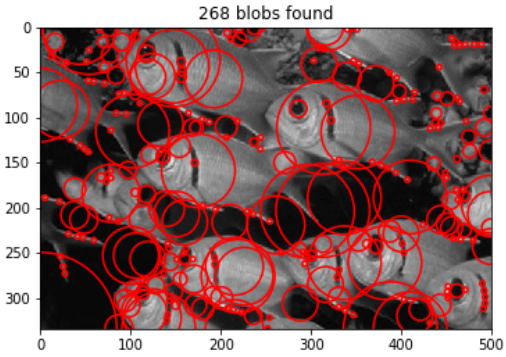


(g) Original Image

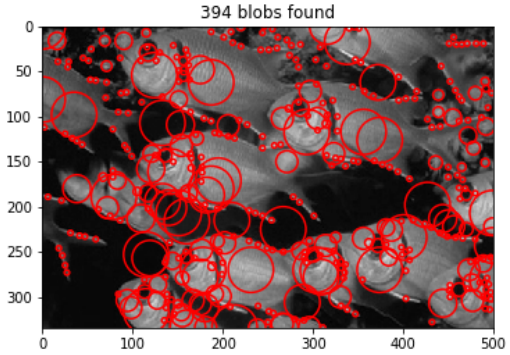
Figure 5: Test Image: 'Einstein'.



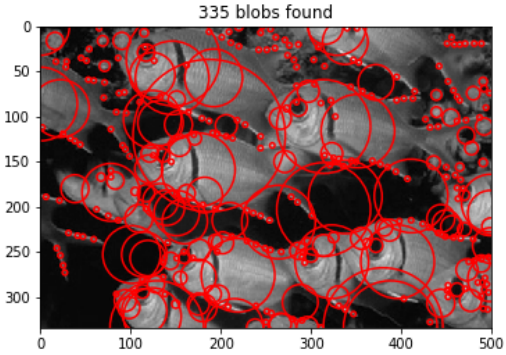
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



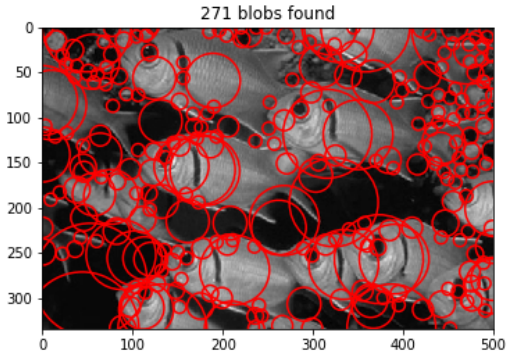
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



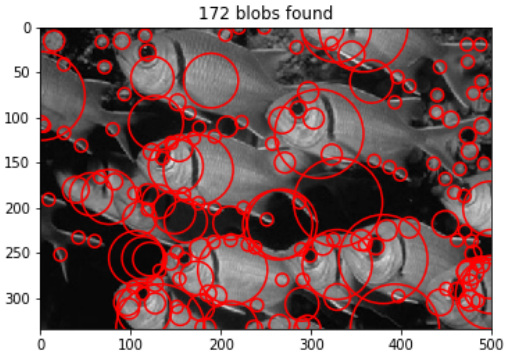
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



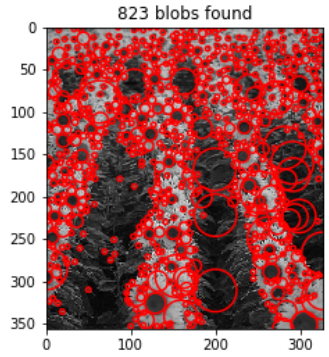
(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG



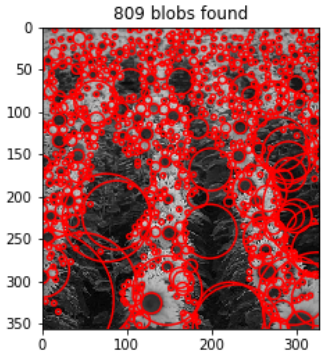
(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

(g) Original Image

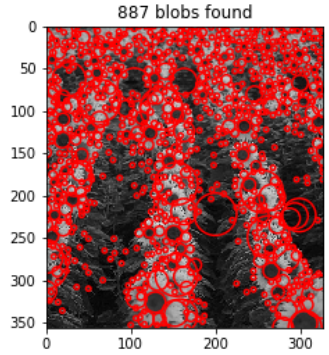
Figure 6: Test Image: 'fishes'.



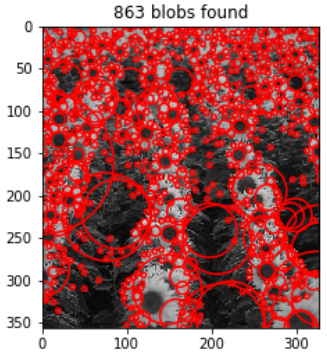
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



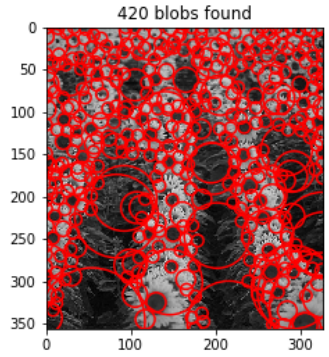
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



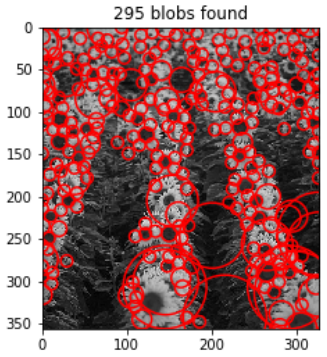
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



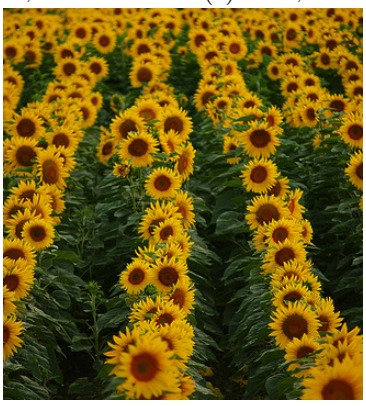
(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG

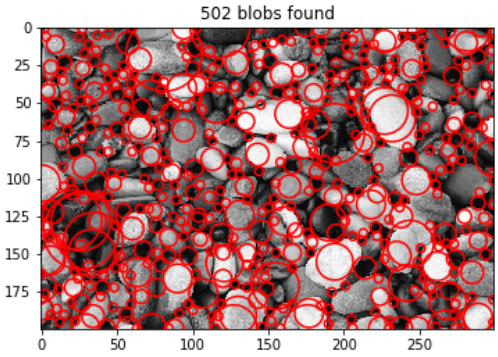


(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

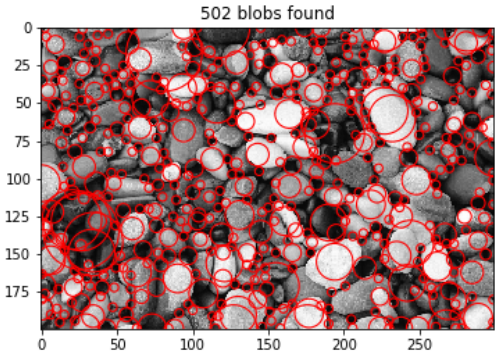


(g) Original Image

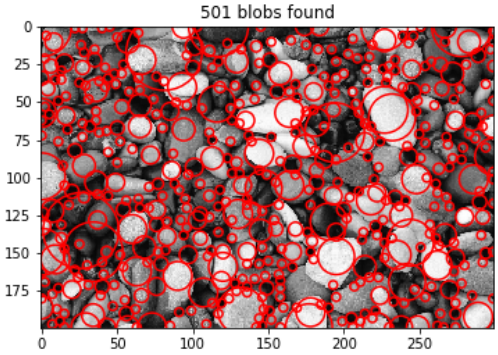
Figure 7: Test Image: 'sunflowers'.



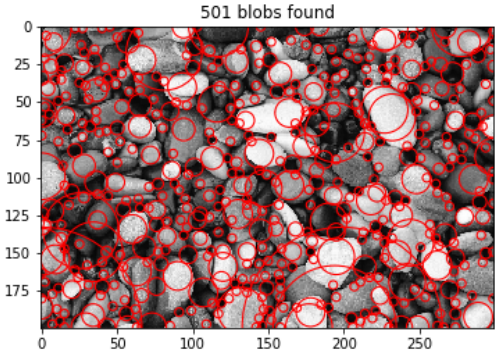
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



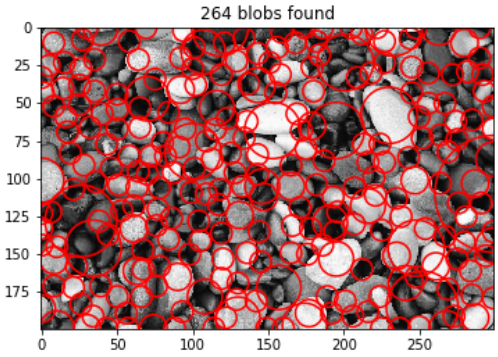
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



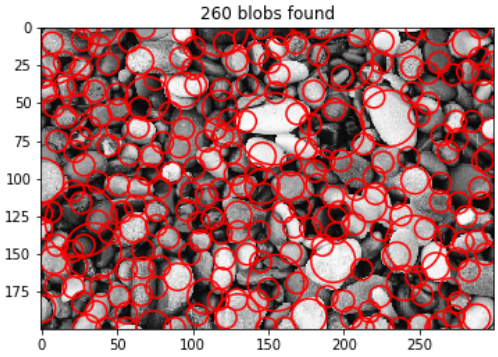
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



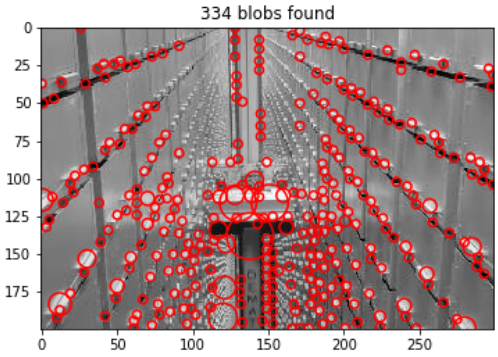
(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG



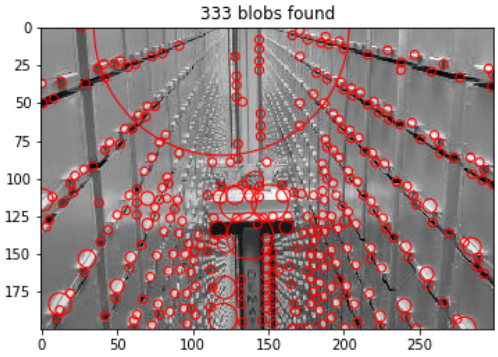
(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

(g) Original Image

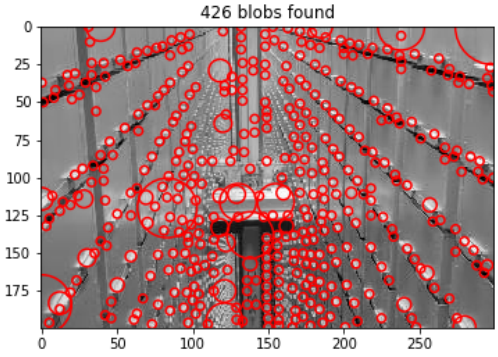
Figure 8: Student Image: 'rocks'.



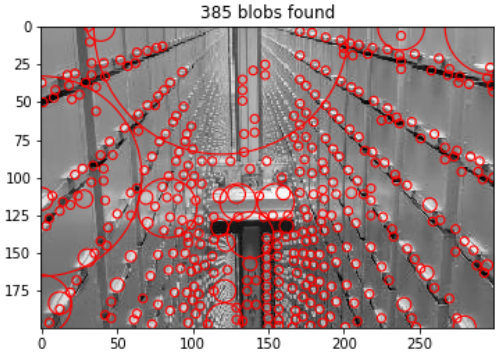
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



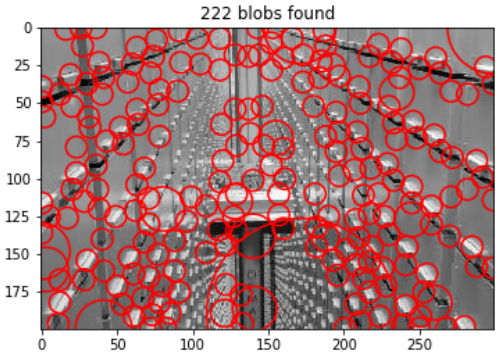
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



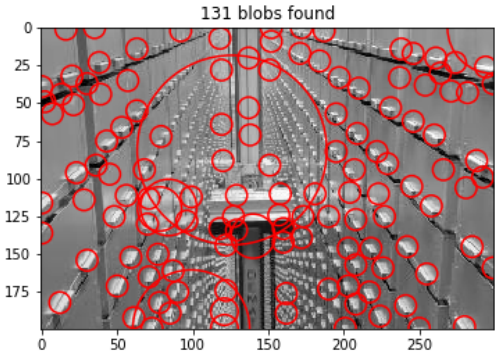
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



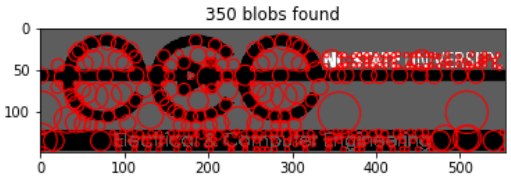
(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG



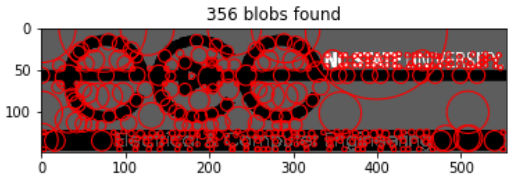
(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

(g) Original Image

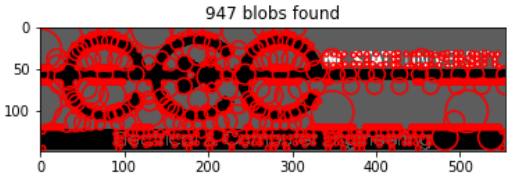
Figure 9: Student Image: 'Hunt'.



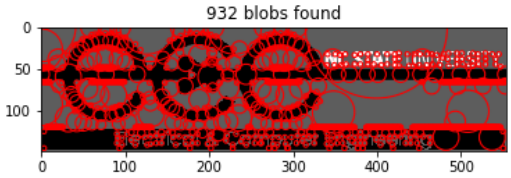
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



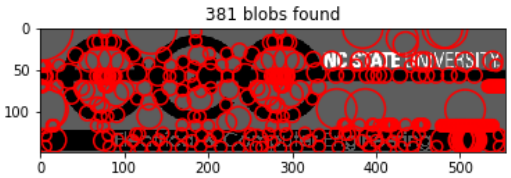
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



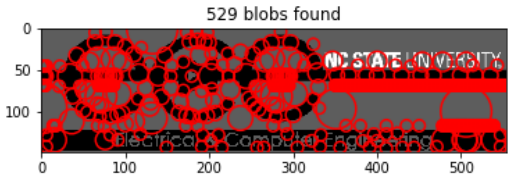
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



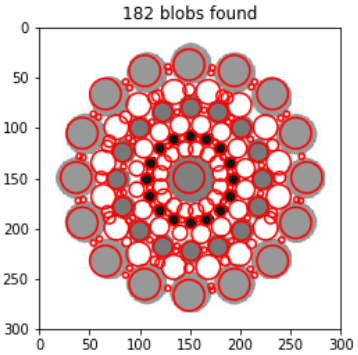
(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG



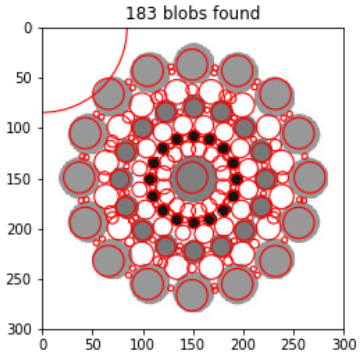
(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

(g) Original Image

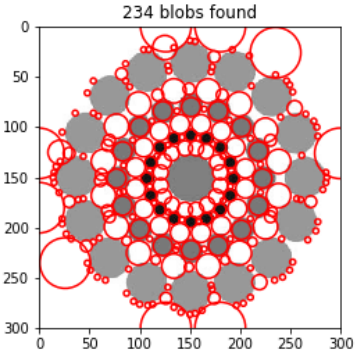
Figure 10: Student Image: 'ECE'.



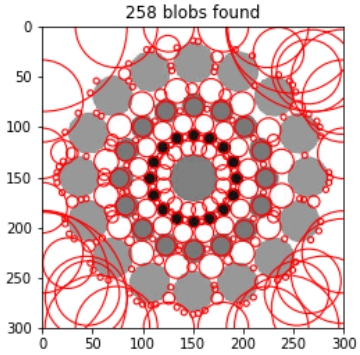
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



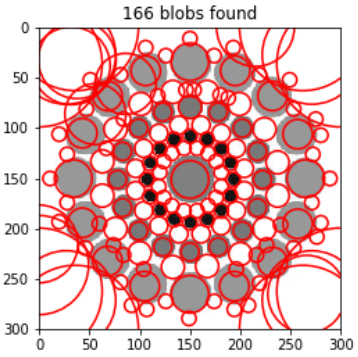
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



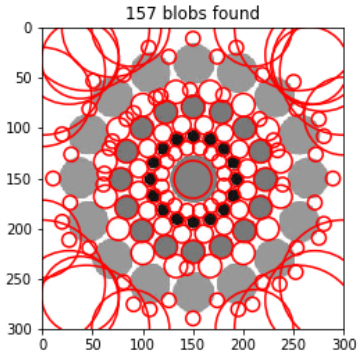
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



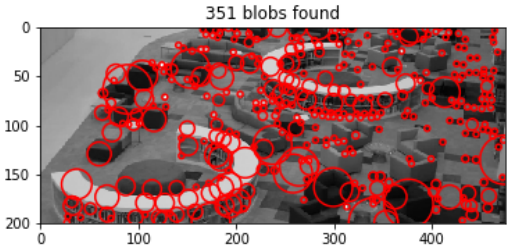
(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG



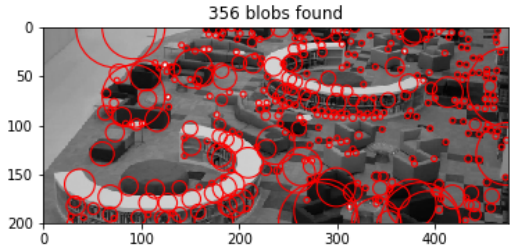
(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

(g) Original Image

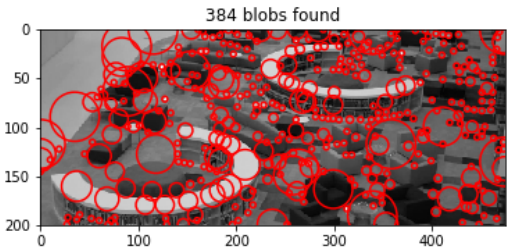
Figure 11: Student Image: 'circles'.



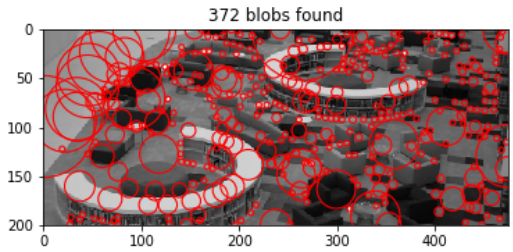
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



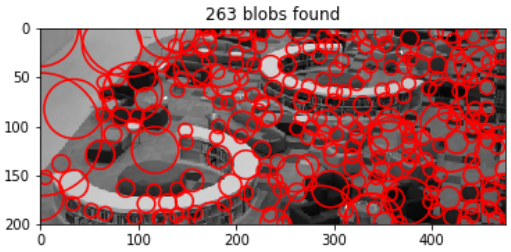
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



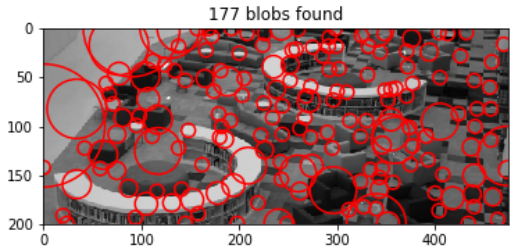
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



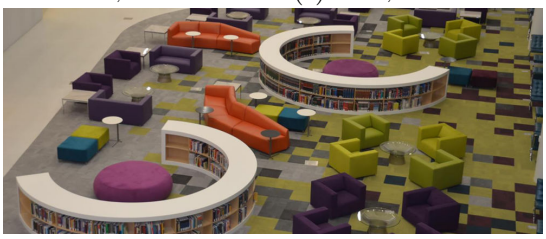
(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG

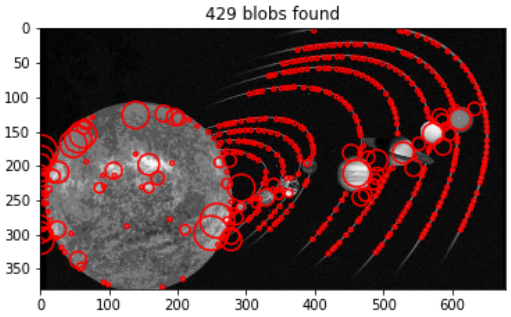


(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG

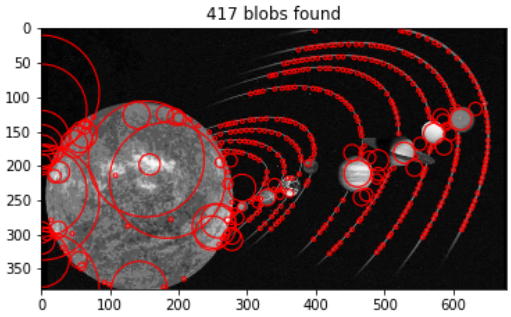


(g) Original Image

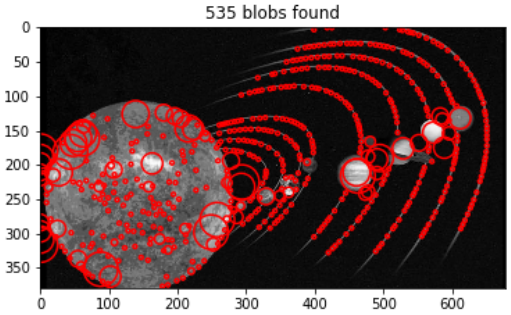
Figure 12: Student Image: 'Hunt-2'.



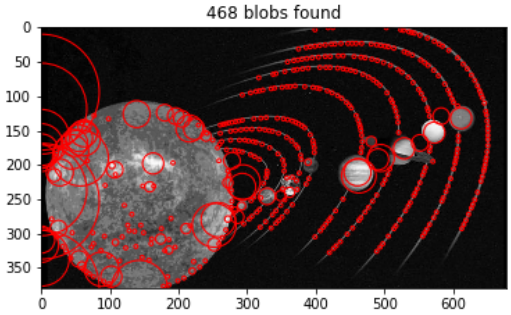
(a) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



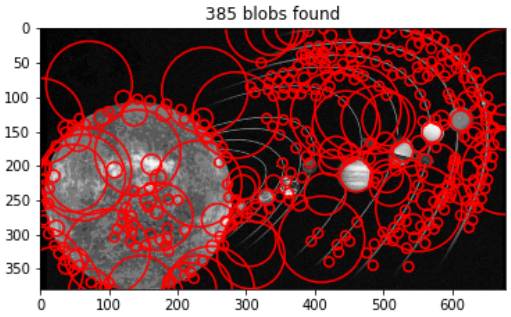
(b) σ= 2, threshold= 0.01, # of scales= 15, Filter= DoG



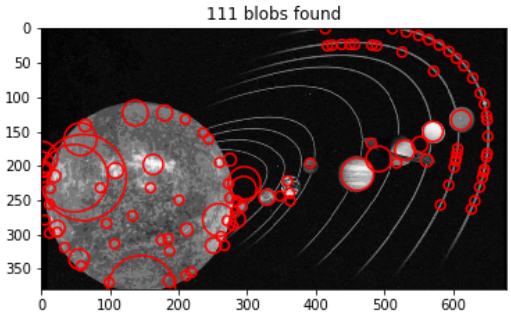
(c) σ= 2, threshold= 0.01, # of scales= 10, Filter= LoG



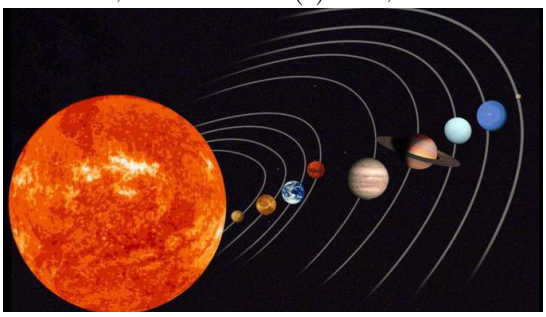
(d) σ= 2, threshold= 0.01, # of scales= 15, Filter= LoG



(e) σ= 5, threshold= 0.01, # of scales= 10, Filter= DoG



(f) σ= 5, threshold= 0.01, # of scales= 10, Filter= LoG



(g) Original Image

Figure 13: Student Image: 'solar system'.

Table 1: Blob detection runs and their results for Test Images.

run	image name	initial scale	scale size	threshold	Filter	time	blobs found
1	butterfly	2	10	0.01	DoG	43.7463	584
2	butterfly	2	15	0.01	DoG	61.4173	573
3	butterfly	2	10	0.01	LoG	43.9239	716
4	butterfly	2	15	0.01	LoG	61.1940	573
5	butterfly	5	10	0.01	DoG	49.2448	483
6	butterfly	5	10	0.01	LoG	49.2765	350
7	einstein	2	10	0.01	DoG	48.1930	329
8	einstein	2	15	0.01	DoG	106.6267	329
9	einstein	2	10	0.01	LoG	48.9386	484
10	einstein	2	15	0.01	LoG	110.7511	483
11	einstein	5	10	0.01	DoG	58.4803	484
12	einstein	5	10	0.01	LoG	58.5772	258
13	fishes	2	10	0.01	DoG	26.0474	280
14	fishes	2	15	0.01	DoG	57.7511	268
15	fishes	2	10	0.01	LoG	26.4342	394
16	fishes	2	15	0.01	LoG	58.0712	335
17	fishes	5	10	0.01	DoG	31.3585	271
18	fishes	5	10	0.01	LoG	31.6313	172
19	sunflowers	2	10	0.01	DoG	19.1962	280
20	sunflowers	2	15	0.01	DoG	41.3874	268
21	sunflowers	2	10	0.01	LoG	19.1193	394
22	sunflowers	2	15	0.01	LoG	42.4214	335
23	sunflowers	5	10	0.01	DoG	22.3885	271
24	sunflowers	5	10	0.01	LoG	22.1346	172

Table 2: Blob detection runs and their results for Student Images.

run	image name	initial scale	scale size	threshold	Filter	time	blobs found
1	rocks	2	10	0.01	DoG	9.8595	502
2	rocks	2	15	0.01	DoG	21.1215	502
3	rocks	2	10	0.01	LoG	9.9684	501
4	rocks	2	15	0.01	LoG	21.1365	501
5	rocks	5	10	0.01	DoG	11.7534	264
6	rocks	5	10	0.01	LoG	11.6995	260
7	hunt	2	10	0.01	DoG	9.6626	334
8	hunt	2	15	0.01	DoG	21.2580	333
9	hunt	2	10	0.01	LoG	9.8065	426
10	hunt	2	15	0.01	LoG	21.0052	385
11	hunt	5	10	0.01	DoG	11.8004	222
12	hunt	5	10	0.01	LoG	11.5975	131
13	ece	2	10	0.01	DoG	13.1386	350
14	ece	2	15	0.01	DoG	28.3683	356
15	ece	2	10	0.01	LoG	13.8723	947
16	ece	2	15	0.01	LoG	29.3962	932
17	ece	5	10	0.01	DoG	16.1879	381
18	ece	5	10	0.01	LoG	31.6313	529
19	circles	2	10	0.01	DoG	13.9422	182
20	circles	2	15	0.01	DoG	30.6848	183
21	circles	2	10	0.01	LoG	14.1321	234
22	circles	2	15	0.01	LoG	31.0029	258
23	circles	5	10	0.01	DoG	17.1214	166
24	circles	5	10	0.01	LoG	17.0934	157
25	hunt2	2	10	0.01	DoG	15.1515	351
26	hunt2	2	15	0.01	DoG	32.7918	356
27	hunt2	2	10	0.01	LoG	15.1435	384
28	hunt2	2	15	0.01	LoG	32.7505	372
29	hunt2	5	10	0.01	DoG	18.0589	263
30	hunt2	5	10	0.01	LoG	17.9040	177
31	solar	2	10	0.01	DoG	41.3488	429
32	solar	2	15	0.01	DoG	89.5480	417
33	solar	2	10	0.01	LoG	41.1300	735
34	solar	2	15	0.01	LoG	90.9282	468
35	solar	5	10	0.01	DoG	48.5028	385
36	solar	5	10	0.01	LoG	46.5999	111

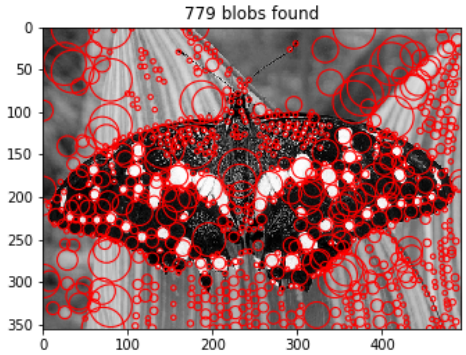
4 Analysis

From the conducted experiments we can observe that the blob detector is capable of detecting blobs correctly. Also, we can observe that the DoG filter can closely approximate the LoG filter since the results of using either one is close to each other. Moreover, we observe that using DoG filter can reduce the computation time in most cases. Another point is the scale size (number of scales to iterate in scale-space) increasing this number will generally increase the number of blobs detected and some larger details in the image will be detected as a result. The other factor that we studied is the initial scale (σ). A good initial scale value turned out to be equal to 2, However, increasing initial scale value will generally result in detecting larger details in the image and larger blobs are returned from the detector.

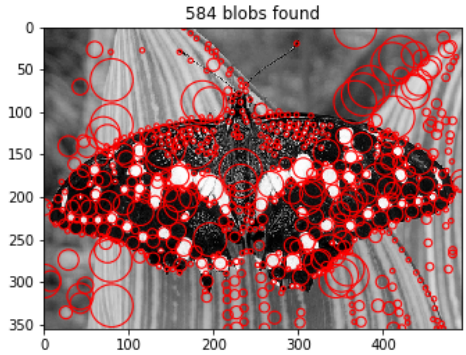
In order to study the effect of threshold on blob detection, another experiment is done on the 'butterfly' image. Same blob detector is applied on this image for different threshold values ([0.005, 0.01, 0.025, 0.05, 0.1, 0.5]). The results are depicted in figure 14. We can observe that increasing the threshold will decrease the number of blobs detected and only retain the ones with very high response to the filter. These blobs will be corresponding to most important ones of the image. As we increase the threshold value we can see that there is no detection when threshold = 0.5.

5 Conclusions

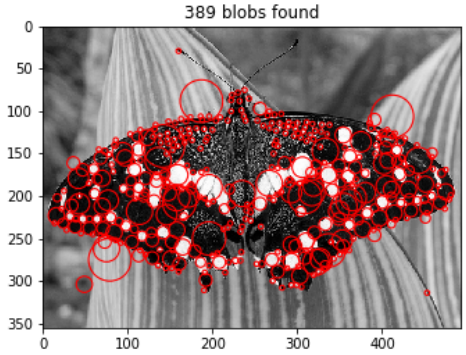
In this project we implemented blob detector in Python without using the in-built function. In order to create the blob detector, we defined multiple functions to handle the tasks of convolution, scale space creation, interpolation and image re-sizing, non-maximum suppression, etc. We applied the developed blob detector and tested that on the given test images as well as student provided images. Analysis of the results and comparison between different configurations are given.



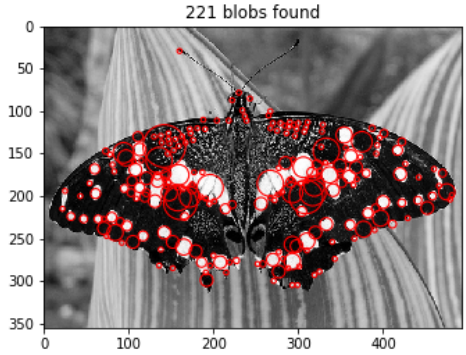
(a) σ= 2, threshold= 0.005, # of scales= 10, Filter= DoG



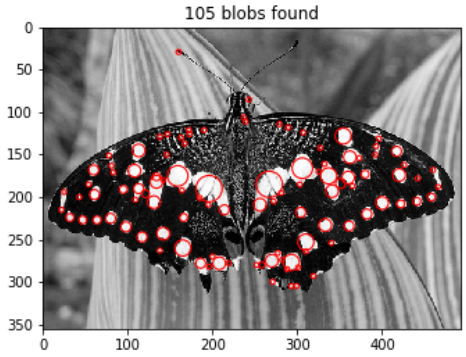
(b) σ= 2, threshold= 0.01, # of scales= 10, Filter= DoG



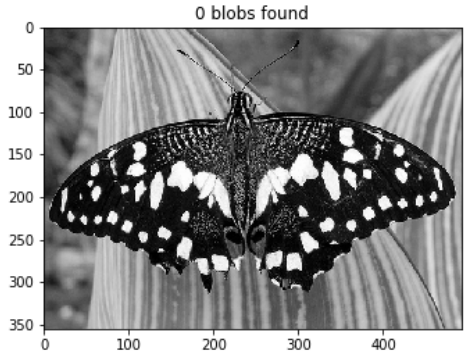
(c) σ= 2, threshold= 0.025, # of scales= 10, Filter= DoG



(d) σ= 2, threshold= 0.05, # of scales= 10, Filter= DoG



(e) σ= 2, threshold= 0.1, # of scales= 10, Filter= DoG



(f) σ= 2, threshold= 0.5, # of scales= 10, Filter= DoG

Figure 14: Effect of different threshold values on blob detector.