CS543 Assignment 2

Your Name: Zong Fan
Your NetId: zongfan2

Part 1 Hybrid Images:

You will provide the following for ***3 different examples*** (1 provided pair, 2 pairs of your own):

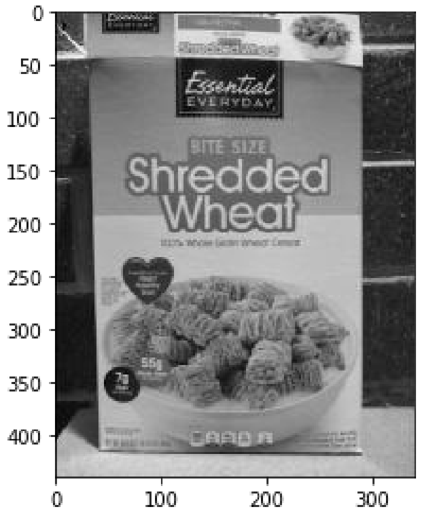
- two input images
- two *filtered* input images
- two generated hybrid image (at different resolutions, similar to images C and D above)
- two σ values (one for each filter)

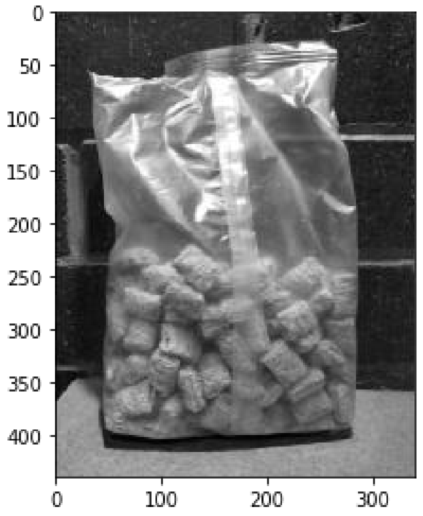
You will provide the following as further discussion overall:

- Explanation of how you chose the σ values
- Discussion of how successful your examples are + any interesting observations

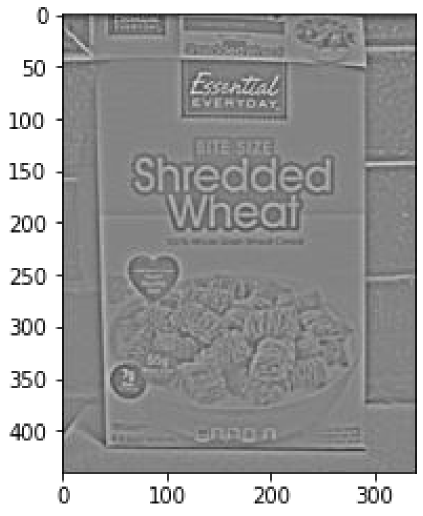
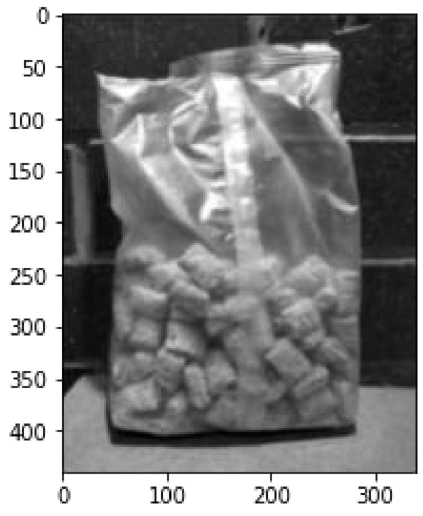
Example 1:

Original images

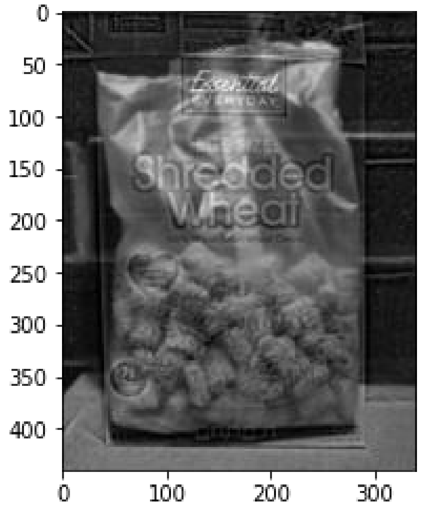




Filtered images:

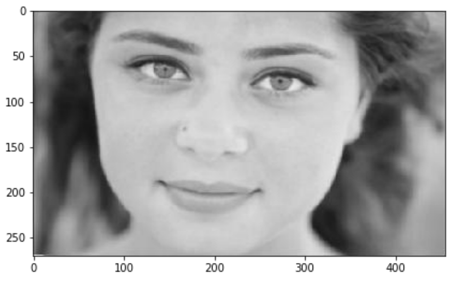
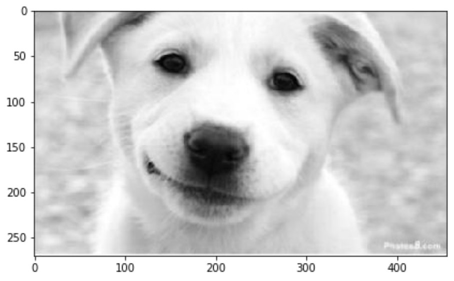
Hybrid images:



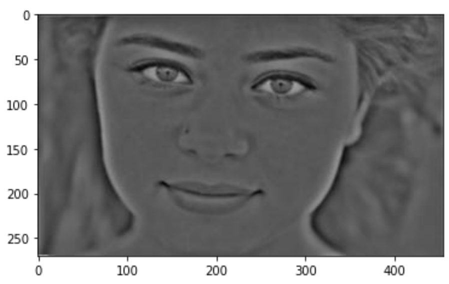
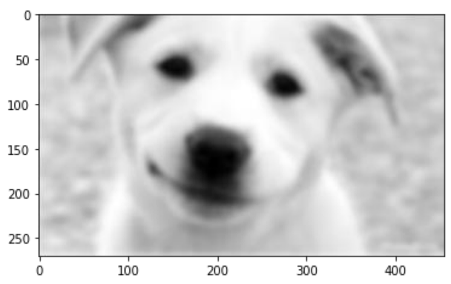
Sigma: image 1 is 3; image 2 is 1.

Example 2:

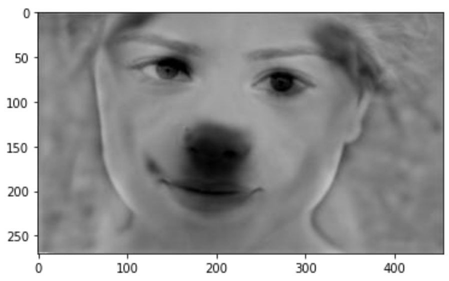
Original images

Filtered images:

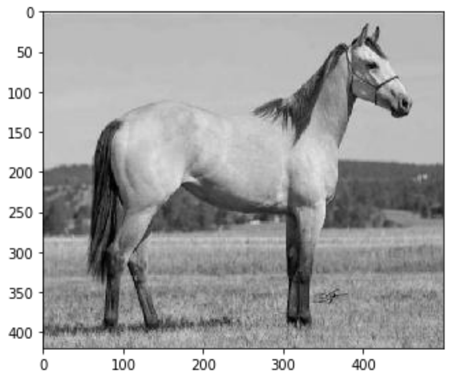
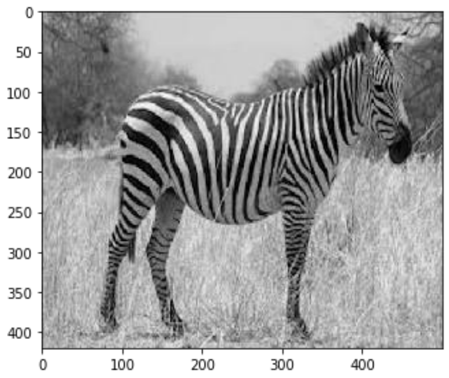
Hybrid images:



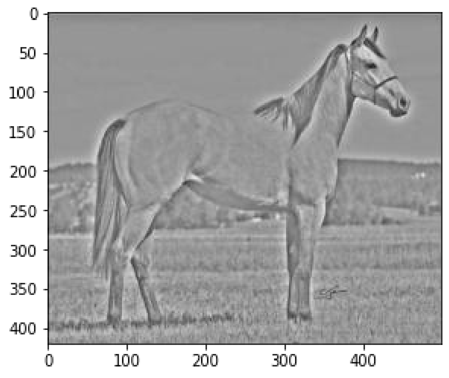
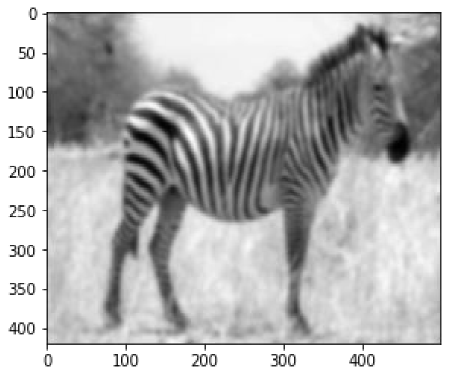
Sigma: image 1 is 7; image 2 is 3.

Example 3:

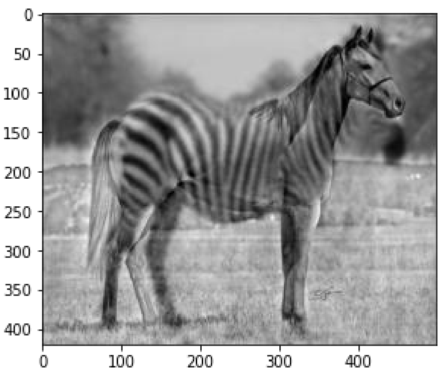
Original images

Filtered images:

Hybrid images:



Sigma: image 1 is 7; image 2 is 3.

Discussion:

1. For high-pass filter, increasing σ makes the edges more obvious in the filtered image. For low-pass filter, increasing σ makes the edges more obscure in the filtered image. When choosing σ, for image 1, larger sigma is desired for high-pass filter to remove most texture information. For image 2, smaller sigma is desired for low-pass filter to keep the texture information while blurring the edges. For detailed values, adjust σ values for relatively clear scaffold image of the first image and blurry images of the second image.
2. When the size of hybrid image is large, the hybrid effect is visually-best that it looks like an chimera with the object contour from one figure and the filling texture from another. However, interestingly, when the hybrid image is down-sampled to 1/4 to 1/5 of original size, it just looks like the original image used for low-pass filtering. This is perhaps because human's visual perception system is sensitive to object edges and contours when the hybrid image is large and contains enough details. These information is high-pass feature. But when the image is small which makes the edge information invisible, human might classify the object just via the overall texture which is low-pass.

Part 2 Scale-Space Blob Detection:

You will provide the following for ***8 different examples*** (4 provided, 4 of your own):

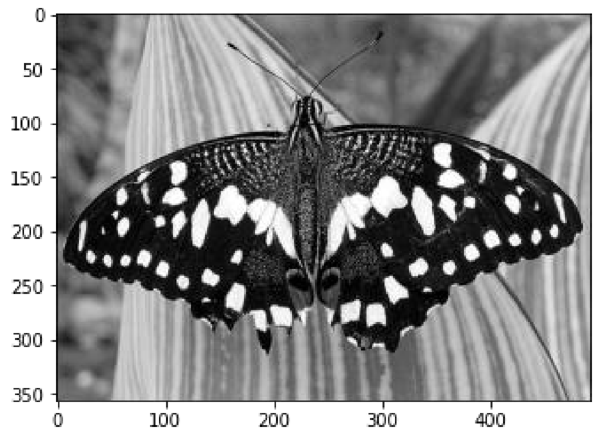
- original image
- output of your circle detector on the image
- running time for the "efficient" implementation on this image
- running time for the "inefficient" implementation on this image

You will provide the following as further discussion overall:

- Explanation of any "interesting" implementation choices that you made.
- Discussion of optimal parameter values or ones you have tried

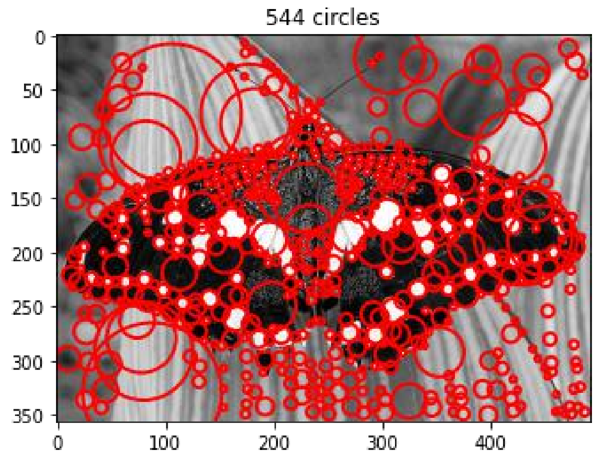
Example 1:

Original image:



Circle detector on image:

Threshold = 0.005

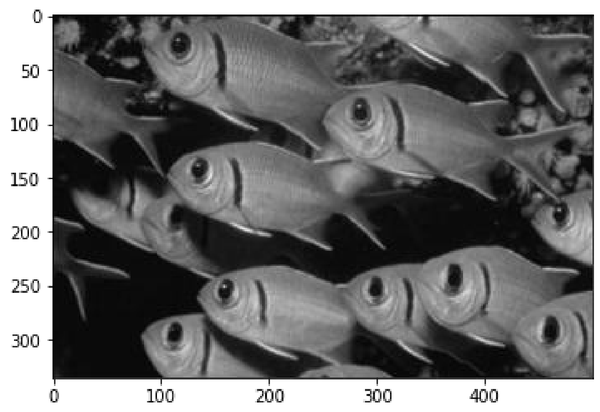


Running time of down-sampling image size method: 0.179s/10 levels

Running time of increasing kernel size method: 0.525s/10 levels

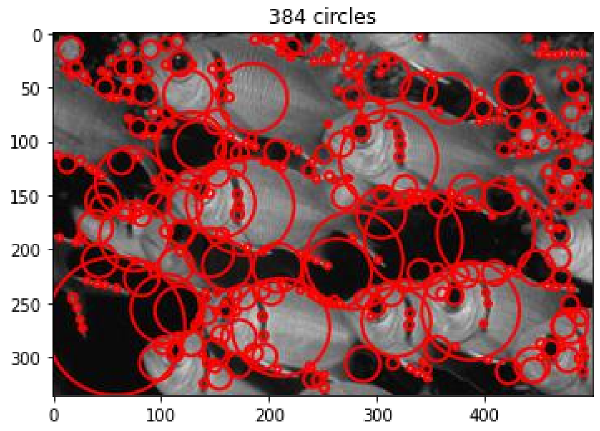
Example 2:

Original image:



Circle detector on image:

Threshold = 0.005

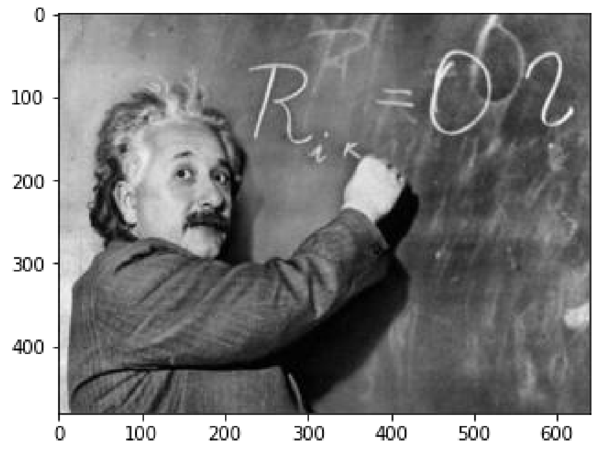


Running time of down-sampling image size method: 0.173s/10 levels

Running time of increasing kernel size method: 0.502s/10 levels

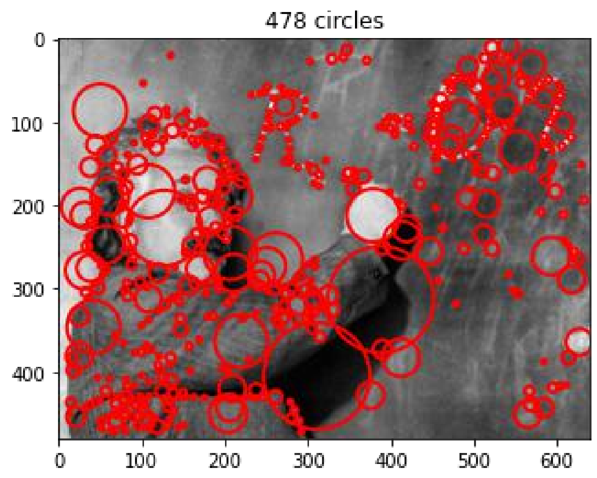
Example 3:

Original image:



Circle detector on image:

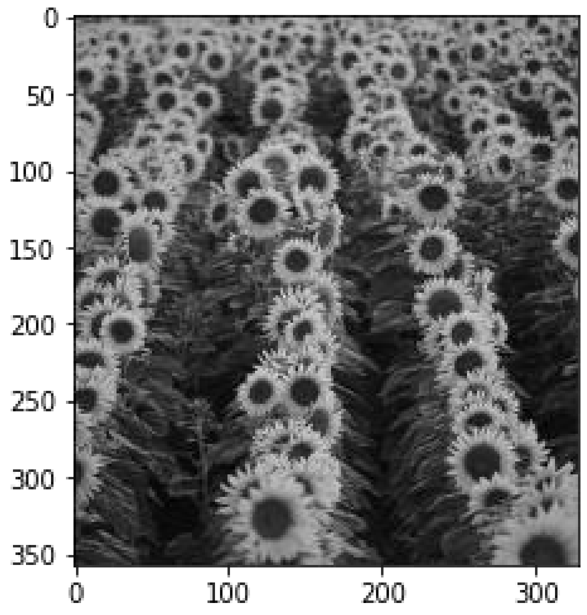
Threshold = 0.005



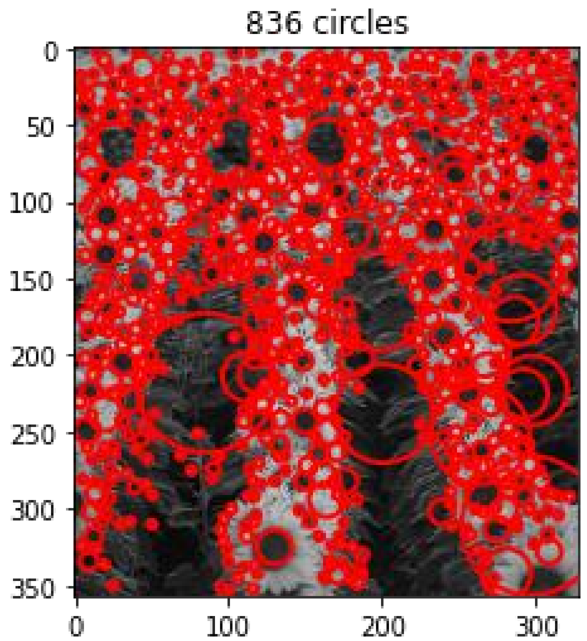
Running time of down-sampling image size method: 0.179s/10 levels
Running time of increasing kernel size method: 0.921s/10 levels

Example 4:

Original image:



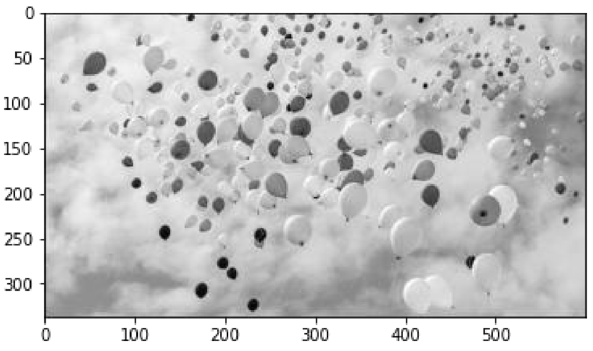
Circle detector on image:
Threshold = 0.005



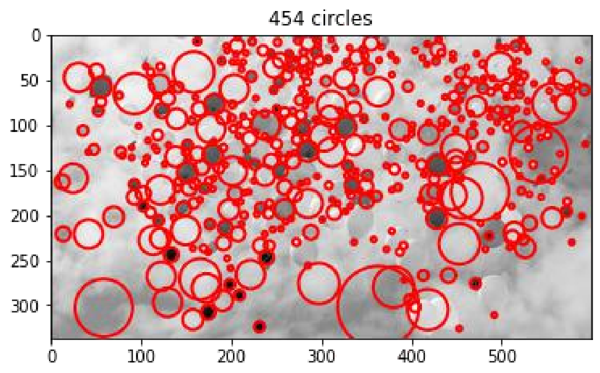
Running time of down-sampling image size method: 0.131s/10 levels
Running time of increasing kernel size method: 0.356s/10 levels

Example 5:

Original image:



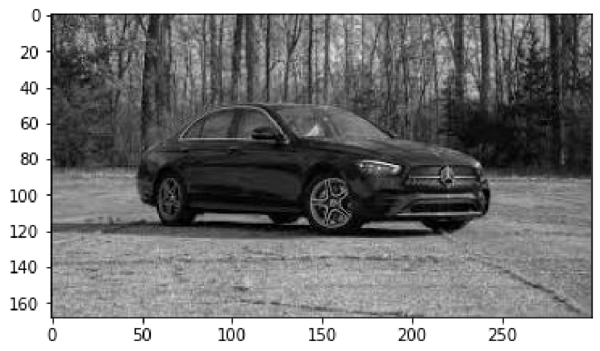
Circle detector on image:
Threshold = 0.005



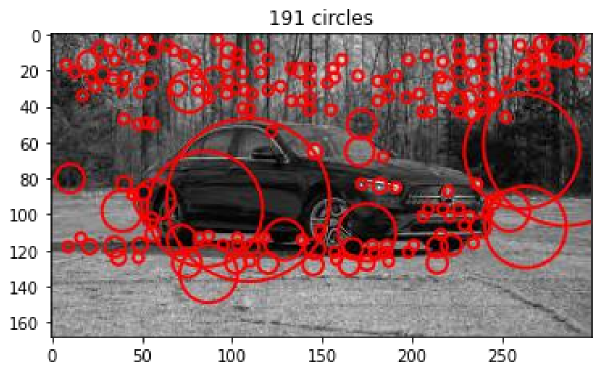
Running time of down-sampling image size method: 0.196s/10 levels
Running time of increasing kernel size method: 0.595s/10 levels

Example 6:

Original image:



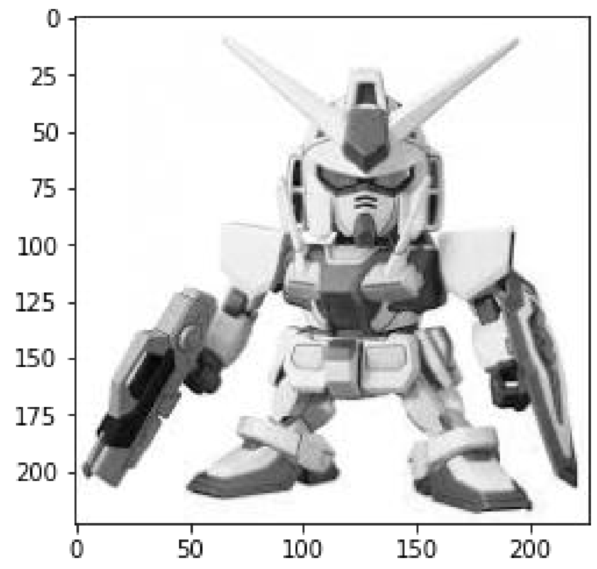
Circle detector on image:
Threshold = 0.005



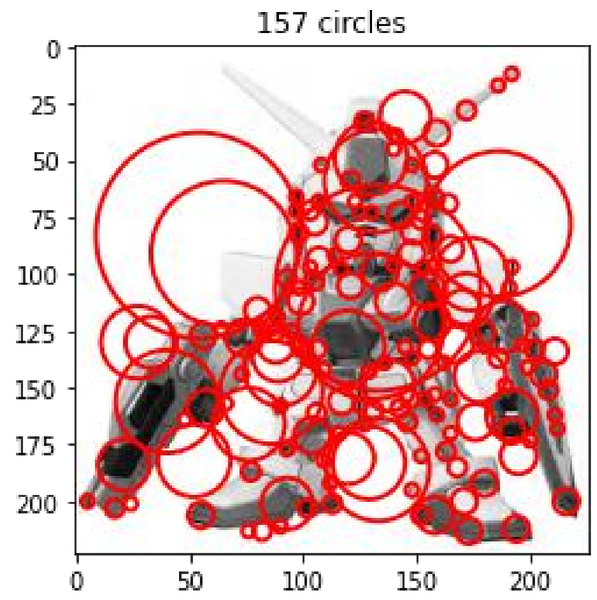
Running time of down-sampling image size method: 0.075s/10 levels
Running time of increasing kernel size method: 0.162s/10 levels

Example 7:

Original image:



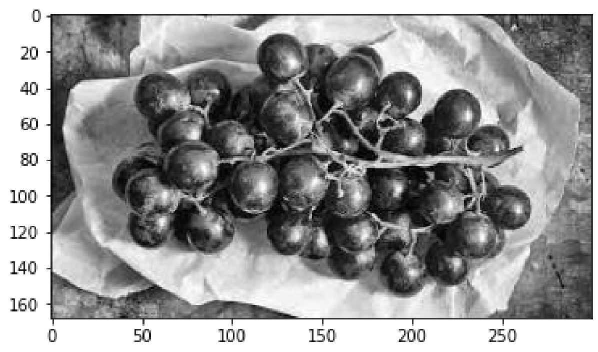
Circle detector on image:
Threshold = 0.005



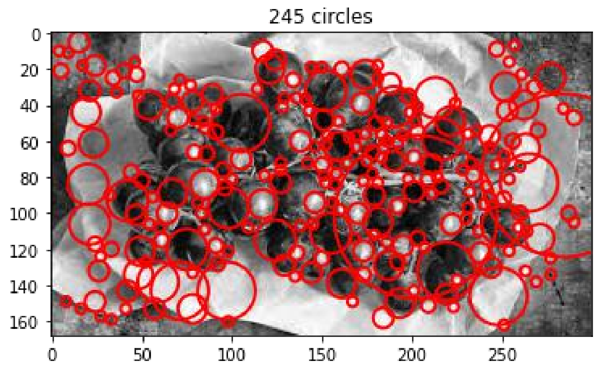
Running time of down-sampling image size method: 0.075s/10 levels
Running time of increasing kernel size method: 0.160s/10 levels

Example 8:

Original image:



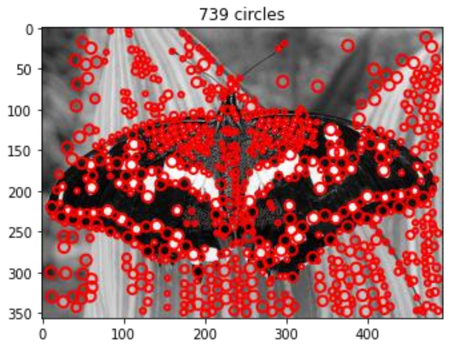
Circle detector on image:
Threshold = 0.005



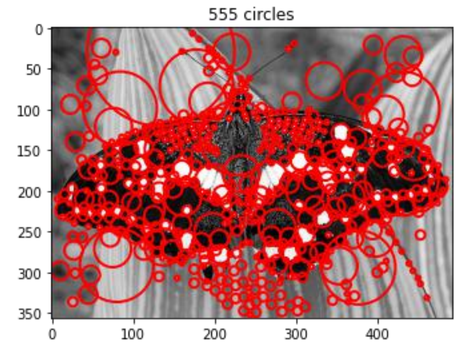
Running time of down-sampling image size method: 0.063s/10 levels
Running time of increasing kernel size method: 0.150s/10 levels

Discussion:

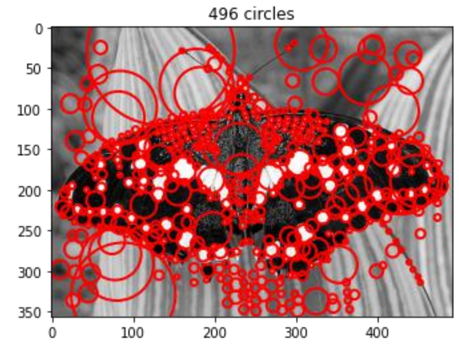
According to the document of scipy.ndimage.filters.gaussian_laplace, the kernel size of Laplacian filter is determined by parameter truncate. The filter size should be odd. In this experiment, it was set to 9 to control the number of detected blobs.
There are other parameters that could affect the number of detected circles, including pyramid expansion ratio k, number of pyramid layers, NMS filtering kernel size, and Laplacian response threshold.
Reduce k from sqrt(2) to 1.1. More circles are detected but most of them are small, since the pyramid size shrinks less.



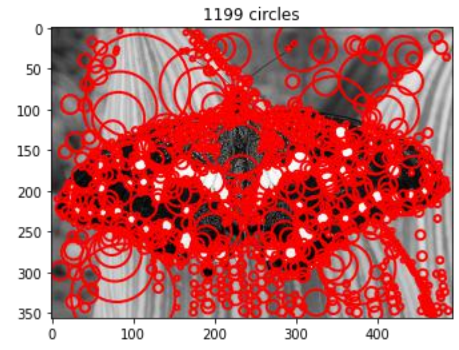
Increase k from sqrt(2) to 2. The number of large blobs seems to be increased, since the size of pyramid increases more sharply.



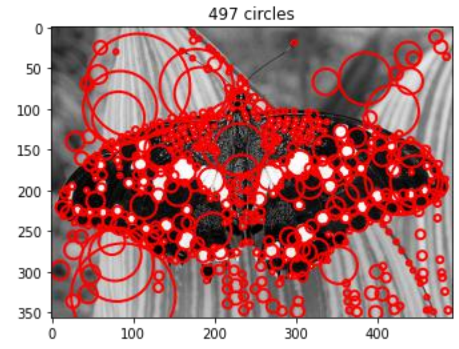
Increase levels from 10 to 12. On the contrary, the number of detected blobs increases trivially, since most of blobs are small which are detected on bottom layers.



Reduce NMS maximum filter kernel size from 7 to 3. Much more blobs are detected, since blobs could have larger overlaps.



Increase the response threshold from 0.005 to 0.01. Fewer blobs are detected.

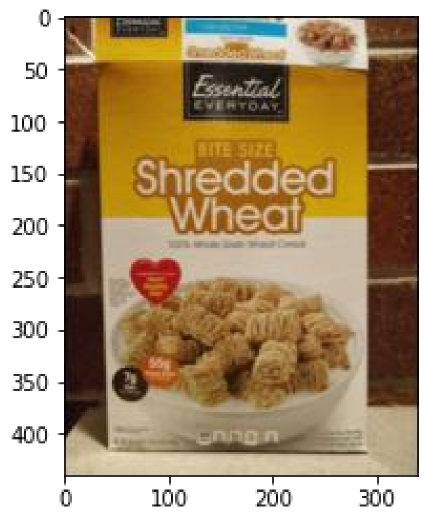


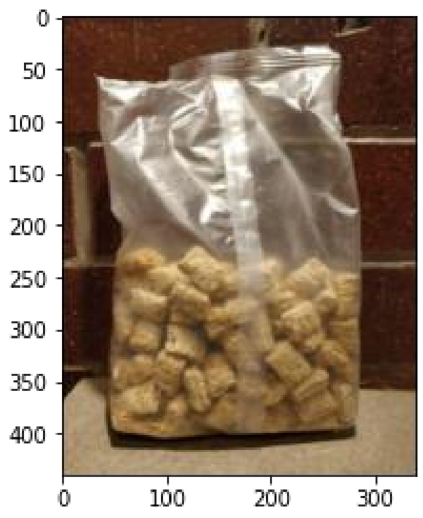
Bonus:

Hybrid Images Extra Credit

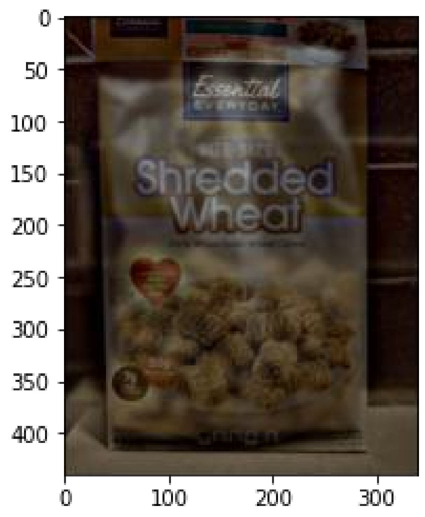
- Discussion and results of any extensions or bonus features you have implemented for Hybrid Images

Extra 1. Hybrid color image
Original image:



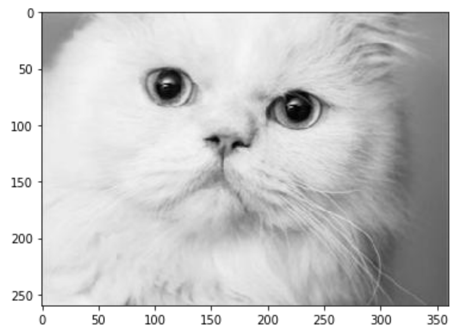


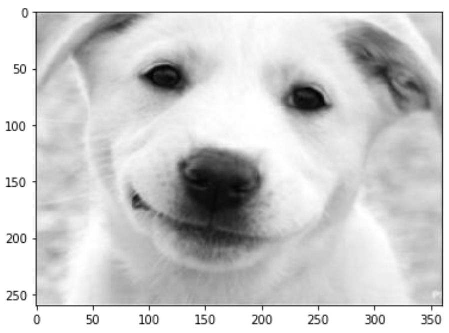
Hybrid image:



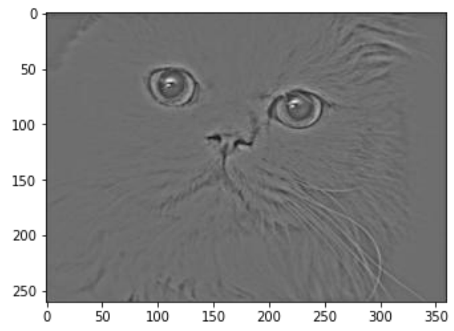
σ: image 1: 11; image 2: 3
The results show that the hybrid effects is also quite obvious in color images. But the overall brightness of color image decreases since the filtering operation.

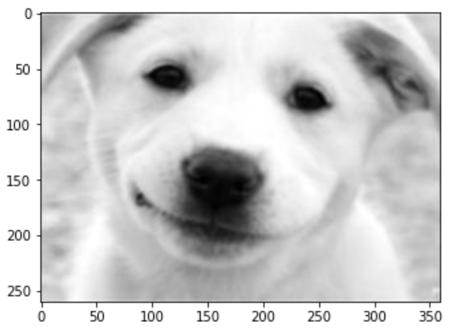
Extra 2: Failure of hybrid image
Original images:



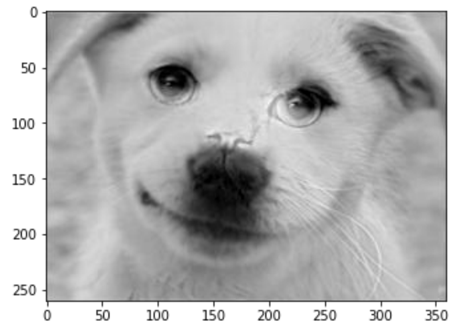


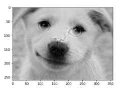
Filtered image:





Hybrid image:

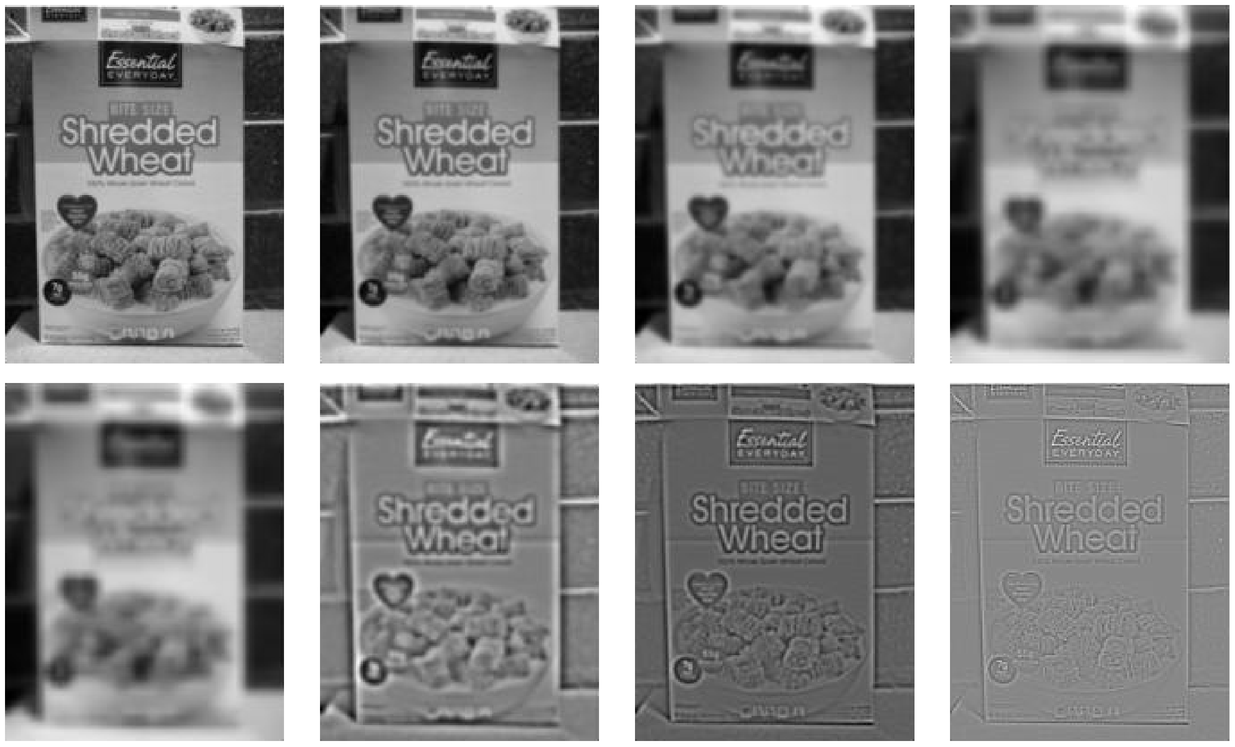




Sigma: image 1 is 7, image 2 is 3.

This hybrid image seems to be a failure case, since the hybrid image looks just like one image without obvious hybrid effect from another image.
This is probably due to two reasons: first, the two images are too similar in both texture and edges. If so, the low-pass information of the two images are visually similar. As we can see the filtered image of the cat, its edges are not so obvious and are overshadowed by the dog face. Second, some parts are not covering each other, such as the nose of the cat, making the hybrid image split. It’s hard to set threshold for the high-pass filter to make it disappear.

Extra 3: Gaussian and Laplacian pyramid



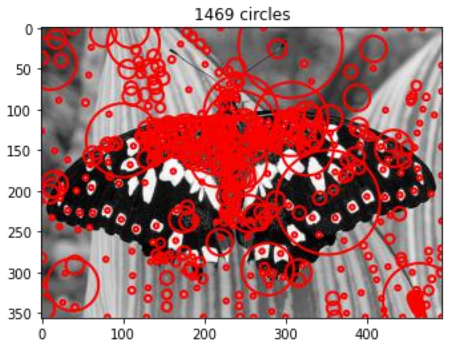
Gaussian pyramid is the downsample the image by half each iteration with Gaussian filter, while Laplacian pyramid is to upsample the blurred image and save the difference between each levels.

Blob-Detection Extra Credit

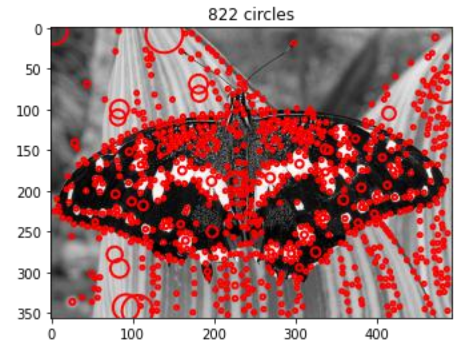
- Discussion and results of any extensions or bonus features you have implemented for Blob-Detection

Extra 1: Implement DoG pyramid.

First, use skimage.feature.blob_dog to get the DoG pyramid. Much more circles are detected compared to original method. The running time is 4.18s. Much slower since multiple octaves need computation.



Then replicate the method in the paper by setting the number of octave = 4. The running time is 0.197s. We found that the image most detected blobs are small. Perhaps increasing the number of octave could make it. Compared to scipy built-in function, the number of detected blobs differs a lot, perhaps due to the threshold setting


822 circles