

Report--Shidi Zhao(szhao41)

Part 1

1. The original, filtered, and hybrid images for all 3 examples (i.e., 1 provided and 2 pairs of your own).

1) original



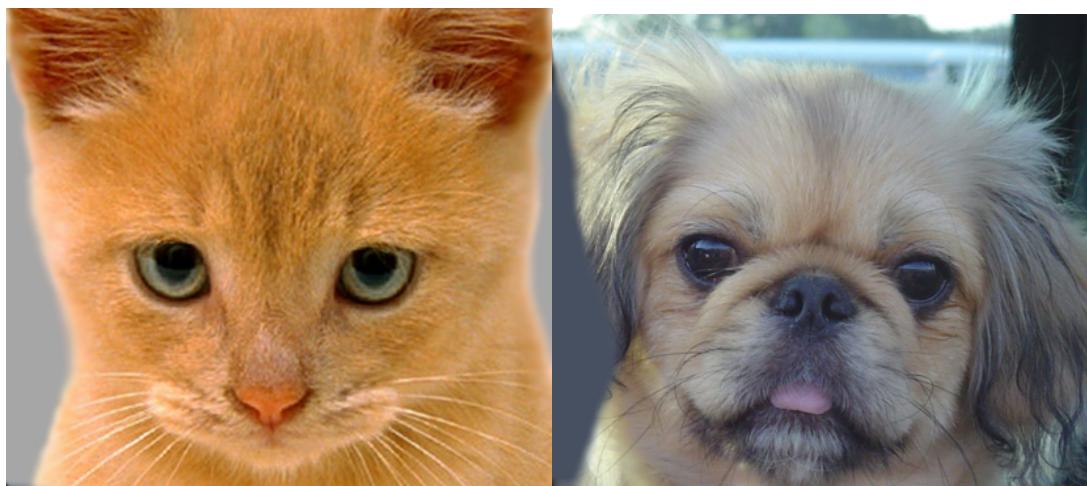
filtered



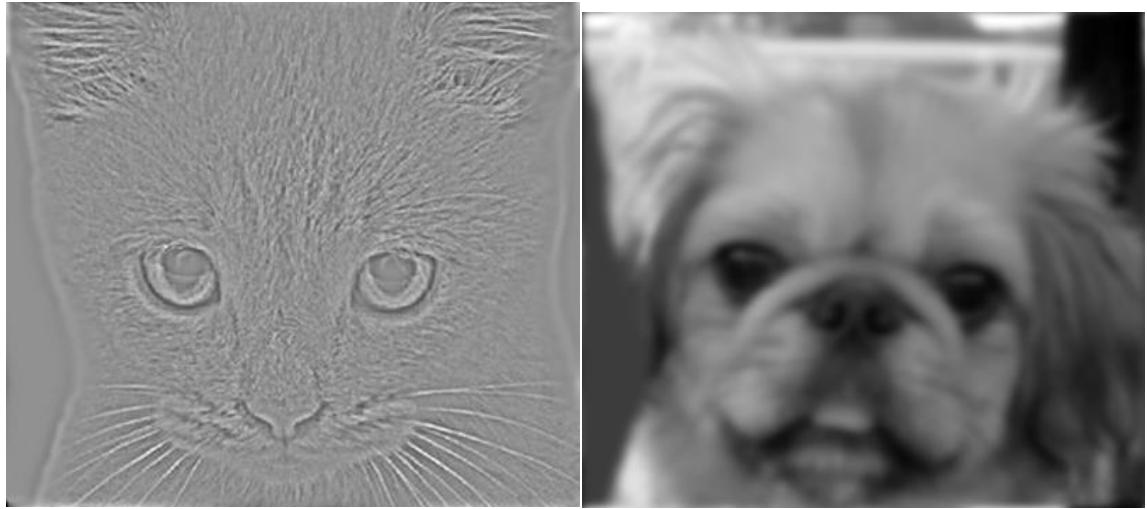
hybrid



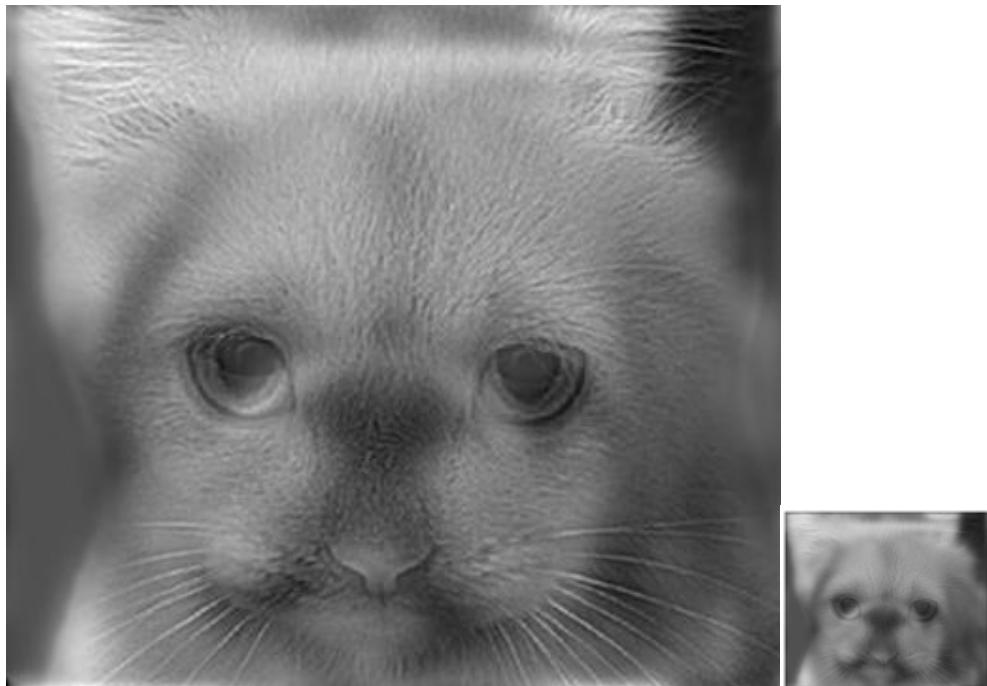
2) original



filtered



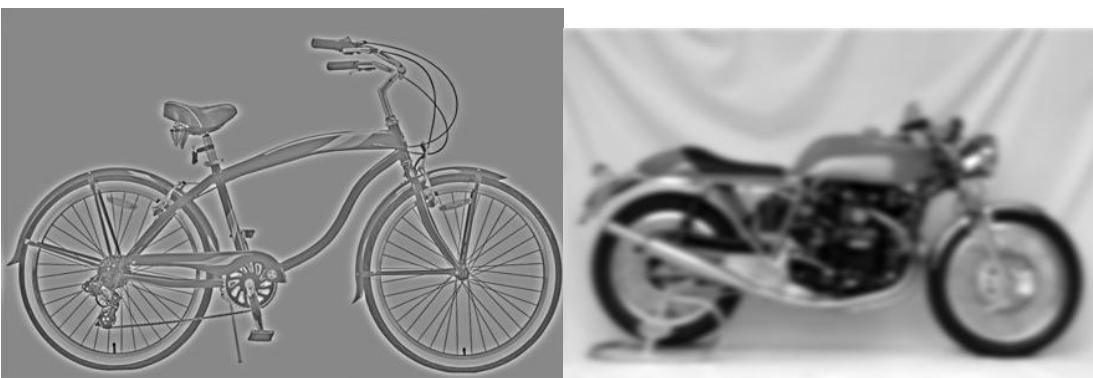
hybrid



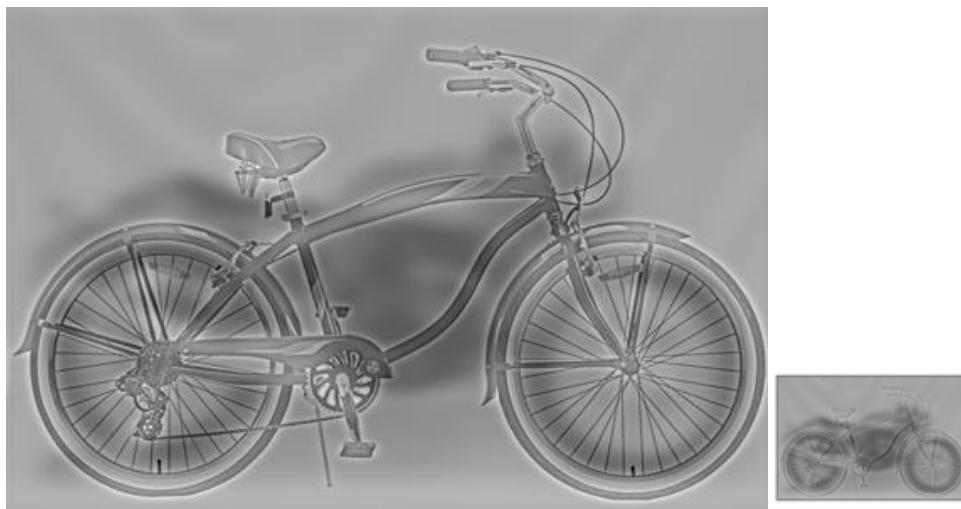
3) original



filtered



hybrid



2. Explanation of any implementation choices such as library functions or changes to speed up computation.

I use `scipy.ndimage.filters.gaussian_laplace` to simplify the `gaussian_laplace` calculation.

3. For each example, give the two σ values you used. Explain how you arrived at these values. Discuss how successful your examples are or any interesting observations you have made.

I firstly chose 20 for both high and low pass, then if I reduced the high_pass σ to 10, the zoom out image show more clear image of low-passed image. But both cases the hybrid image looks good.

4. Optional: discussion and results of any bonus items you have implemented.

Part 2

5. The output of your circle detector on all the images (four provided and four of your own choice), together with running times for both the "efficient" and the "inefficient" implementation.

1) inefficient-time: 1.4 seconds

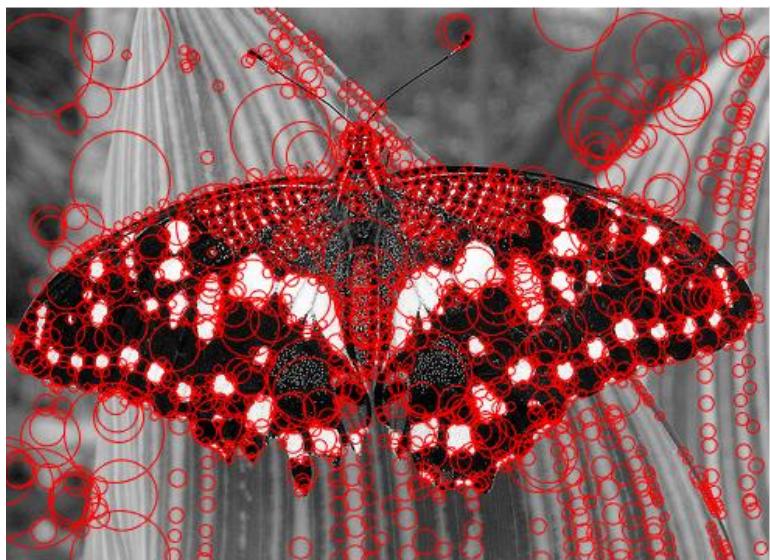
efficient-time:0.4 seconds





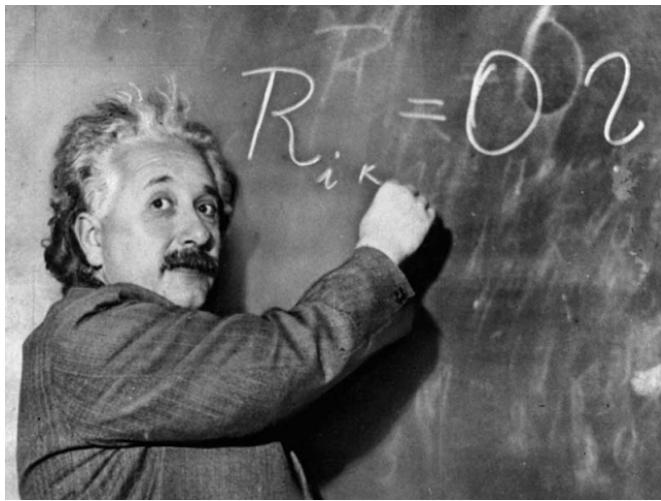
inefficient

efficient

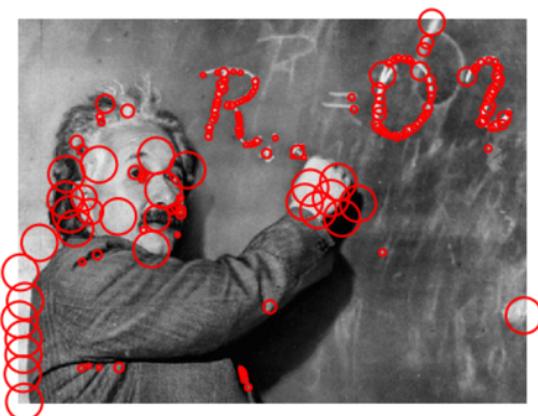


2) inefficient-time: 1.2 seconds

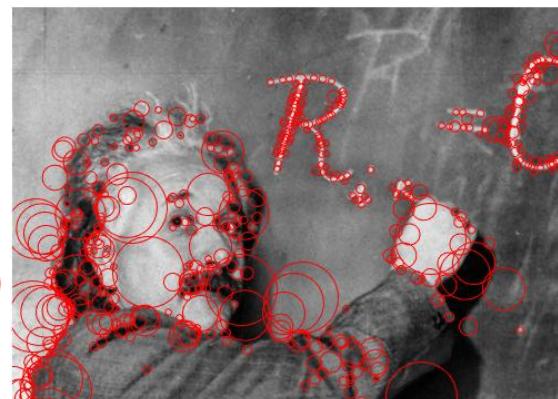
efficient-time:0.36seconds



inefficient



efficient

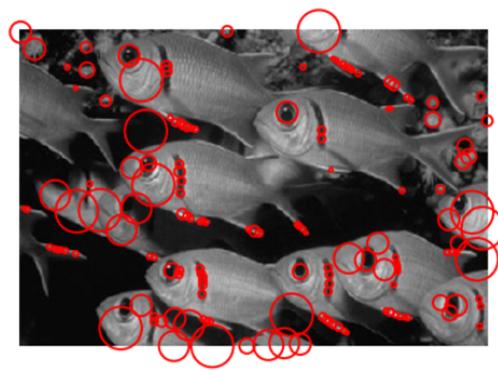


3) inefficient-time: 1.2 seconds

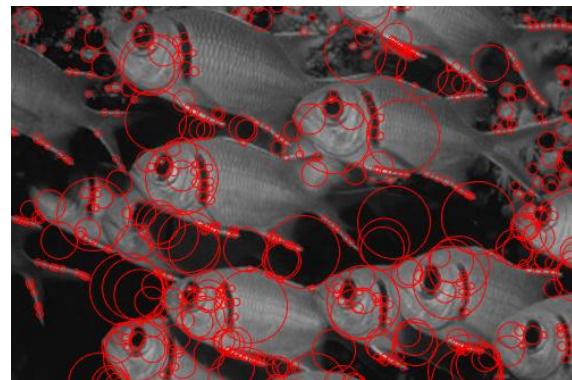
efficient-time: 0.32 seconds



inefficient



efficient



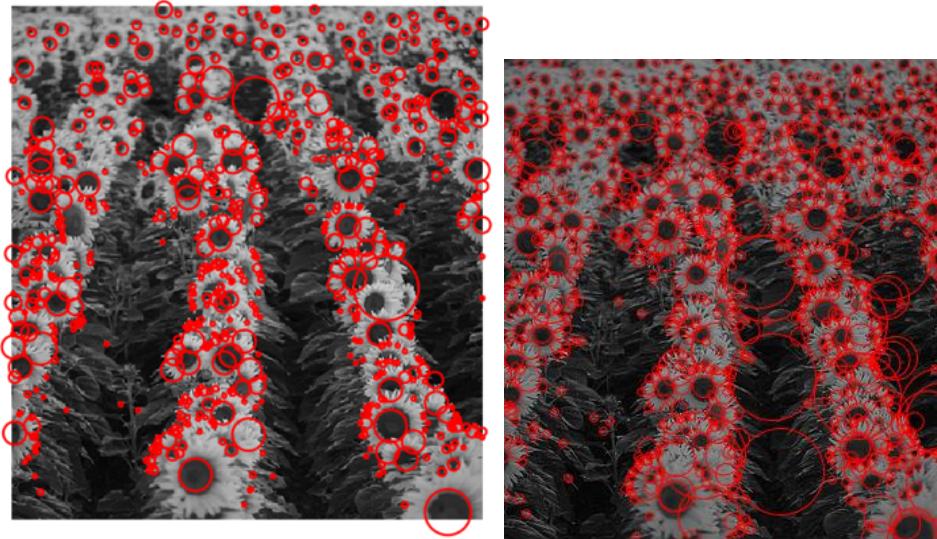
4) inefficient-time: 1.3 seconds

efficient :0.3 seconds



inefficient

efficient



5) inefficient-time: 1.5 seconds

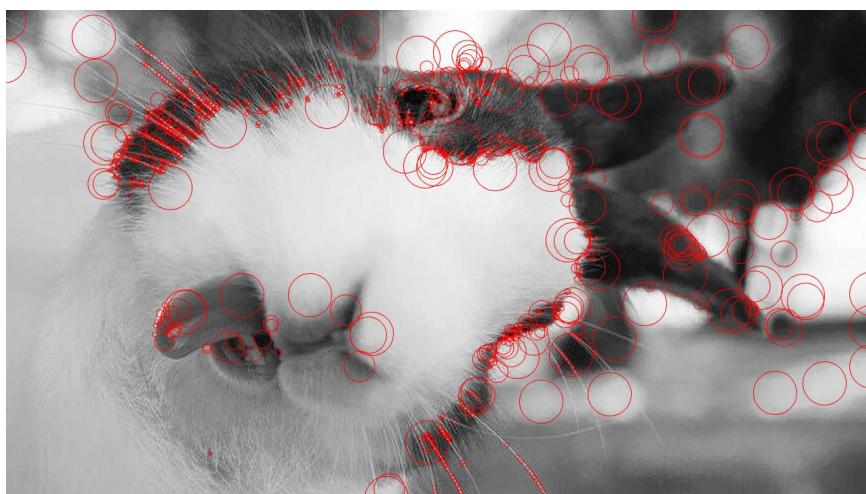
efficient : 0.4s



inefficient



efficient



6) inefficient-time: 1.2 seconds

efficient : 0.28s



inefficient



efficient



7) inefficient-time: 1.4 seconds

efficient : 0.4s



inefficient

effcient



6. An explanation of any "interesting" implementation choices that you made.

If the image is less complicated and has same color at a large area, the circle are fewer but it may not take less time to get the result. And it is not necessary related the edge should be sharp.

7. An explanation of parameter values you have tried and which ones you found to be optimal.

I chose the different thresholds for each picture. Here above, Butterlfy.jpg = 0.03 Sunflowers.jpg = 0.03 fishes = 0.02, Einstein = 0.02. The thresholds would effect the number of detections, which means the lower the

threshold is, the more blobs you would get. I realized a value ~ 0.01 works best for given other parameters. And also the initial setting of the max radius of the circle effect the max size of blob you can get.

8. Discussion and results of any extensions or bonus features you have implemented.

the selection of thresholds should be less than 0.03, but not too small, otherwise, you would get a lot of small circles and messy plot, and it is slow.

Another is we could use some extra function to get over more overlap.