Homework Assignment 2, CS696, applied computer vision

Hsuan Yu, Liu

**My algorithm and decision,**

Before starting to filter, used two if-statements to make sure width and height of a filter are odd. I used two for loop to implement the filter. One for tracing a row of a filter and one for tracing a column of the filter.

At first, that is because the function has to pad an image with zeros, so the algorithm will create a zeros matrix whose size is (filter row/2 + image row)\*( filter column/2 + image column). After that, use one for loop to deal with different layers of a image. Black&White only has one layer. Color has three layers.

Secondly, for every layer, the algorithm will copy the matrix of the layer into center of padding matrix. Then the algorithm can get a padding image with zeros. Also, create a zero matrix to contain values which will be results from the filter.

Thirdly, values of result from a filter will be sum of multiplying cells by filters.

Example:

Assume Filter = [0,1,0].

image(1,1) = SUM ( pad\_image(1,1)\*filter(1,1), pad\_image(1,2)\*filter(1,2), pad\_image(1,3)\*filter(1,3) )

Finally, after calculated a new layer, the algorithm will put the layer into output image. The output image may be 2D matrix or 3D matrix. If it’s a black&white image, an output is 2D. Otherwise is 3D.

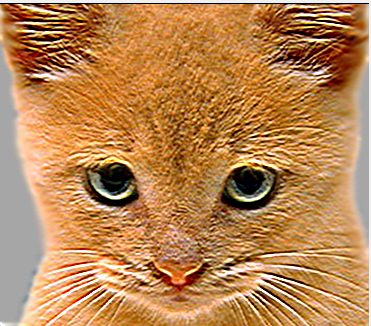
The way I decided is calculate a 2D convolution of matrix, because it’s the fastest algorithm. At beginning, I used matrix selection and element-wise multiplication. Even though it’s just 2 nested loops, the speed is still 1/3 of my convolution algorithm. My 2D convolution has 5 nested loops.

**Test and verification,**

The results of my\_filter() are the same as the results of imfilter() with the same filters.

Filter: [ 1, 2, 1 ; 0, 1, 0 ; -1, -2, -1]

Imfilter(): My\_imfilter():

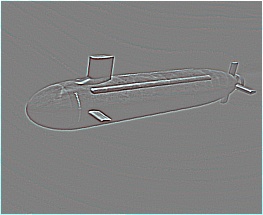
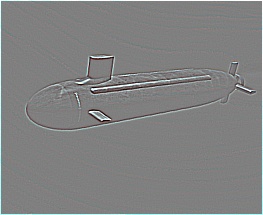
Filter: Small blur with a box filter. [1/9, 1/9, 1/9 ; 1/9, 1/9, 1/9 ; 1/9, 1/9, 1/9]

Imfilter(): My\_imfilter():

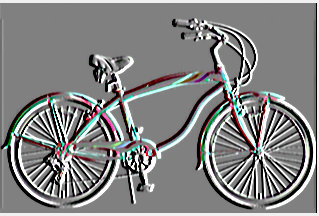
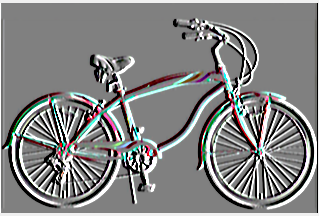
Laplacian Filter [-1 -1 -1; -1 8 -1; -1 -1 -1]

Imfilter(): My\_imfilter():

Sobel filter = [-1, 0, 1 ; -2, 0, 2 ; -1, 0, 1]

Imfilter(): My\_imfilter():

**The intermediate images in the hybrid image pipeline**,

Dog and Cat🡺 cutoff\_frequency = 5

Add one empty column into each image because the columns of the cat image and the dog image are even.

Dog, Low frequencies Cat, High frequencies

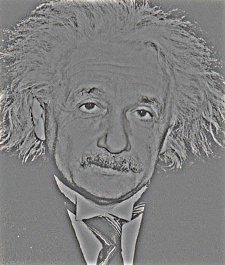
 

Dog & Cat Hybrid scales

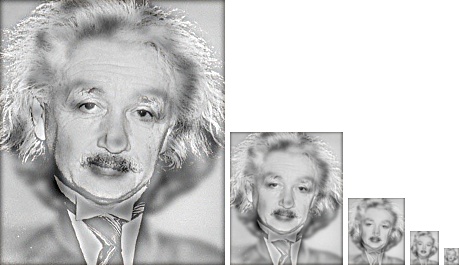


Marilyn and Einstein 🡺 cutoff\_frequency = 3

Marilyn, Low frequencies Einstein, High frequencies

Einstein & Marilyn Hybrid Scales



Bird and Plane 🡺 cutoff\_frequency = 5

Bird, Low frequencies plane, High frequencies

Bird & Plane Hybrid Scales

