Homework 2 Questions

Instructions

- 4 questions.
- Write code where appropriate.
- Feel free to include images or equations.
- Please make this document anonymous.
- **Please use only the space provided and keep the page breaks.** Please do not make new pages, nor remove pages. The document is a template to help grading.
- If you really need extra space, please use new pages at the end of the document and refer us to it in your answers.

Questions

Q1: Explicitly describe image convolution: the input, the transformation, and the output. Why is it useful for computer vision?

A1: Assuming the input is $n \times n$ data and transformation is $2k + 1 \times 2k + 1$, then the output will be $n - 2k \times n - 2k$ data.

And convolution means

$$h[m, n] = \sum_{k,l} f[k, l] I[m - k, n - l]$$

Q2: What is the difference between convolution and correlation? Construct a scenario which produces a different output between both operations.

Please use imfilter to experiment! Look at the 'options' parameter in MATLAB Help to learn how to switch the underlying operation from correlation to convolution.

A2: The most different point is the direction of the calculation. In other words, in the image, the direction in which the filter operates is different.

2D convolution is

$$h[m,n] = \sum_{k,l} f[k,l]I[m-k,n-l].$$

2D correlation is

$$h[m,n] = \sum_{k,l} f[k,l]I[m+k,n+l].$$

The order of the calculations is different only because the k,l sign is negative or positive.

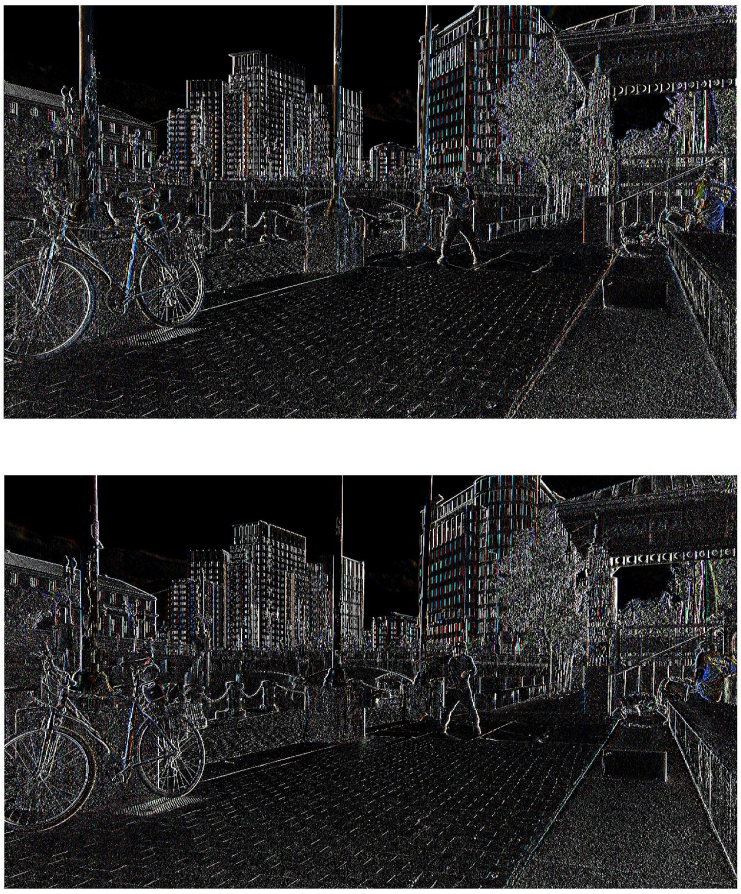

Figure 1: *Up:* correlation. *Down:* convolution. both use Vertical Edge(Sobel) filter.

The two pictures emphasize the opposite side of each other.

Q3: What is the difference between a high pass filter and a low pass filter in how they are constructed, and what they do to the image? Please provide example kernels and output images.

A3: Your answer here. The low pass filter was used fspecial('gaussian',a,b) and the high pass filter was used to subtract the value of fspecial ('gaussian',a,b) from the original image. The a,b value on both sides were adjusted by viewing the picture being printed. There are hybrid images.

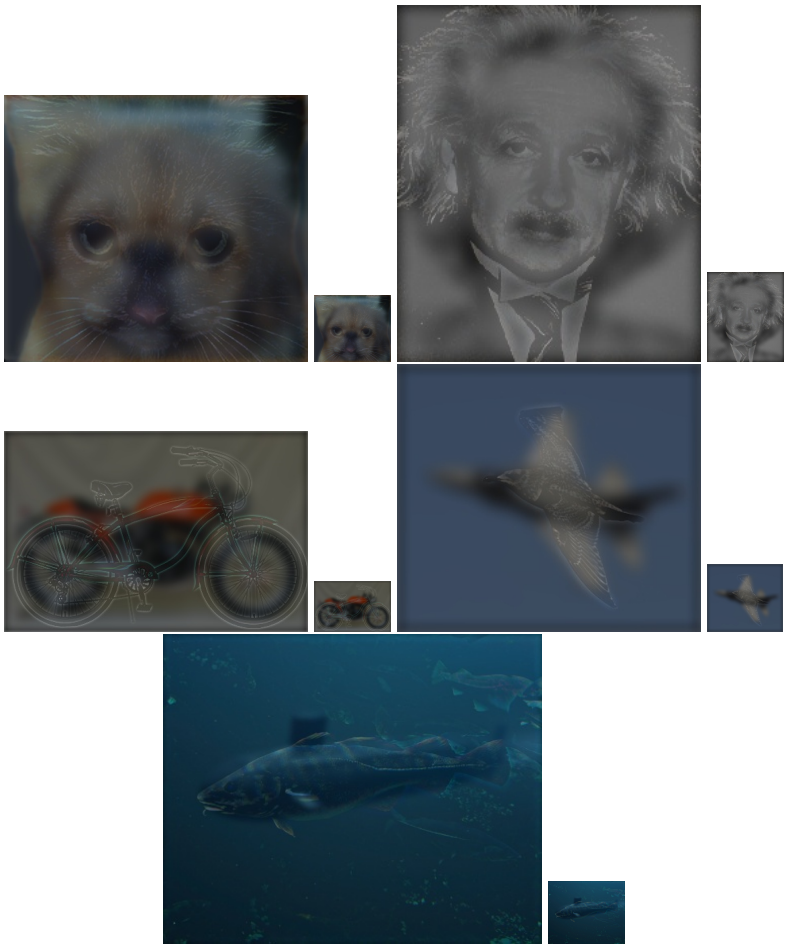

Figure 2: *Left:* We can see cat/einstein/bicycle/bird/fish. *Right:* We can see dog/marilyn/-motorcycle/palne/submarine.

Q4: How does computation time vary with filter sizes from 3×3 to 15×15 (for all odd and square sizes), and with image sizes from 0.25 MPix to 8 MPix (choose your own intervals)? Measure both using $imfilter$ to produce a matrix of values. Use the $imresize$ function to vary the size of an image. Use an appropriate charting function to plot your matrix of results, such as $scatter3$ or $surf$.

Do the results match your expectation given the number of multiply and add operations in convolution?

See RISDance.jpg in the attached file.

A4: Your answer here.

1. Multiple and add as large as filter size for each pixel.
2. Repeat for each pixel.

When the filter size is $(2k+1) \times (2k+1)$ and the number of pixels b, the total number of operations is $(2k+1) \times (2k+1) \times b \times (multiple + add)$.

And the test results and codes are below.

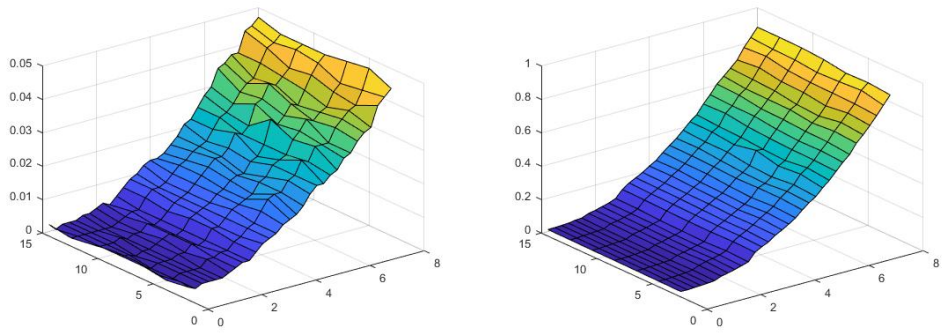

Figure 3: *Left:* just one try. *Right:* sum of 20 tries.

A4 Countinued:

```
i1=im2double(imread('RISDance.jpg'));
[s1,s2,s3] = size(i1);
size1 = s1*s2*s3;
sizegap = 262144/size1;
Time = zeros(7,32);
x=3:2:15;
y=0.25:0.25:8;
[X,Y]=meshgrid(y,x);
for a=1:7
    filter = fspecial('Gaussian', [2*a+1 2*a+1], 10);
    for b=1:32
        I = imresize(i1,sizegap*b,'bilinear');
        t1 = clock;
        for c = 1:20
            newone = imfilter(I,filter,'conv');
        end
        t2 = clock;
        Time(a,b) = etime(t2,t1);
    end
end
surf(X,Y,Time);
```