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: When you have the image matrix I[width,height] and filter f[width,height], the convolution matrix h is following

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

We can get the response by going through the filter over the image. For example, we can extract the horizontal edge of the image by calcualting the convolution of the image and [1 0 1] filter.

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: Correlation shows the similarity of two data. Therefore, we can calculate the similarity of two images by calculating the correlation of them after normalize. (Cosign similarity is one of the method) When you use this on two same image, the normalized correlation value is maximized when two images are exactly overlapped.

On the other hand, convolution is the multicative response when the data go through the signal. For example, when the image go through the low pass filter, the high frequency component dissapear.

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. In case of the low pass filter, it has a tendency to average the wide area. Thus, the image which is passed the low pass filter is robust on the pixel level difference and gets smoother. On the other hand, in case of the high pass filter, it detect the stiff cliff of the pixel value. Therefore, it has a large value when some pixel's value drastically changed compare to the neighbor pixel. Following is the example of the LPF and HPF.

$$LPF = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} HPF = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 2 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix}$$

Following is the example of the iamges which passed the low pass filter and high pass filter.

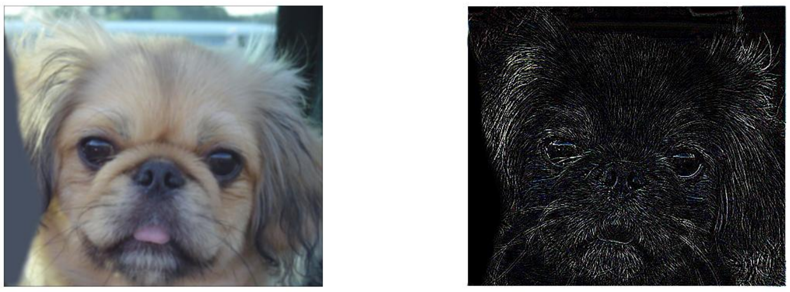

Figure 1: Left is LPF image and right one passed HPF

LPF image got blurred and HPF image shows the edges

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: Let's assume that there is a special operation for the matirx multiplication so the computation time does not follow the linear proportion of the number of element of thematrix. If the assumtion is true, then the computation time for 8x8 filter and 15x15 filter will have linear or constant difference. However, as far as it requires same number of computation with the image size, (or larger matrix) the time for computation with 8Mpixel image will take almost 32times of 0.25Mpixel image's one.

Following is the experiment result. 1000times repeated and calcultated the average of the time.

0.25M pixcel, 8x8 filter, 0.004084sec/img

0.25M pixcel, 15x15 filter, 0.006916sec/img

8M pixcel, 8x8 filter, 0.107183sec/img

8M pixcel, 15x15 filter, 0.112047sec/img

16M pixcel, 15x15 filter, 0.232047sec/img

the time difference between 8x8 filter and 15x15 filter does not follow the linear proportion, and the difference between 0.25M image and 8M image 16M image roughly follow the linear proportion.