ELET 6397

Homework 04

Due Saturday April 27 at 11:59 pm

MatLab offers a function that will create certain common filter matrices for you. The function, called fspecial, requires an argument that specifies the kind of filter you would like. A full description of fspecial is available in MATLAB. Similarly, the imfilter function can be used to apply spatial filtering operations of convolution and correlation. Also imnoise function can be used to corrupt images with noise.

1. Design a simple 'averaging' filter, w, as a 31 x 31 array of ones. Evaluate the effect of using the filter w, with and without normalization. The proper normalization for this mask would be to divide it by (31*31). Load the gray-scale image hw4_im1.tif which is the image of a 2 x 2 checkerboard grid with white and black squares. Filter this gray-scale image using 'replicate' for boundary options, first with the unnormalized 31x31, and second using the normalized averaging window. What differences, if any, are there between the two filter outputs. Why do you think the differences occur?

2. Compare the effects of few built-in classes of linear filters in MatLab, namely the Gaussian filter, the disk filter, and the averaging filter, on the image hw4_im2.tif. Design three lowpass filters/masks, namely a gaussian mask of size 11 and sigma 1.5, an disk mask of radius 5, and an 11 x 11 averaging mask. Using 'replicate' for boundary options, apply spatial linear filtering (convolution) with each mask to the input gray scale image (hw4_im2.tif) and determine the filtered output image,. Show each image and compare and contrast the three filters in terms of what each does to the gray scale image (i.e. explain the differences and your understanding of what features of the filter cause the differences you see). Why would one want to apply any of these filters to real images?

3. Filter image hw4_im3.tif. Use fspecial to
   a. design a 3x3 Sobel filter to detect horizontal edges
   b. design a 3x3 Sobel filter to detect vertical edges
   c. Add the horizontal edge image from (a) to the vertical edge image from (b) display the resulting image.
   d. Next use the laplacian option of fspecial to generate a gradient image of building.tif

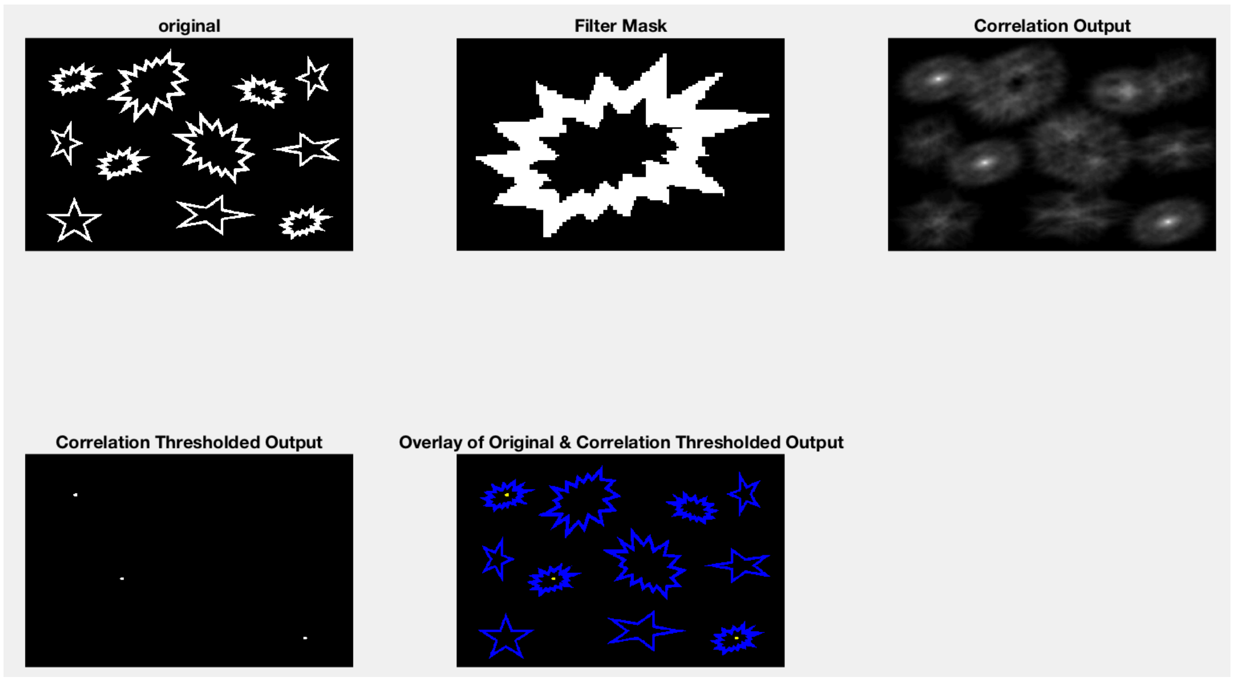
   Show the filtered images and compare and contrast the four operations.

4. In this question, you will compare unsharp and high boost filtering. The algorithm is as follows:
   1. Blur the original image $f(x, y)$ to produce $\bar{f}(x, y)$.
   2. Subtract blurred image from original (result is mask): $g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$.
   3. Add mask to original image: $g(x, y) = f(x, y) + k \cdot g_{mask}(x, y)$.

   - For $k = 1$, the above process is called *unsharp masking*.
   - For $k > 1$, the above process is called *highboost filtering*.

   Use the hw4_im4.tif image, and first blur it with a Gaussian filter of size 7 and sigma of 2. Next used the output of the gaussian filtering and generate output images using unsharp masking (k=1) and high boost filtering (with k=2). Compare and contrast the resulting images.

5. Use correlation for template matching. Read the image, hw4_im5.tif and filter mask hw4_mask5.tif. Calculate correlation between image and mask. Identify peaks of correlation output by thresholding and overlay peaks onto original image to confirm matching. See sample output below.


original
Filter Mask
Correlation Output
Correlation Thresholded Output
Overlay of Original & Correlation Thresholded Output