Lab 5 – Convolution – 2-D

EE3221-051 Digital Signal Processing

Dr. Marek Trawicki

By: Andrew Iliescu, Eduardo Diaz, & Than Win

30 January 2020

Introduction

The main objective of this lab consists of the use of MATLAB to apply an input signal to a system (impulse) response. This response will be analyzed by processing them through a series filter by using 2-D convolution in MATLAB.

Procedure

Materials:

* Laptop with MATLAB Software

MATLAB Procedure:

* Load and look at the original built-in input images x[n, m] into the Command Window using the *imread* command in Matlab, where n and m represent the spatial position in the 2-dimensional plane for the image intensity or color x[n, m].
* Given the kernels (masks) h[n, m] and input images x[n, m] , determine the output images using the *imfilter* command in Matlab.
  + y[n, m] = x[n, m] \* h[n, m] = ∑ ∑ x[k, l] h[n-k, m-l]
* Graph the images as the input images x [n, m] and output images y [n, m] using the *imshow* command in Matlab.
* Given the original kernel (mask) and circles.png input image x[n, m], determine the output image y[n, m].



* Graph the input image x[n, m] and output image y [n, m] in a 1 × 2 plot.
* Given the Average LPF and Average HPF kernels (masks) and circles.png input image x[n, m], determine the output images y [ n, m].
* Graph the input image x[n, m] and output images y[n, m] in 2 x 2 plot.





* Given the Weighted Average LPF and Weighted Average HPF kernels (masks) and circles.png input image x[n, m], determine the output images y[n, m] . Graph the input image x[n, m] and output images y[n, m] in 2 x 2 plot.





* Given the edge detection kernel (mask) and coins.png, coloredChips.png, and circlesBrightDark.png input images x[n, m], determine the output images y[n, m]. Graph the input images x[n, m] and output images y[n, m] in 3 x 2 plot.



* Given the sharpen kernels (masks) and spine.tif input image x[n, m] , determine the output images y[n, m]. Graph the input image x[n, m] and output images y[n, m] in 2 x 2 plot.



Results

First, the original input image “circles.png” went through a basic kernel mask resulting in Figure 1.



Figure 1: The input image (Left) and the output image (Right) using the kernel (mask)

Loading the “circles.png” using the *imread* command and running it through the *imfilter* command, will result in Figure 1. Comparing the two images the yori[n, m] output signal looks to have straighter sides on the circles. The input image has the circle’s sides to be more pixilated. Second, the “circles.png” input image went through an average low and average high pass filter.



Figure 2: Input image (Left) and LPF and HPF output image (Right)

Above, in Figure 2, the top right image is the input image ran through the LPF (Low Pass Filter). This output image does not seem to give much of a difference compared to the original input image. On the other hand, the bottom right image is the input image ran through the HIF (High Pass Filter) which has the inner part of the circle components missing. The output image seems to just be the outline of the entire circle cluster. Third, the original image was put through another set of low and high pass filters. However, these filters were processed with a weight on the average, different from the previous set.



Figure 3: Input image (Left) and Weighted Average LPF and HPF output image (Right)

Above, in Figure 2, the top right image is the input image ran through the weighted average LPF (Low Pass Filter). This output image does not seem to give much of a difference compared to the original input image. On the other hand, the bottom right image is the input image ran through the weighted average HIF (High Pass Filter). This output image has a much more drastic change. The entire shape of the circle cluster is gone and is only a few dots of white remain.

Using new input images of coins, chips, and various circles, they were put in an edge detection kernel and the results are shown below.



Figure 4: Input images (Left) and the edge detected output images

The left side column in Figure 4 contains the original input images and the right column is the output image of the input image ran through the edge detection kernel. The output image blacks out the background and the edges of the shape in the original image is exemplified and pops out.

Last, x-rays of a spine went through a sharpening filter to clarify the image



Figure 5: Spine image with the applied sharpening kernels

In Figure 5, the original spine image is dim, and the details of the spine are not very apparent. When the sharpening mask is applied, the three different output images are much more detailed. Staring with kernel 1, the spine itself is brighter and edges can be seen clearly. Kernel 2 improves this affect and brightens up the image even further. Kernel 3 is the brightest out of the 4 images and the most detailed.

Conclusion

Overall, the images used in this lab were modified and changed by passing it through different kinds of filters and analyzing them using 2-D convolution. As each different mask was used and the original image was changed, the output image showed how the original signal was changed specifically. These changes were specific and precise.

Questions & Answers

1. Given the lighten kernels (masks)



And



and yellowlily.jpg input image x[n, m], determine the output images y[n, m]. Graph the input image x[n, m] and output images y[n, m] in separate 2 x 2 plots. Label all the appropriate quantities on the plots. Comments on the results.

The input image of yellow lilies was put through two filters to lighten the image. The first filter has a mask of an identity matrix and the second filter has a mask of a matrix of 1’s. Due to this difference in masks, the output images will vary. The matrixes of these associated filters are given in the question. Figure 6 shows the results of the first filter.



Figure 6: Yellow Lily image with the lightening kernels applied

As described previously, Figure 7 shows the results of the second filter. This second filter has different matrix entries, so it still lightens the images, but to a different degree.



Figure 7: Yellow Lily image with the lightening kernels applied

In Figures 6 and 7, the lightening kernels is applied to the *yellowlily.jpg* image*.* Each progressing image is brighter than the last and the detail of the image is decreasing. Comparing the lighten 3 images with the original, the details of the lily can no longer be seen.

1. Given the emboss kernel (mask)



and lighthouse.png and trailer.jpg input images x[n, m] , determine the output images y[n, m]. Graph the input images x[n, m] and output images y[n, m] in 2 x 2 plot. Label all the appropriate quantities on the plots. Comments on the results.

An image of a light house and a trailer were put through an Emboss filter and the results are shown below.



Figure 8: Trailer and Light House Image with the Emboss Kernel Applied

In Figure 8, the light house and trailer image had the Emboss kernel applied to it. Due to the emboss filter, the images seem to have a higher contrast causing sharper lines. The colors of the resulting pictures are not as soft, and the saturation seems to have increased a bit. This may be caused of how grainy the picture has become. The shadows seem to be a bit darker and the highlights a bit brighter. The edges in the pictures are given the effect of being “etched” in.