Department of Computer Science
Digital Image Processing

N. Komodakis, Course instructor

Autumn 2022

Image Filtering

The goal of this assignment is to get you familiarized with image convolutions and how to use these operations to perform some simple image processing tasks.

1 Implementing 2D convolutions

In the first part of this assignment you are called to implement your own routines for performing 2D convolution on images. You will consider both the general case of convolution with an arbitrary 2D filter and the case of convolution with a separable 2D filter.

(a) **Convolution with an arbitrary 2D filter:** you will implement a function *my_conv(img, kernel)* that returns as an output the **valid** convolution of the input image *img* with the 2D filter *kernel*. To that end, you should use the Matlab functions *im2col, col2im, fliplr, flipud*.

(b) **Convolution with a separable 2D filter:** you will implement a function *my_separable_conv(img, hor_kernel, ver_kernel, pad_type)* that returns as an output the **same-size** convolution of an input image *img* with the 2D separable filter *hor_kernel* $*$ *ver_kernel*, where *hor_kernel*, *ver_kernel* are respectively 1D horizontal, 1D vertical filters and $*$ denotes the convolution operator. The padding type *padType* can be either (i) zero-padding, or (ii) pixel boundary value replication. For the implementation of this function, you will make use of your routine *my_conv* from (a) and the Matlab function *padarray*.

2 Building Your Own Filters

You will need to implement three different image filtering operations, each having a different impact on the image. You are required to present your results and your comments in a report (.pdf). In more detail the operations are the following:

2.1 Smoothing and Binarization

In this part you will apply a Gaussian filter, using three different values for the standard deviation parameter, $\sigma : 3, 5, 7$. Suggest the best padding operation regarding the boundaries of the image.
You will apply the filtering operation in two ways, using the requested functions that are mentioned:

- **matlab_gaussian_filter(img, pad_type)**: Matlab's *imgaussfilt()*
- **my_gaussian_filter(img, pad_type)**: *my_separable_conv()* and Matlab's *fspecial()*

Finally, compare (visually) the binarized smoothed images (both implementations), with the binarized original image. You will apply the requested filtering operations to this image:
`http://www.csd.uoc.gr/~hy371/images/bsds86016.png`

2.2 Local Standard Deviation and Binarization

Implement a routine that uses convolution with box averaging filter to compute an output image that contains the local standard deviation of the pixel intensities (over a 5x5 neighborhood) of an input image, known as local standard deviation filter. The operation of the local standard deviation filter on image I, using box averaging E, is defined as:

$$J = \sqrt{E(I^2) - E^2(I)}$$

You will define the local standard deviation filter in two ways:

- **matlab_stddev_filter(img, pad_type)**: Matlab's *fspecial() and imfilter()*
- **my_stddev_filter(img, pad_type)**: *my_separable_conv()* and Matlab's *fspecial()*

Finally, compare (visually) the binarized filtered images (both implementations), with the binarized original image. You will apply the requested filtering operations to this image:
`http://www.csd.uoc.gr/~hy371/images/facade.png`

2.3 Laplacian and Binarization

For the last part of this assignment you will apply a Laplacian of Gaussian (LoG) filter. You will use an approximation of that filter that is defined as the difference of two Gaussian filters, G_1, G_2. To define the filters, consider that the standard deviation, σ_1, for G_1 is 1.28 times the standard deviation of G_2, noted as σ_2. Evaluate the filtering operation for $\sigma_2 : 1, \sqrt{2}, 2, 2\sqrt{2}$.

You will define the LoG filter in two ways:

- **matlab_log_filter(img, pad_type)**: Matlab's *fspecial() and imfilter()*
- **my_log_filter(img, pad_type)**: *my_separable_conv()* and Matlab's *fspecial()*

Finally, compare (visually) the binarized filtered images, with the binarized original image, and comment on the impact of the value of the standard deviation parameter. You will apply the requested filtering operations to this image:
`http://www.csd.uoc.gr/~hy371/images/plate_usa.png`

Other useful functions: *imread, im2double, imbinarize*