



Lab 3

Averaging Filters

1. Filtering:
 - a. Read an image of your choice.
 - b. Apply salt & pepper noise with density = 0.05
 - c. Implement your own median filtering algorithm using a mask of size 3*3. A pseudocode can be found that can help you.
 - d. Compare your results with python function ***skimage.filters.median***
 - e. Apply **Gaussian Filtering** to the noisy image with different sigma values (ex: 8, 3, & 0.2). Experiment a little to find a suitable value of sigma that fits the image. Be ready to justify your answer.

```
edgex := (window width / 2) rounded down
edgey := (window height / 2) rounded down
for x from edgex to image width - edgex
{
    for y from edgey to image height - edgey
    {
        allocate colorArray[window height][window width]
        for fx from 0 to window width
            for fy from 0 to window height
                colorArray[fy][fx] := inputPixelValue [y + fy - edgey] [x + fx - edgex]
        sort all entries in colorArray[][]
        outputPixelValue[y][x] := colorArray [window height / 2] [window width / 2]
    }
}
```

Useful New Functions and Attributes

Name	Attribute or Function	Usage
round	Function	Round decimal values
np.zeros	Function	To generate matrix with a given shape all elements with zero values
range	Function	To generate range of indexes from low range to high range with an optional step
gaussian	Function	To apply gaussian filter on an image.
np.median	Function	To get median value. <i>To get the median value for all the matrix (not for a certain axis), provide parameter axis=None</i>
skimage.filters.median	Function	To apply median filter on an image.

To make the range of a gray-scale image uint8 (from 0 – 255), use `astype(np.uint8)`. Make sure that the range of values are between 0 & 255 before casting.