

# Lecture 02 - Image Filtering

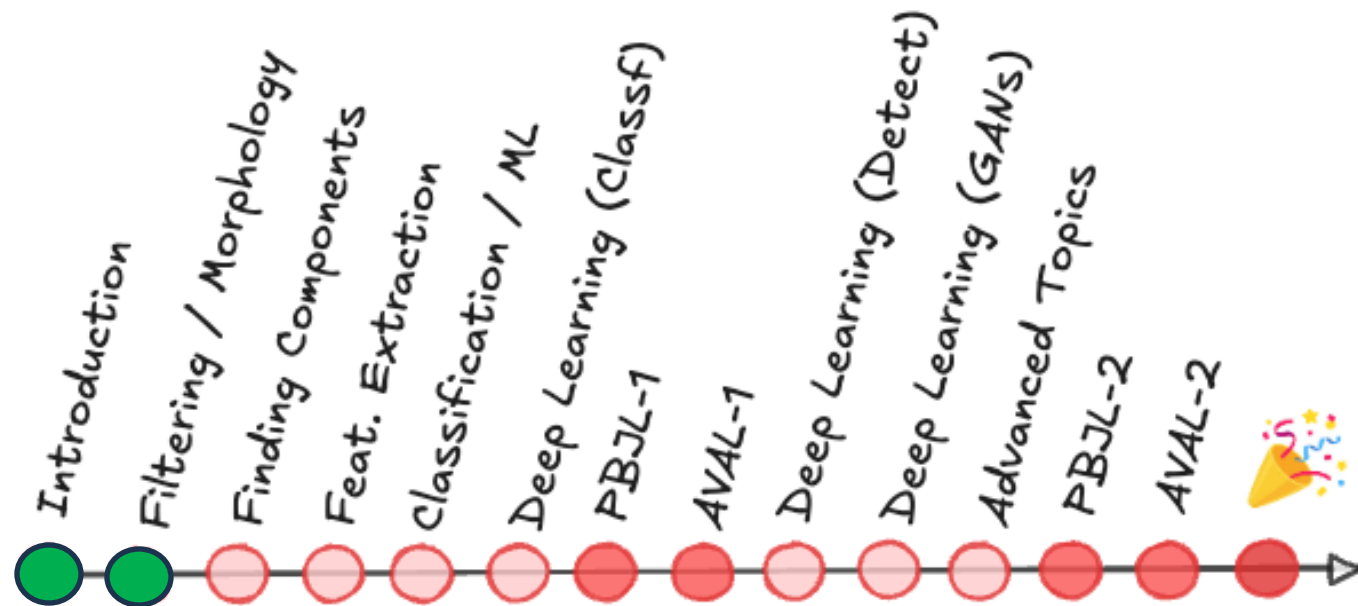
Prof. André Gustavo Hochuli

[gustavo.hochuli@pucpr.br](mailto:gustavo.hochuli@pucpr.br)

[aghochuli@ppgia.pucpr.br](mailto:aghochuli@ppgia.pucpr.br)

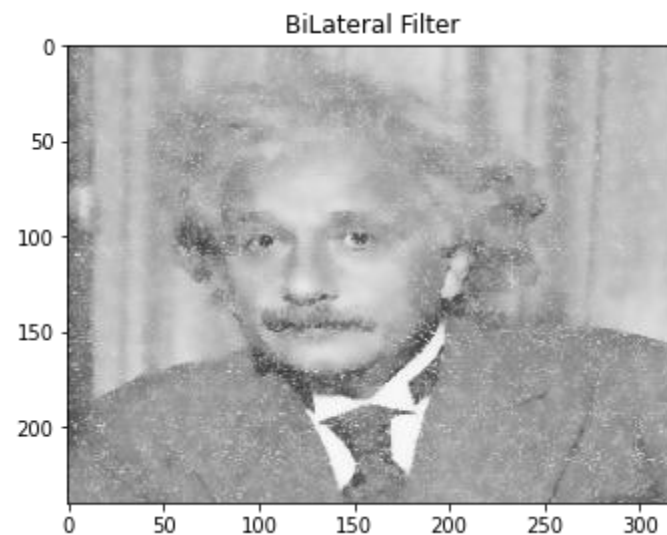
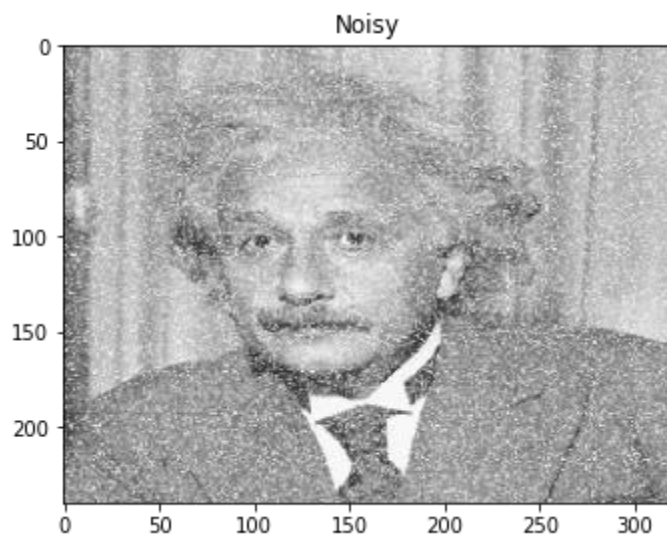
# Topics

- Discussion of Practice 01
- Image Filtering
  - Convolution
  - Mean, Median, Gaussian Filters
- Practice



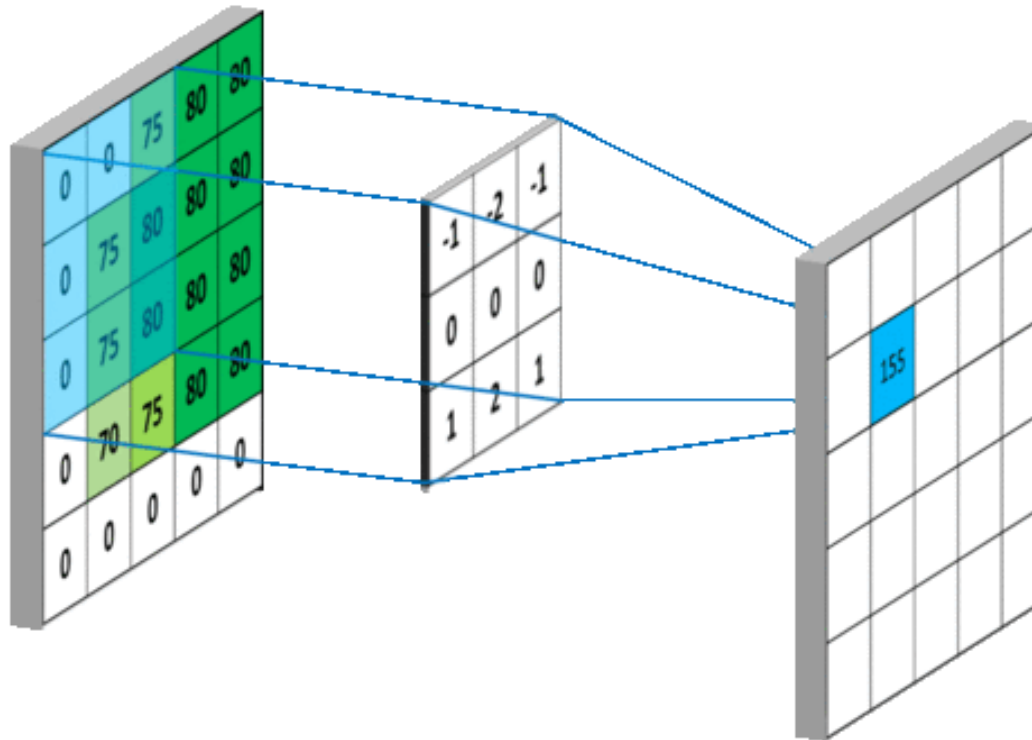
# Image Filtering

- Image Enhancement
- Noise Reduction
- Mathematical Operations



# Convolution

- Slides a kernel (a.k.a convolution filter) in the entire image
- Transforming the pixel in the center of the kernel by the weights of its neighbors



# Convolution

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

\*

1	0	-1
1	0	-1
1	0	-1

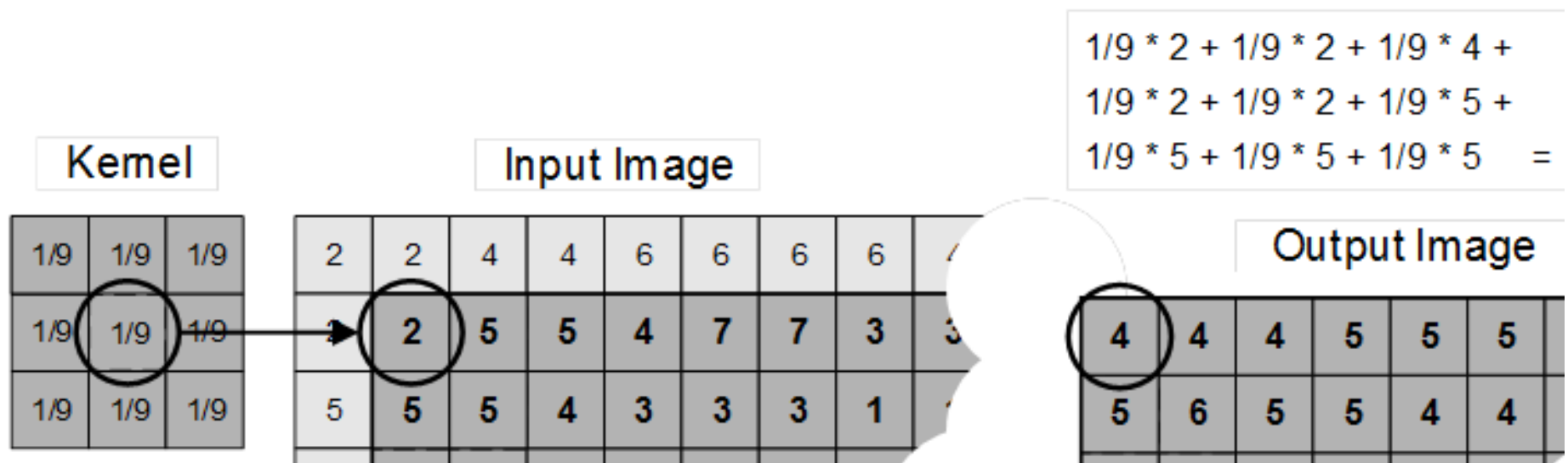
=

6		

$$\begin{aligned} &7 \times 1 + 4 \times 1 + 3 \times 1 + \\ &2 \times 0 + 5 \times 0 + 3 \times 0 + \\ &3 \times -1 + 3 \times -1 + 2 \times -1 \\ &= 6 \end{aligned}$$

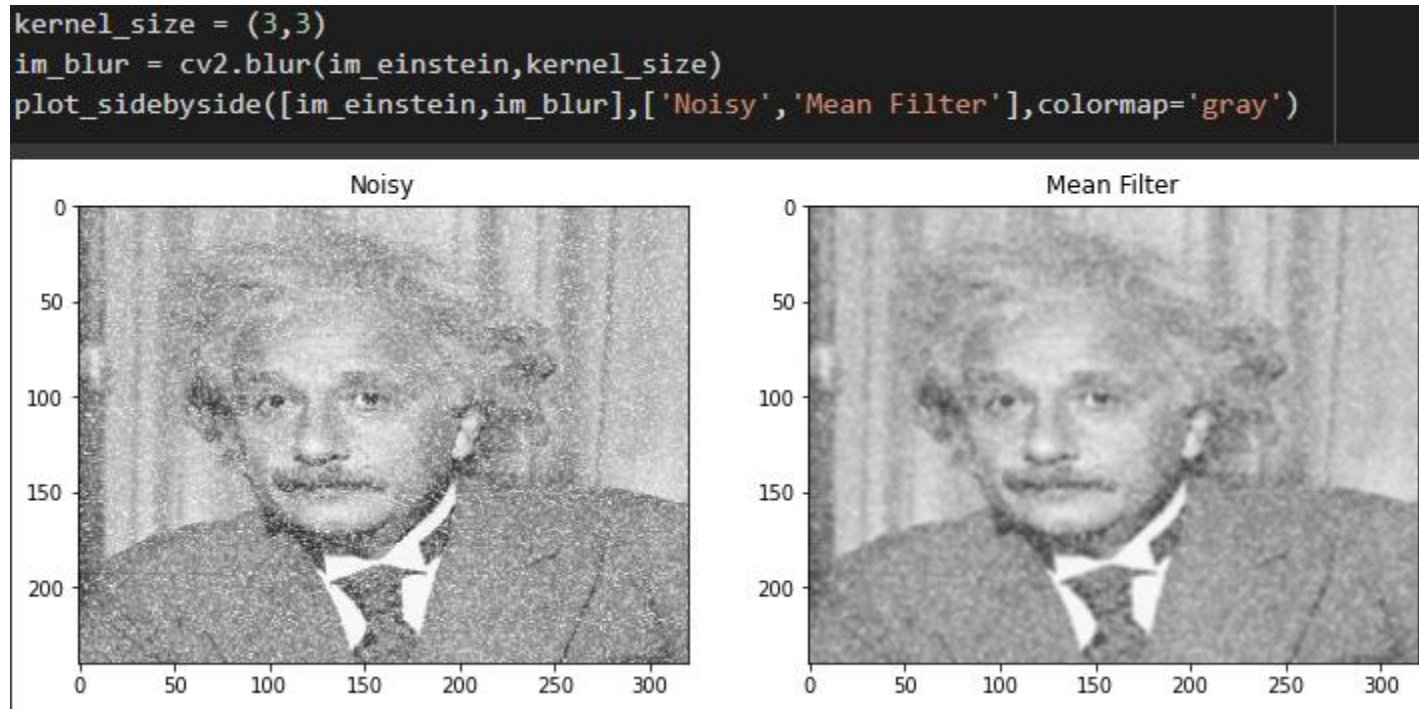
# Mean Filter

- Replaces the center pixel with the mean of its neighborhood
- Spreads the outlier value to its neighbors
- Details are smoothed



# Mean Filter

- Replaces the center pixel with the mean of its neighborhood
- Spreads the outlier value to its neighbors
- Details are smoothed



# Median Filter

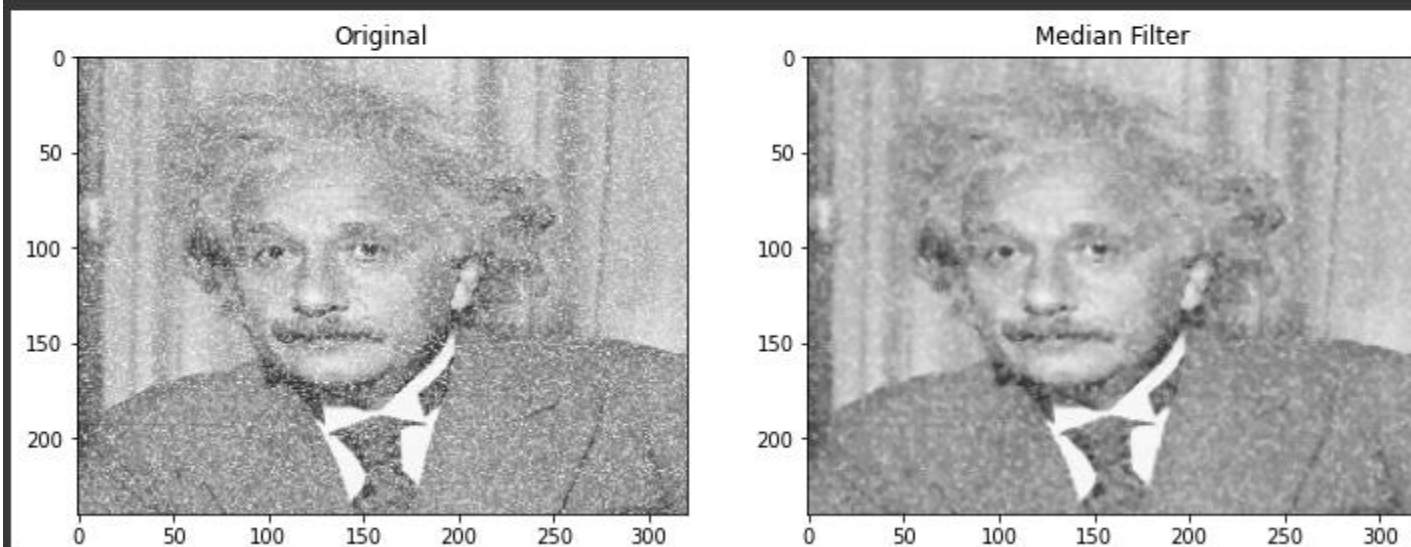
- Replaces the center pixel with by a median of its neighborhood
- Preserves more details when compared to the mean filter

```
im_blur = cv2.medianBlur(im_einstein,3)  
plot_sidebyside([im_einstein,im_blur],['Noisy','Median Filter'],color
```

Input					
1	4	0	1	3	1
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

Output					
1	4	0	1	3	1
2	1	1	1	1	3
1	1	1	1	2	0
1	1	1	1	1	2
2	2	2	2	2	5
1	1	4	2	3	0

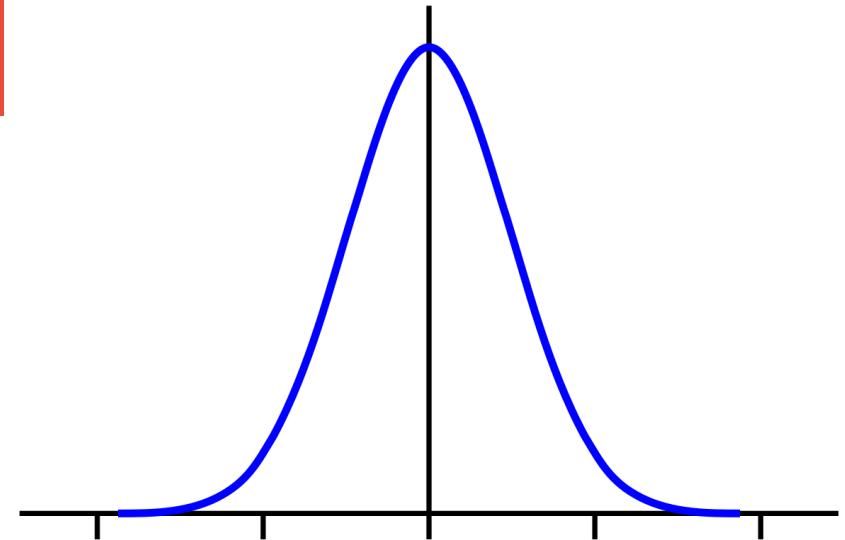
Sorted:0,0,1,1,1,2,2,4,4





# Gaussian Filter

- Gaussian distribution of pixels
- The kernel is composed of probabilities
- Weighted Mean
- The standard deviation determines the blur degree



$$\frac{1}{16}$$

×

1	2	1
2	4	2
1	2	1

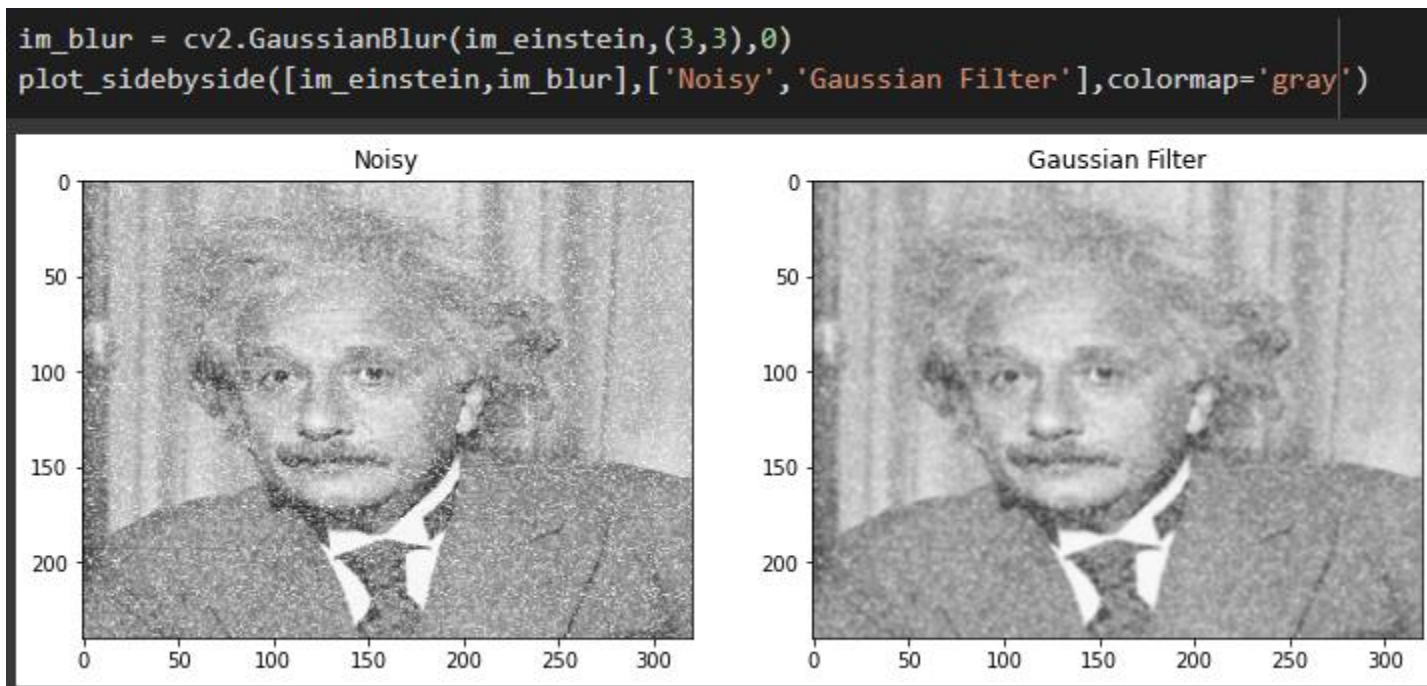
$$\frac{1}{273}$$

×

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

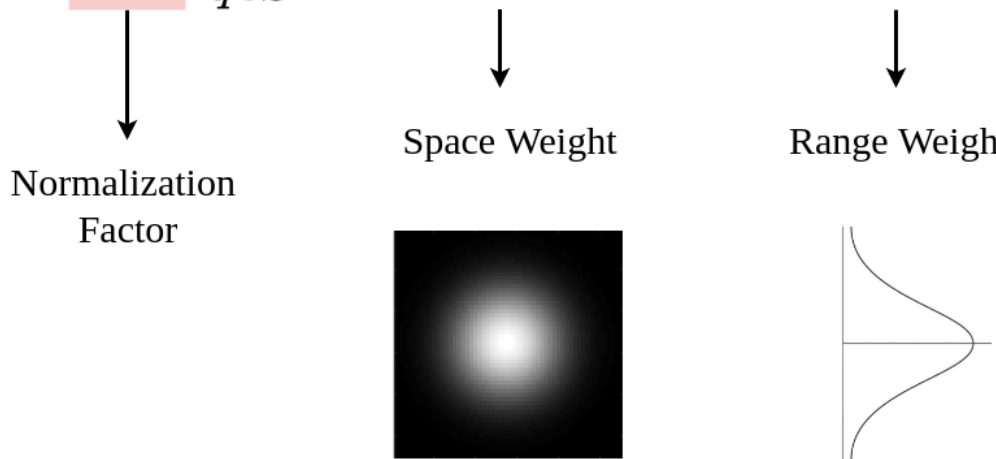
# Gaussian Filter

- Gaussian distribution of pixels
- The kernel is composed of probabilities
- Weighted Mean
- The standard deviation determines the blur degree



# Bilateral Filter

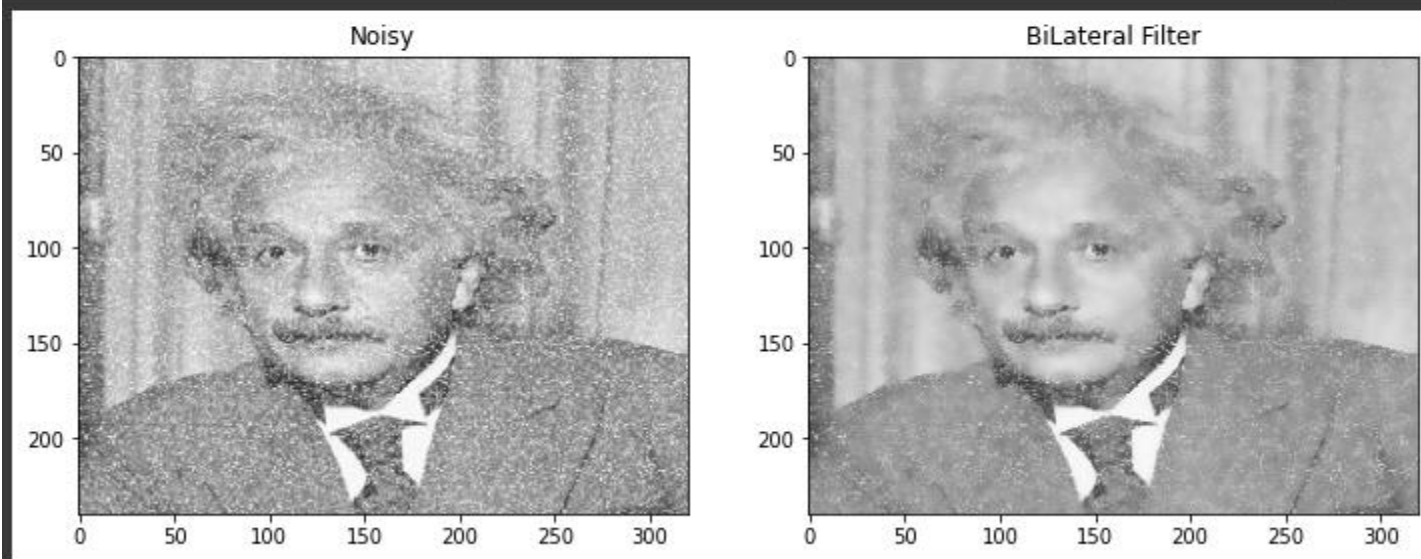
- Gaussian Distribution based (spacial and pixel intensity)
- Add Normalization Factors and Range Weight
- Preserve details

$$BF[I]_p = \underbrace{\frac{1}{W_p}}_{\text{Normalization Factor}} \sum_{q \in S} \underbrace{G_{\sigma_s}(\|p - q\|)}_{\text{Space Weight}} \underbrace{G_{\sigma_r}(|I_p - I_q|)}_{\text{Range Weight}} I_q$$


# Bilateral Filter

- Gaussian Distribution based (spacial and pixel intensity)
- Add Normalization Factors and Range Weight
- Preserve details

```
# Apply bilateral filter with d = 15,  
# sigmaColor = sigmaSpace = 75.  
im_blur = cv2.bilateralFilter(im_einstein,9,75,75)  
plot_sidebyside([im_einstein,im_blur],['Noisy','BiLateral Filter'],colormap='gray')
```



# Practice

[Link: Practice 02](#)