Lecture 02 - Image Filtering

Prof. André Gustavo Hochuli

gustavo.hochuli@pucpr.br
aghochuli@ppgia.pucpr.br

Topics

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

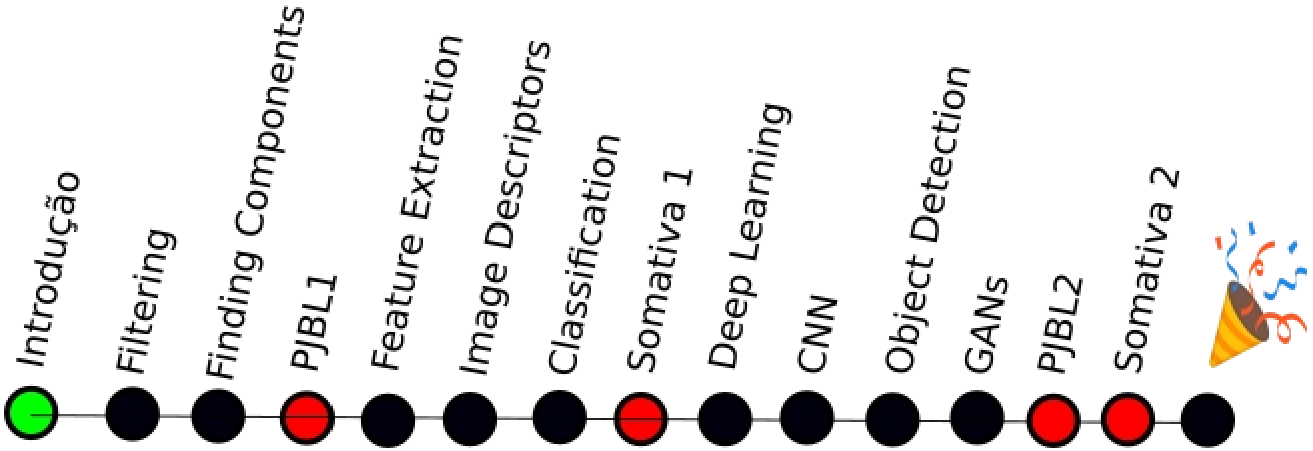
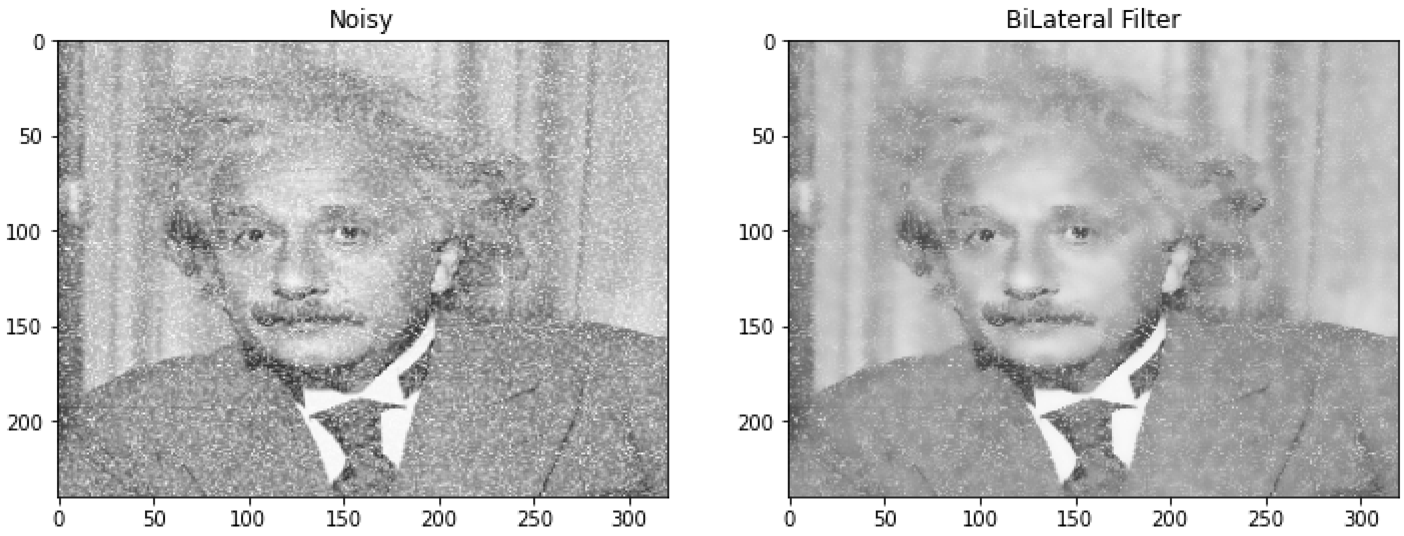


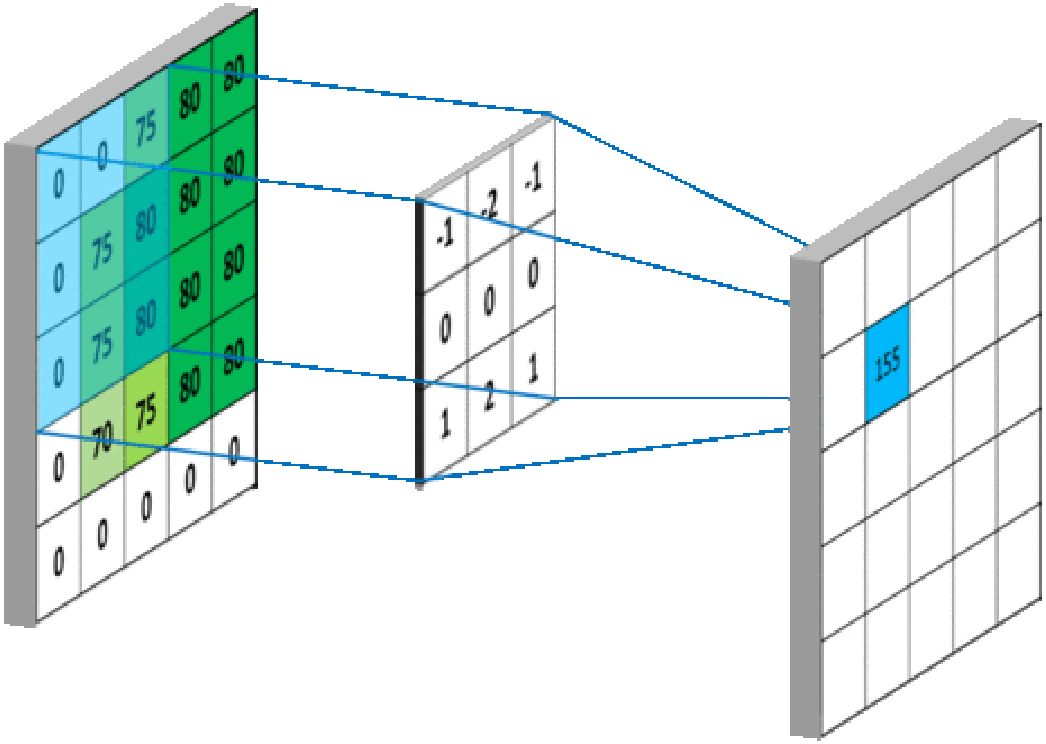
Image Filtering

- **Image Enhacement**
- **Noise Reduction**
- **Mathematical Operations**



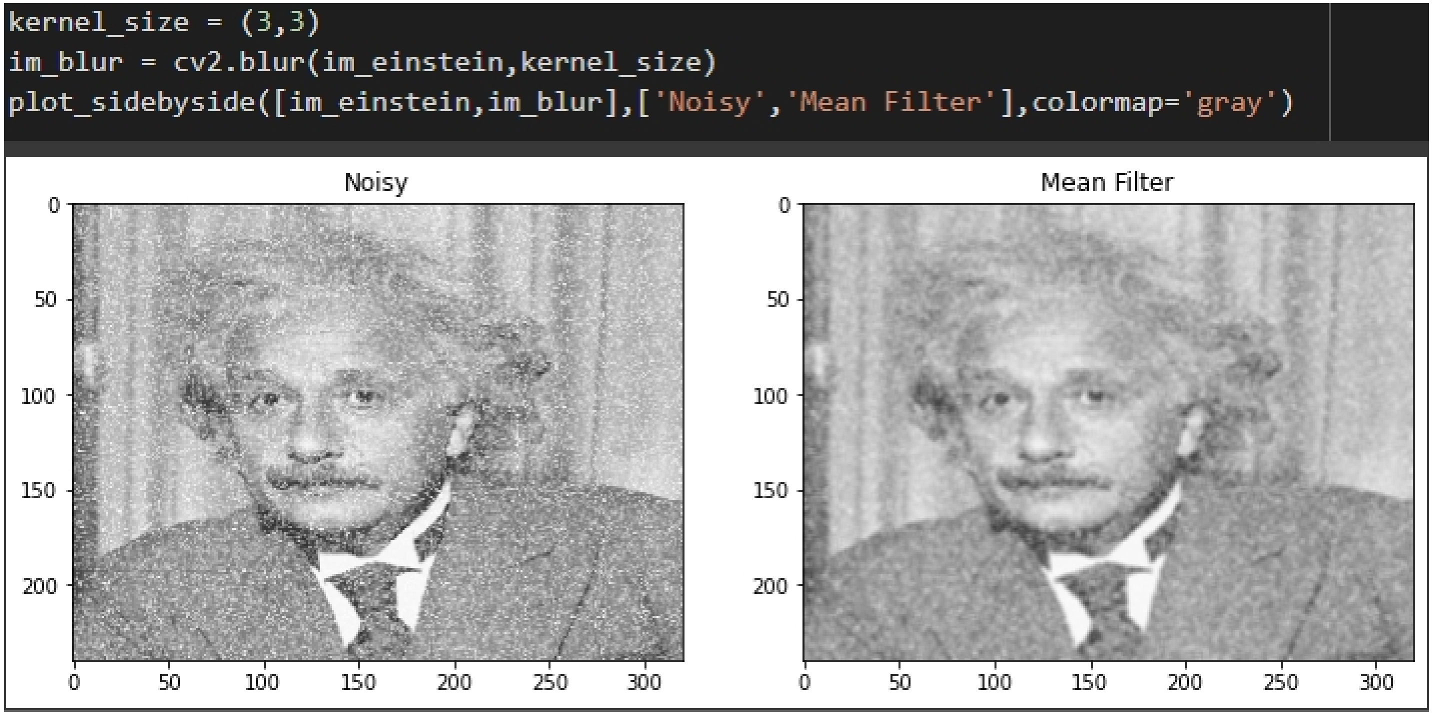
Convolution

- **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**



Mean Filter

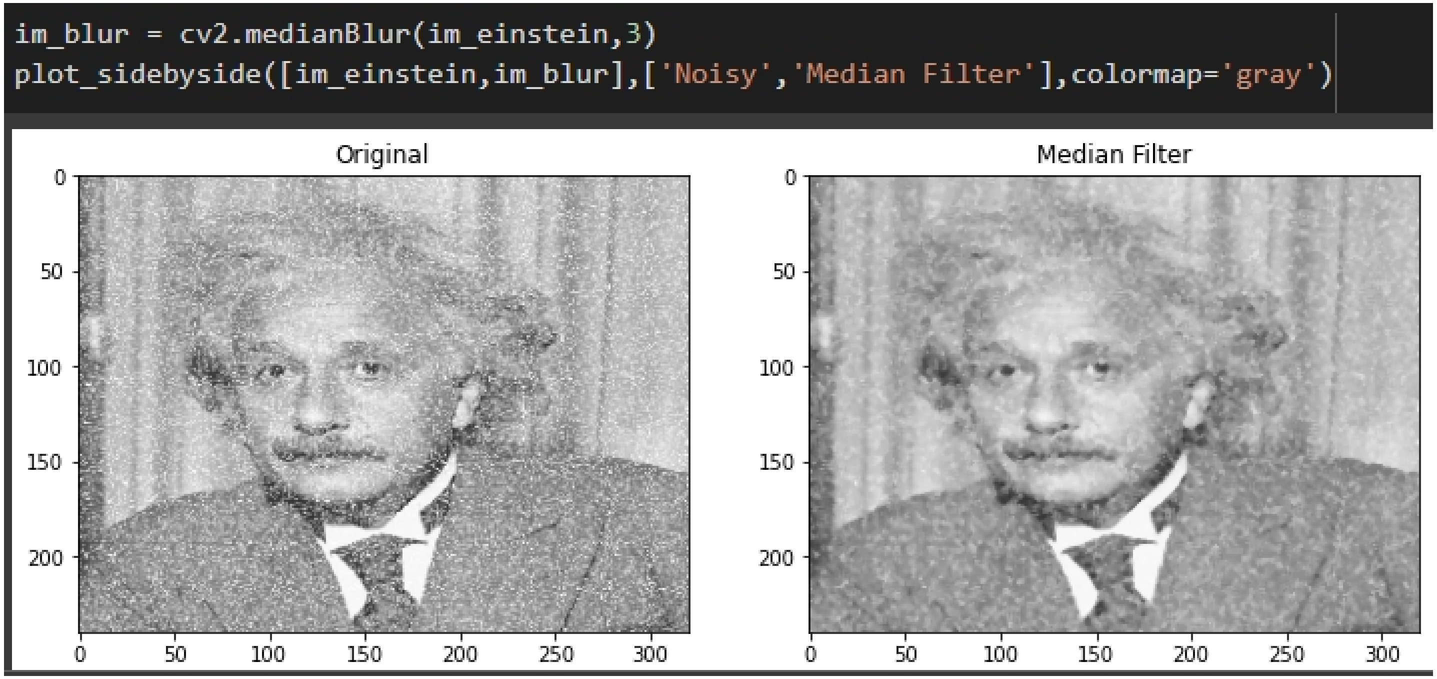
- Replaces the center pixel with the mean of its neighborhood
- Spreads the outlier value to its neighbors
- Details are smoothed



```
kernel_size = (3,3)
im_blur = cv2.blur(im_einstein,kernel_size)
plot_sidebyside([im_einstein,im_blur],['Noisy','Mean Filter'],colormap='gray')
```

Median Filter

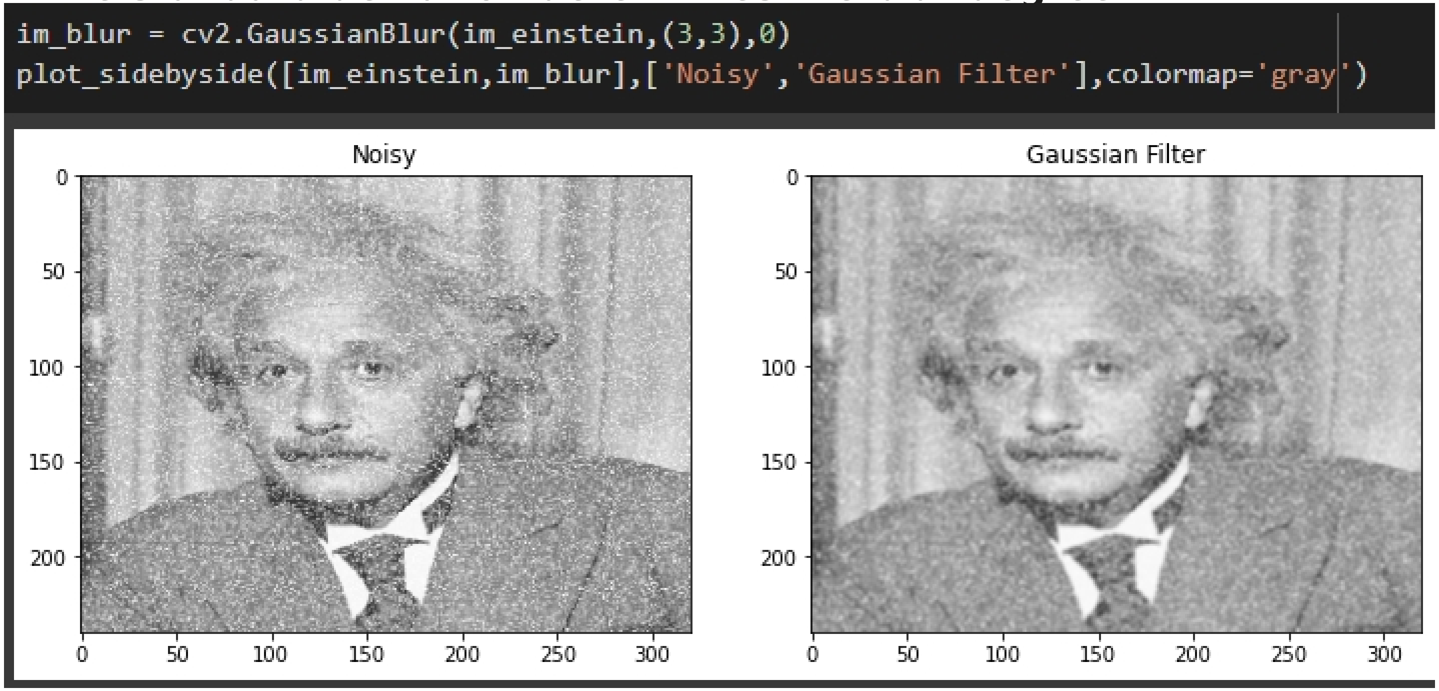
- Replaces the center pixel with by a median of its neighborhood
- Preserves more details when compared to the mean filter



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

Gaussian Filter

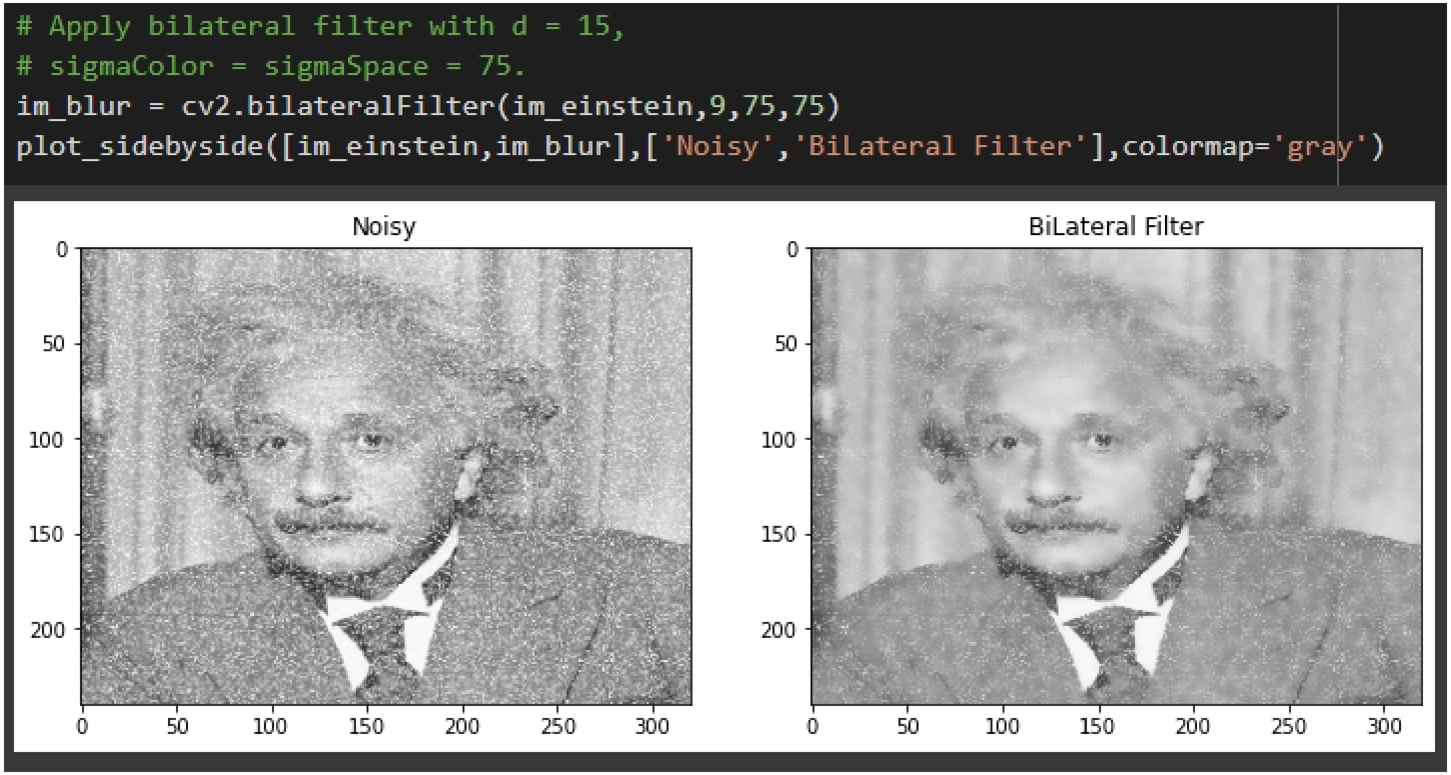
- **Gaussian distribution of pixels**
- **The kernel is composed of probabilities**
- **Weighted Mean**
- **The standard deviation determines the blur degree**



```
im_blur = cv2.GaussianBlur(im_einstein,(3,3),0)
plot_sidebyside([im_einstein,im_blur],['Noisy','Gaussian Filter'],colormap='gray')
```

Bilateral Filter

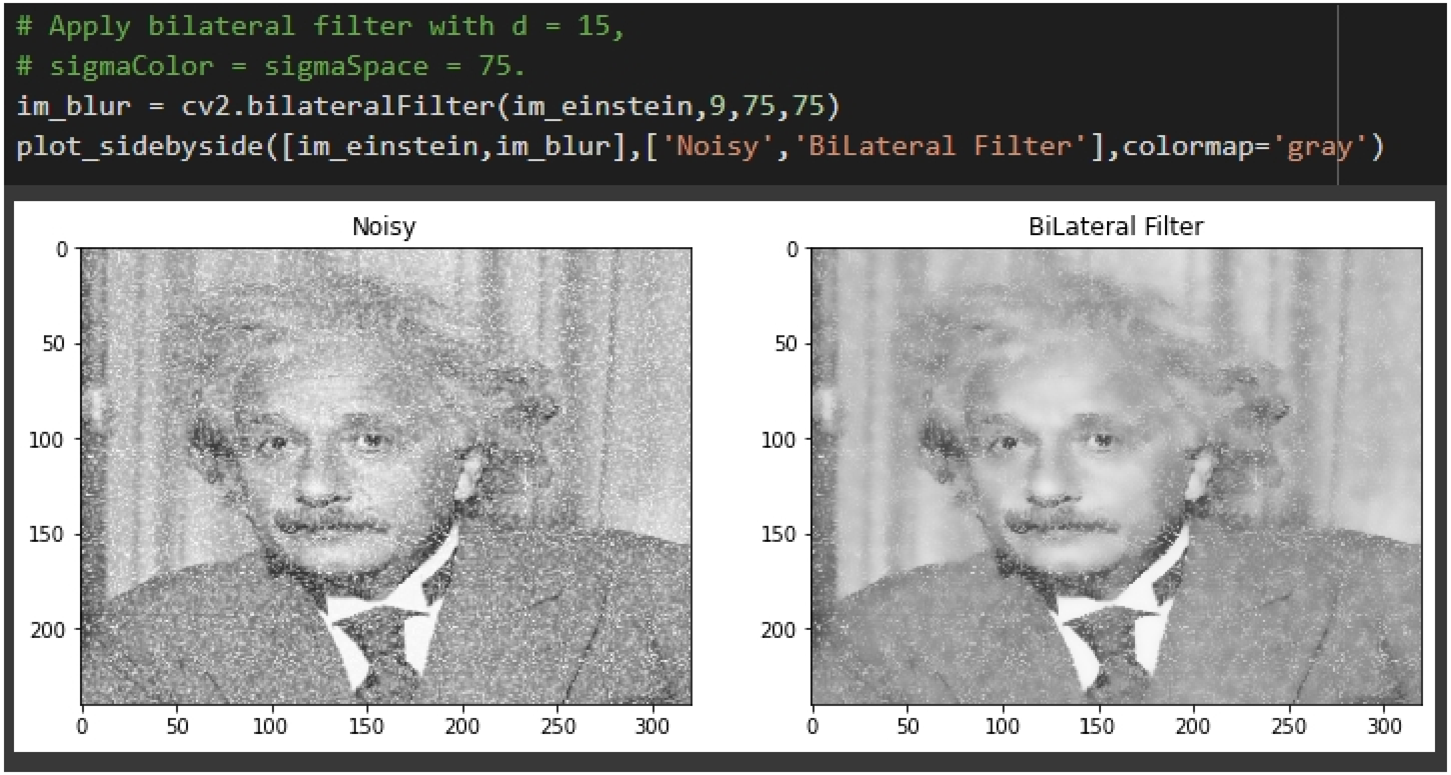
- **Gaussian Distribution based**
- **Add Normalization Factors and Range Weight**
- **Preserve details**



```
# 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')
```

Bilateral Filter

- **Gaussian Distribution based**
- **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')
```

Noisy

BiLateral Filter

Practice

Link: Practice 02