Computer Vision

CVI620

Session 7
01/2025

Noise

Definition: Anything that deviates from the ideal image or hinders achieving your imaging goal.

Noise Source

- Environmental factors
- Imaging device limitations
- Electrical interference
- Digitization process
- And more

Noise Characteristics

- Additive and random
- Represented as:

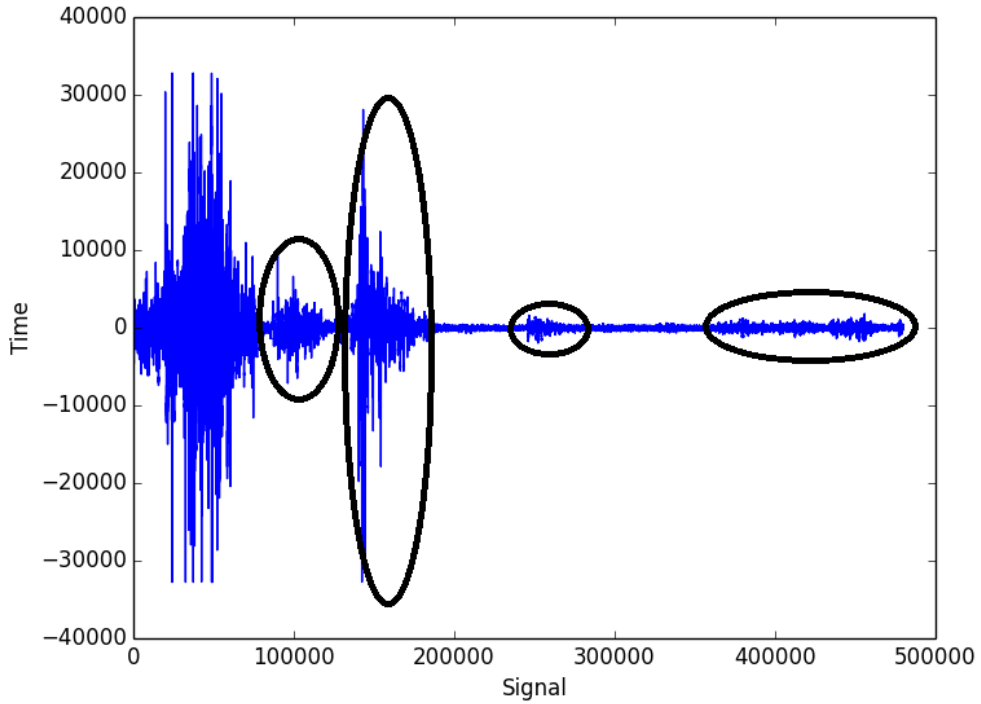
$$P(i, j) = I(i, j) + n(i, j)$$

P(i, j): Pixel value in the noisy image

I(i, j): Pixel value in the ideal image

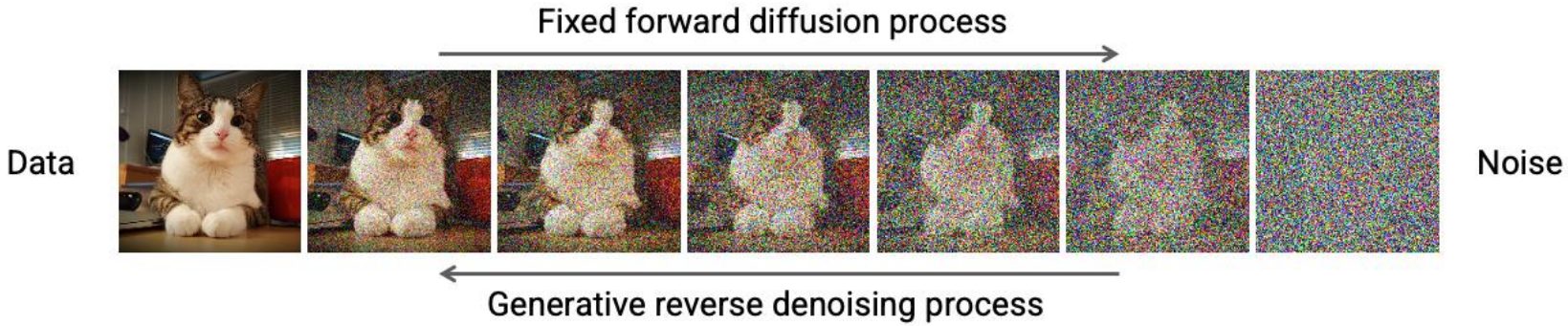
n(i, j): Noise value

Signal/Audio Noise



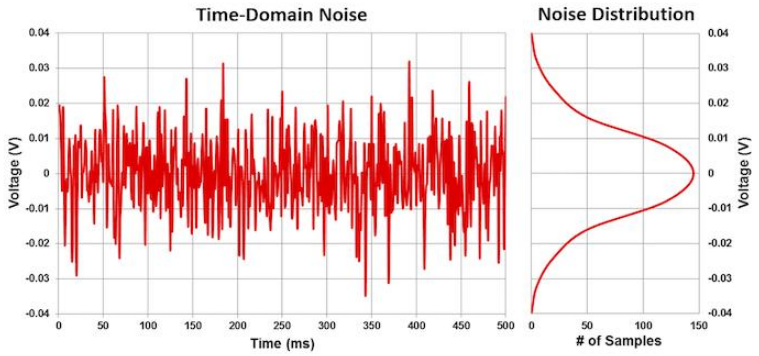
Why we need to study noise?

- Ensures image accuracy for analysis or display.
- Critical for applications in medical imaging, machine vision, and remote sensing.
- Used in state-of-art generative models like DALLE



Noise Types

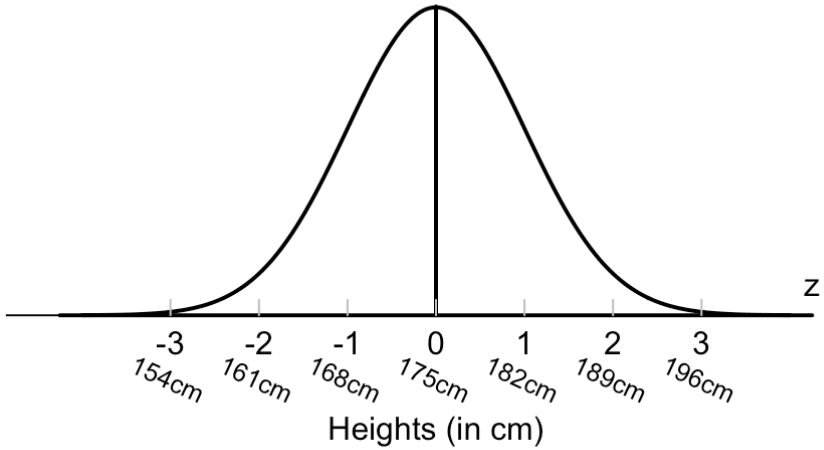
- Gaussian noise
- Salt and Pepper
- Poisson noise
- Speckle noise
- Thermal noise



Gaussian Noise

- Noise that follows a normal (Gaussian) distribution
- Normal distribution and why it is important:

  - Symmetry: Centered around the mean (μ).
  - Mean, Median, and Mode: All are equal and located at the peak.
  - Width determined by standard deviation (σ).



Gaussian Noise in CV2

- Choose random samples from normal distribution
- Add it to the image

```
image = cv2.imread('Lucy.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

mean = 0
std_dev = 25

gaussian_noise = np.random.normal(mean, std_dev, image.shape).astype('float32')

noisy_image = cv2.add(image.astype('float32'), gaussian_noise)

noisy_image = np.clip(noisy_image, 0, 255).astype('uint8')

plt.figure(figsize= (10, 5))
plt.subplot(1, 2, 1)
plt.imshow(image)

plt.subplot(1, 2, 2)
plt.imshow(noisy_image)

plt.show()
```

Impulsive Salt and Pepper Noise

- A type of impulse noise where random pixels in the image are replaced with:
  - White (salt): Maximum intensity (e.g. 255 in an 8-bit image)
  - Black (pepper): Minimum intensity (e.g. 0 in an 8-bit image)
- Appears as random white and black dots in an image

$$P(i,j) = \begin{cases} I(i,j), & \text{with probability } (1-p) \\ 0 \text{ (pepper)}, & \text{with probability } p/2 \\ 255 \text{ (salt)}, & \text{with probability } p/2 \end{cases}$$

Salt and Pepper

```python
import cv2
import numpy as np

image = cv2.imread('Lucy.jpg', cv2.IMREAD_GRAYSCALE)

def add_noise(img):
    row , col = img.shape

    number_of_pixels = np.random.randint(300, 10000)
    for i in range(number_of_pixels):
        y_coord = np.random.randint(0, row - 1)
        x_coord = np.random.randint(0, col - 1)
        img[y_coord, x_coord] = 255


    number_of_pixels = np.random.randint(300, 10000)
    for i in range(number_of_pixels):
        y_coord = np.random.randint(0, row - 1)
        x_coord = np.random.randint(0, col - 1)
        img[y_coord, x_coord] = 0

    return img

noisy_img = add_noise(image)
cv2.imshow('Noisy Image', noisy_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Question?

We have added noise so far.

How can we remove it?