Spatial Filtering

- **Image Smoothing** are two key techniques used in image processing to enhance the quality of an image by reducing noise or improving visual appearance. Both techniques involve manipulating the pixel values of an image based on its surrounding pixels.
- **1. Spatial Filtering**
- Spatial filtering is a technique used in image processing where a filter (or kernel) is applied to an image to modify pixel values based on their neighbors. It operates directly on the pixels in the image, meaning it processes each pixel one at a time and applies a mathematical operation to compute the new pixel value from its neighboring pixels.
- **Types of Spatial Filters:**
- **Linear Filters**:
  - These filters work by computing the weighted sum of the pixel values in the neighborhood. The most common linear filters are **averaging filters** and **Gaussian filters**.
  - **Example**: A **Gaussian filter** is commonly used for image smoothing.
- **Non-Linear Filters**:
  - These filters do not rely on weighted sums but instead use operations like **median** or **maximum** values from the neighborhood. These filters are often used for **noise removal**, especially for salt-and-pepper noise.
  - **Example**: A **median filter** is a non-linear filter commonly used to remove noise.

How Spatial Filters Work:

- A spatial filter is usually represented as a **kernel** (a small matrix, e.g., 3x3, 5x5) that slides over the image.
- For each pixel in the image, the kernel is placed over the pixel and its neighbors. The weighted sum of the pixel values in the kernel's area is then calculated and assigned to the central pixel.
- This process is repeated for every pixel in the image.
- **Example: Convolution Operation (Linear Filter)**
- If you apply a **3x3 averaging filter** (a type of linear filter), it would look like this:

  [1/9 1/9 1/9]
  [1/9 1/9 1/9]
  [1/9 1/9 1/9]
- Each pixel’s new value is computed by taking the average of itself and its eight neighboring pixels.

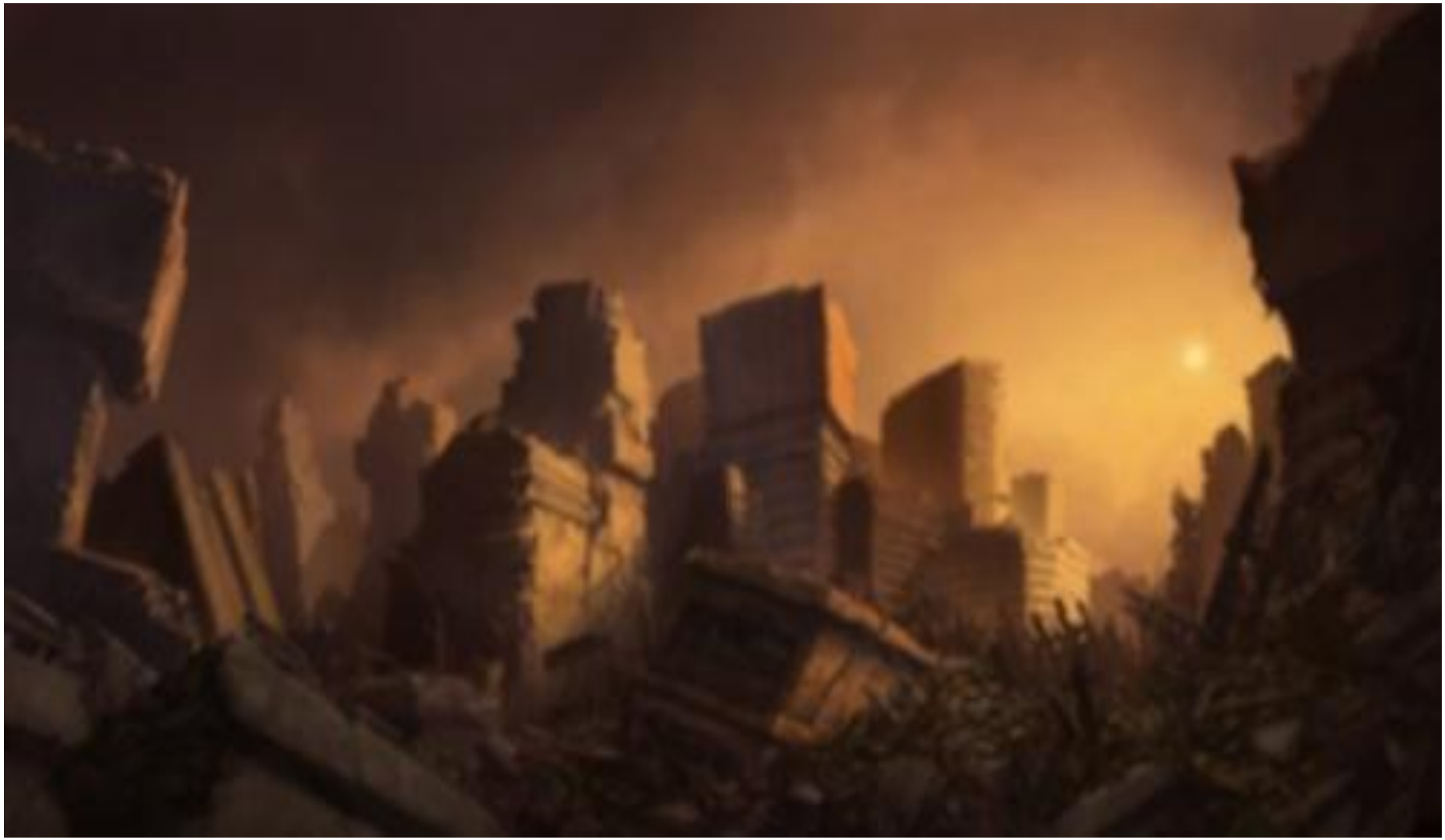
```python
import cv2
import numpy as np

# Read the image
image = cv2.imread("D://SJC//Image Processing//images//image1.jpg")

# Apply a 3x3 averaging filter
kernel = np.ones((3, 3), np.float32) / 9
filtered_image = cv2.filter2D(image, -1, kernel)

# Save or display the result
cv2.imshow('spatial_filtered_image.jpg', filtered_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Spatial image



Smoothing Linear Filters

- Smoothing, linear spatial filter computes the average of the pixels contained in the neighborhood of the filter mask.
- These filters sometimes are called *averaging filters and sometimes they are also referred as lowpass filters.*
- During such filtration, the value of every pixel in an image is replaced by the average of the intensity levels in the neighborhood defined by the filter mask.

Smoothing Linear Filters

- This results in an output image with reduced “sharp” transitions in intensities.
- Random noise typically consists of sharp transitions in intensity levels, hence smoothing linear filter is used for noise reduction.
- Averaging filters have the undesirable side effect that they blur edges.

Smoothing Linear Filters

- Figure shows 3 x 3 smoothing filter.
- Use of this filter yields the standard average of the pixels under the mask.
- Equation:

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i$$

which is the average of the intensity levels of the pixels in the 3 x 3 neighborhood defined by the mask.

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

Smoothing Linear Filters

- The coefficients of the filter are all 1s to make it computationally more efficient.
- At the end of the filtering process the entire image is divided by 9.
- An m x n mask would have a normalizing constant equal to $1/mn$.
- A spatial averaging filter in which all coefficients are equal is called a box filter.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Smoothing Linear Filters

- This mask uses *weighted average, (pixels are multiplied by* different coefficients i.e. giving more importance (weight) to some pixels at the expense of others).

- In the mask shown, the pixel at the center is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation.

- The other pixels are inversely weighted as a function of their distance from the center of the mask.

- The diagonal terms are further away from the center than the orthogonal neighbors and, thus, they are weighed less than the immediate neighbors of the center pixel.

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Smoothing Linear Filters

- Highest weighing given to the center and the value of the coefficients are decreased as a function of increasing distance is an attempt to reduce blurring in the smoothing process.

- The sum of all the coefficients in the mask of figure is equal to 16, an attractive feature for computer implementation because it is an integer power of 2.

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Smoothing Linear Filters

- With reference to earlier equation, the general implementation for filtering an M x N image with a weighted averaging filter of size m x n (*m and n odd) is* given by the expression

$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)}$$

Where, $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$

The denominator in equation is simply the sum of the mask coefficients and, therefore, it is a constant that needs to be computed only once.

2. Image Smoothing

- Image smoothing is a specific type of spatial filtering where the goal is to **reduce noise** or **blurring** in an image. This helps to smooth out variations in pixel intensity, which is especially useful for eliminating grainy noise or for visual effect.
- Smoothing works by replacing each pixel value with a value derived from its neighboring pixels, and the result is a "softer" or "smoother" version of the original image.
- **Common Image Smoothing Techniques:**
- **Averaging Filter (Mean Filter)**:
  - In this method, each pixel in the image is replaced by the **average value** of the pixels around it (within a specified window size, such as 3x3, 5x5).
  - **Purpose**: Reduces high-frequency noise (but also blurs edges).
- **Gaussian Blur**:
  - This method uses a **Gaussian function** to assign weights to the neighboring pixels. The weights are higher for pixels that are closer to the center and lower for those that are farther away. This results in a smoother image with better preservation of edges compared to the mean filter.
  - **Purpose**: Softens the image while preserving edge information more effectively than the averaging filter.

Cnt..

- **Median Filtering**:
  - The pixel value is replaced by the **median** of the pixel values in its neighborhood. Median filtering is particularly effective at removing salt-and-pepper noise while preserving edges.
  - **Purpose**: Excellent for removing impulse noise (like salt-and-pepper noise).
- **Bilateral Filter**:
  - A non-linear filter that preserves edges while smoothing the regions of uniform color. It does this by considering both the spatial distance between pixels and their color intensity differences.
  - **Purpose**: Edge-preserving smoothing. Often used for reducing noise while maintaining sharp boundaries.

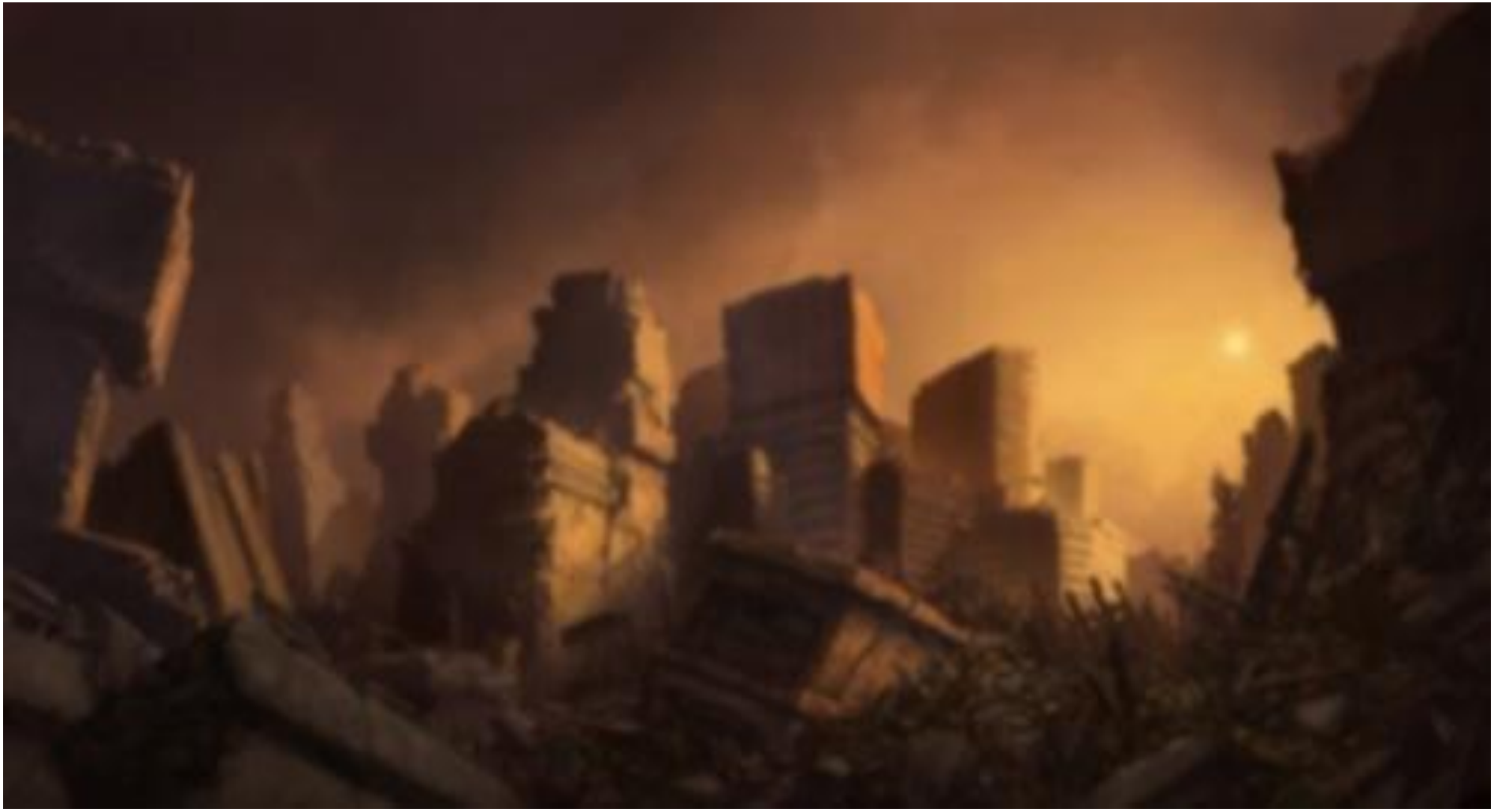
```python
import cv2

# Read the image
image = cv2.imread("D://SJC//Image Processing//images//image1.jpg")

# Apply Gaussian Blur
blurred_image = cv2.GaussianBlur(image, (5, 5), 0)  # (5, 5) is the kernel size

# Save or display the result
cv2.imwrite('smoothed_image.jpg', blurred_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Smoothing



Differences Between Spatial Filtering and Image Smoothing:

- **Spatial filtering** is a broader term that refers to applying any filter (linear or non-linear) to the image. It can be used for various purposes such as noise reduction, edge detection, and image enhancement.
- **Image smoothing** is a specific type of spatial filtering aimed at reducing noise and blurring the image to create a softer look or remove undesirable artifacts.

Common Use Cases for Image Smoothing:

- **Noise Removal**: Smoothing helps reduce various types of noise, such as **Gaussian noise** or **salt-and-pepper noise**.
- **Pre-processing for Edge Detection**: Before applying edge detection algorithms (like Sobel or Canny), smoothing the image helps in reducing noise and unwanted artifacts, leading to more accurate results.
- **Visual Enhancement**: In photography or film, smoothing can be used to create a blur effect, soften an image, or give it a more dreamy or artistic look.

Summary of Techniques:

Technique	Purpose	Common Use
Averaging Filter	Smooth out noise by averaging surrounding pixels	Basic smoothing, noise reduction
Gaussian Blur	Apply a weighted average to soften the image	Noise reduction, edge smoothing
Median Filter	Replace pixel with the median of neighbors	Salt-and-pepper noise removal
Bilateral Filter	Preserve edges while smoothing regions	Edge-preserving smoothing