**Lab-3: To perform Image Smoothing and Blurring using Python OpenCV**

Image smoothing and blurring are common techniques used in image processing to reduce noise and detail in an image. OpenCV provides several methods for this in Python.

* Image noise refers to random variations in brightness or color information within an image, often appearing as grainy or speckled patterns.

**✅ Common Methods in OpenCV for Smoothing/Blurring:**

1. **Averaging (Mean Filter)**
2. **Gaussian Blur**
3. **Median Blur**
4. **Bilateral Filter**

**📌 Required Library:**

pip install opencv-python

**✅ Example Code: Image Smoothing and Blurring in Python (Using OpenCV)**

**🔍 Explanation:**

| **Method** | **Description** | **Use Case** |
| --- | --- | --- |
| cv2.blur() | Averages pixels in a kernel area (simple smoothing) | Basic denoising |
| cv2.GaussianBlur() | Uses Gaussian kernel for weighted smoothing | Better for natural blurring |
| cv2.medianBlur() | Takes the median value of surrounding pixels | Good for removing salt-and-pepper noise |
| cv2.bilateralFilter() | Smooths image while keeping edges sharp | Best for face smoothing, edge-preserving |

1. **Gaussian Blur Example-**

import cv2

import numpy as np

# Load the image

image = cv2.imread('your\_image.jpg')

# Check if image loaded successfully

if image is None:

print("Error: Could not load image.")

else:

# Apply Gaussian blurring with a 5x5 kernel and sigmaX=0 (auto-calculated)

blurred\_gaussian = cv2.GaussianBlur(image, (5, 5), 0)

# Display the original and blurred images

cv2.imshow('Original Image', image)

cv2.imshow('Gaussian Blurred Image', blurred\_gaussian)

cv2.waitKey(0)

cv2.destroyAllWindows()