python programs to load image and apply smoothing linear spatial filters

import cv2

# Load image

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

# Apply Gaussian blur

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

# Display original and blurred images

cv2.imshow('Original Image', image)

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

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

# Load image

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

# Apply Median blur

blurred\_image = cv2.medianBlur(image, 5)

# Display original and blurred images

cv2.imshow('Original Image', image)

cv2.imshow('Blurred Image (Median)', blurred\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

# Load image

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

# Apply Bilateral filter

blurred\_image = cv2.bilateralFilter(image, 9, 75, 75)

# Display original and blurred images

cv2.imshow('Original Image', image)

cv2.imshow('Blurred Image (Bilateral)', blurred\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

Non linear spatial filter

import cv2

# Load the image

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

# Apply median filter

median\_filtered\_image = cv2.medianBlur(image, 5) # Adjust kernel size as needed

# Display the original and filtered images

cv2.imshow('Original Image', image)

cv2.imshow('Median Filtered Image', median\_filtered\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

Bilateral filter

import cv2

# Load the image

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

# Apply bilateral filter

bilateral\_filtered\_image = cv2.bilateralFilter(image, 9, 75, 75) # Adjust parameters as needed

# Display the original and filtered images

cv2.imshow('Original Image', image)

cv2.imshow('Bilateral Filtered Image', bilateral\_filtered\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Non-Local Means Denoising**

import cv2

# Load the image

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

# Apply non-local means denoising

denoised\_image = cv2.fastNlMeansDenoisingColored(image, None, 10, 10, 7, 21) # Adjust parameters as needed

# Display the original and denoised images

cv2.imshow('Original Image', image)

cv2.imshow('Denoised Image', denoised\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()