Python | Image blurring using OpenCV

Difficulty Level : Medium • Last Updated : 17 Apr, 2019

Image Blurring refers to making the image less clear or distinct. It is done with the help of various low pass filter kernels.

Advantages of blurring:

- It helps in Noise removal. As noise is considered as high pass signal so by the application of low pass filter kernel we restrict noise.
- It helps in smoothing the image.
- Low intensity edges are removed.
- It helps in hiding the details when necessary. For e.g. in many cases police deliberately want to hide the face of the victim, in such cases blurring is required.

Important types of blurring:

- **Gaussian Blurring:**Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. It is also used as a preprocessing stage before applying our machine learning or deep learning models.
  E.g. of a Gaussian kernel(3×3)

$$1/16 \quad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- **Median Blur:** The Median Filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. It is one of the best algorithms to remove Salt and pepper noise.
- **Bilateral Blur:** A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Thus, sharp edges are preserved while discarding the weak ones.

Below is the Python code:

```
# importing libraries
import cv2
import numpy as np

image = cv2.imread('C://Geeksforgeeks//image_processing//fruits.jpg')

cv2.imshow('Original Image', image)
cv2.waitKey(0)
```

```
cv2.waitKey(0)

# Median Blur
median = cv2.medianBlur(image, 5)
cv2.imshow('Median Blurring', median)
cv2.waitKey(0)


# Bilateral Blur
bilateral = cv2.bilateralFilter(image, 9, 75, 75)
cv2.imshow('Bilateral Blurring', bilateral)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:

Previous

Image Processing in Python (Scaling, Rotating, Shifting and Edge Detection)

Next

Erosion and Dilation of images using OpenCV in python

RECOMMENDED ARTICLES

Page : 1 2 3

01 **What is Image Blurring**
14, Oct 18

02 **Convert OpenCV image to PIL image in Python**
17, Dec 20

03 **OpenCV - Facial Landmarks and Face Detection using dlib and OpenCV**
18, May 20

04 **Transition from OpenCV 2 to OpenCV 3.x**
15, Aug 20

05 **Image resizing using Seam carving using OpenCV in Python**
17, Aug 20

06 **Python | Detect corner of an image using OpenCV**
15, Oct 18

07 **Image Pyramid using OpenCV | Python**
16, May 19

08 **Negative transformation of an image using Python and OpenCV**
04, Mar 20

Article Contributed By :

Sourabh_Sinha
@Sourabh_Sinha

Vote for difficulty

Current difficulty : Medium

Article Tags : Image-Processing, OpenCV, Python

Improve Article Report Issue