# Apply a Gauss filter to an image with Python

Last Updated : 29 Aug, 2024

A Gaussian Filter is a low pass filter used for reducing noise (high frequency components) and blurring regions of an image. The filter is implemented as an Odd sized Symmetric Kernel (DIP version of a Matrix) which is passed through each pixel of the Region of Interest to get the desired effect. The kernel is not hard towards drastic color changed (edges) due to it the pixels towards the center of the kernel having more weightage towards the final value then the periphery. A Gaussian Filter could be considered as an approximation of the Gaussian Function (mathematics). In this article we will learn methods of utilizing Gaussian Filter to reduce noise in images using Python programming language.

We would be using the following image for demonstration:
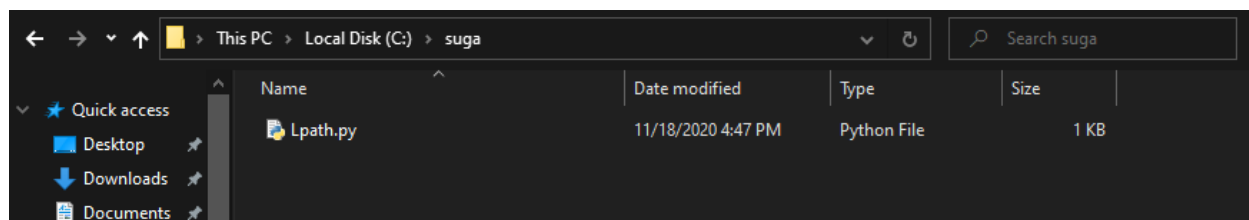
A screenshot of a segment of windows explorer

Process to Apply a Gauss filter

In the process of using Gaussian Filter on an image we firstly define the size of the Kernel/Matrix that would be used for demising the image. The sizes are generally odd numbers, i.e. the overall results can be computed on the central pixel. Also the Kernels are symmetric & therefore have the same number of rows and column. The values inside the kernel are computed by the Gaussian function, which is as follows:

$$G(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

2 Dimensional gaussian function

Where,

$x \rightarrow$ X coordinate value

$y \rightarrow$ Y coordinate value

$\pi \rightarrow$ Mathematical Constant PI (value = 3.13)

$\sigma \rightarrow$ Standard Deviation

Using the above function a gaussian kernel of any size can be calculated, by providing it with appropriate values. A 3×3 Gaussian Kernel Approximation(two-dimensional) with Standard Deviation = 1, appears as follows

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

Implementing the Gaussian kernel in Python

We would be using PIL (Python Imaging Library) function named *filter()* to pass our whole image through a predefined Gaussian kernel. The function help page is as follows:

Syntax: *Filter(Kernel)*

Takes in a kernel (predefined or custom) and each pixel of the image through it (Kernel Convolution).

Parameter: *Filter Kernel*

Return: Image Object

In the following example, we would be blurring the aforementioned image.
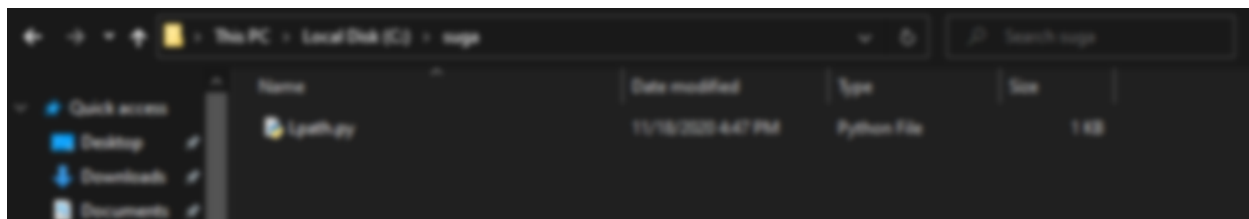
Python3

```python
# ImageFilter for using filter() function
from PIL import Image, ImageFilter

# Opening the image
# (R prefixed to string in order to deal with '\' in paths)
image = Image.open(r"IMAGE_PATH")

# Blurring image by sending the ImageFilter.
# GaussianBlur predefined kernel argument
image = image.filter(ImageFilter.GaussianBlur)

# Displaying the image
image.show()
```

Output:

Blurred Image

Explanation:

Firstly we imported the Image and ImageFilter (for using *filter()*) modules of the PIL library. Then we created an image object by opening the image at the path *IMAGE_PATH* (User defined). After which we filtered the image through the filter function, and providing *ImageFilter.GaussianBlur* (predefined in the ImageFilter module) as an argument to it. The kernel dimensions of ImageFilter.GaussianBlur is 5×5. In the end we displayed the image.

Note: The size of kernel could be manipulated by passing as parameter (optional) the radius of the kernel. This changes the following line from.

image = image.filter(ImageFilter.GaussianBlur)

to

image = image.filter(ImageFilter.GaussianBlur(radius=x))

where x => blur radius (size of kernel in one direction, from the center pixel)

Blurring a small region in an image:

Instead of the whole image, certain sections of it could also be selectively blurred. This could be performed by firstly cropping the desired region of the image, and then passing it through the filter() function. The output of which (the blurred sub image) would be pasted on top of the original image. This would give us the desired output.

The code for which is as follows:

Python3

```python
from PIL import Image, ImageFilter

image = Image.open(r"FILE_PATH")

# Cropping the image
smol_image = image.crop((0, 0, 150, 150))

# Blurring on the cropped image
blurred_image = smol_image.filter(ImageFilter.GaussianBlur)

# Pasting the blurred image on the original image
image.paste(blurred_image, (0,0))

# Displaying the image
image.save('output.png')
```
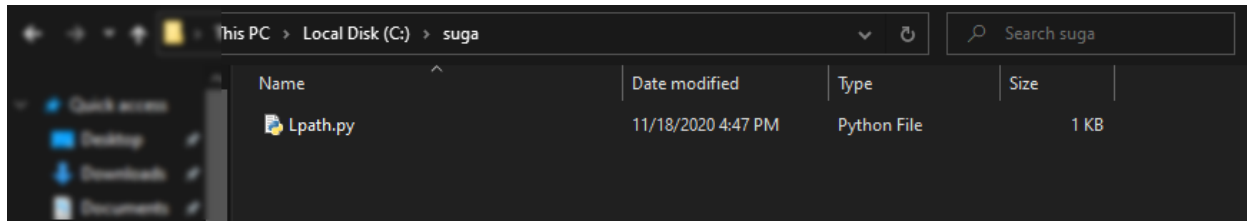
Output:

Only the top left region of the image blurred

💬 Comment   More info ⌄   Advertise with us

Next Article ›

Spatial Filtering and its Types

Similar Reads

Computer Vision Tutorial

Computer vision is a field of study within artificial intelligence (AI) that focuses on enabling computers to Intercept and extract information from images and videos, in a manner similar to human vision. It involves...

🕐 9 min read

Introduction to Computer Vision   ⌄

Image Processing & Transformation   ⌃

Digital Image Processing Basics

Digital Image Processing means processing digital image by means of a digital computer. We can also say that it is a use of computer algorithms, in order to get enhanced image either to extract some useful...

🕐 7 min read

Difference Between RGB, CMYK, HSV, and YIQ Color Models

The colour spaces in image processing aim to facilitate the specifications of colours in some standard way. Different types of colour models are used in multiple fields like in hardware, in multiple applications of...

🕐 3 min read

Image Enhancement Techniques using OpenCV - Python

Image enhancement is the process of improving the quality and appearance of an image. It can be used to correct flaws or defects in an image, or to simply make an image more visually appealing. Image...

🕐 15+ min read

Image Transformations using OpenCV in Python

In this tutorial, we are going to learn Image Transformation using the OpenCV module in Python. What is Image Transformation? Image Transformation involves the transformation of image data in order to retrieve...

🕐 5 min read

How to find the Fourier Transform of an image using OpenCV Python?

The Fourier Transform is a mathematical tool used to decompose a signal into its frequency components. In the case of image processing, the Fourier Transform can be used to analyze the frequency content of an...

🕐 5 min read

Python | Intensity Transformation Operations on Images

Intensity transformations are applied on images for contrast manipulation or image thresholding. These are in the spatial domain, i.e. they are performed directly on the pixels of the image at hand, as opposed to bein...

🕐 5 min read

Histogram Equalization in Digital Image Processing

A digital image is a two-dimensional matrix of two spatial coordinates, with each cell specifying the intensity level of the image at that point. So, we have an N x N matrix with integer values ranging from a minimum...

🕐 6 min read

Python - Color Inversion using Pillow

Color Inversion (Image Negative) is the method of inverting pixel values of an image. Image inversion does not depend on the color mode of the image, i.e. inversion works on channel level. When inversion is used on...

🕐 4 min read

Image Sharpening Using Laplacian Filter and High Boost Filtering in MATLAB

Image sharpening is an effect applied to digital images to give them a sharper appearance. Sharpening enhances the definition of edges in an image. The dull images are those which are poor at the edges. There ...

🕐 4 min read

Wand sharpen() function - Python

The sharpen() function is an inbuilt function in the Python Wand ImageMagick library which is used to sharpen the image. Syntax: sharpen(radius, sigma) Parameters: This function accepts four parameters as...

🕐 2 min read

Python OpenCV - Smoothing and Blurring

In this article, we are going to learn about smoothing and blurring with python-OpenCV. When we are dealing with images at some points the images will be crisper and sharper which we need to smoothen or...

🕐 7 min read

Python PIL | GaussianBlur() method

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The ImageFilter module contains definitions for a pre-defined set of filters, which can be used with the...

🕐 1 min read

Apply a Gauss filter to an image with Python

A Gaussian Filter is a low pass filter used for reducing noise (high frequency components) and blurring regions of an image. The filter is implemented as an Odd sized Symmetric Kernel (DIP version of a Matrix)...

🕐 4 min read

Spatial Filtering and its Types

Spatial Filtering technique is used directly on pixels of an image. Mask is usually considered to be added in size so that it has specific center pixel. This mask is moved on the image such that the center of the mask...

🕐 3 min read

Python PIL | MedianFilter() and ModeFilter() method

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The ImageFilter module contains definitions for a pre-defined set of filters, which can be used with the...

🕐 1 min read

Python | Bilateral Filtering

A bilateral filter is used for smoothening images and reducing noise, while preserving edges. This article explains an approach using the averaging filter, while this article provides one using a median filter....

🕐 2 min read

Python OpenCV - Morphological Operations

Python OpenCV Morphological operations are one of the Image processing techniques that processes image based on shape. This processing strategy is usually performed on binary images. Morphological operations...

🕐 7 min read

Erosion and Dilation of images using OpenCV in python

Morphological operations are a set of operations that process images based on shapes. They apply a structuring element to an input image and generate an output image. The most basic morphological...

🕐 2 min read

Introduction to Resampling methods

While reading about Machine Learning and Data Science we often come across a term called Imbalanced Class Distribution, which generally happens when observations in one of the classes are much higher or...

🕐 8 min read

Python | Image Registration using OpenCV

Image registration is a digital image processing technique that helps us align different images of the same scene. For instance, one may click the picture of a book from various angles. Below are a few instances that...

🕐 3 min read

Feature Extraction and Description   ⌄

Deep Learning for Computer Vision   ⌄

Object Detection and Recognition   ⌄

Image Segmentation   ⌄

3D Reconstruction   ⌄

50+ Top Computer Vision Projects [2025 Updated]

Computer Vision is a field of Artificial Intelligence (AI) that focuses on interpreting and extracting information from images and videos using various techniques. It is an emerging and evolving field within AI. Computer...

🕐 6 min read

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305

Advertise with us

Company
About Us
Legal
Privacy Policy
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA
Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies
HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS

Python Tutorial
Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

Bootstrap

Web Design

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects