

## Image filtration in Python using openCV

Sahil Singh <sup>a</sup> Ritu Verma <sup>b</sup> and Ankit kumar Singh<sup>c</sup>

<sup>a,b,c</sup>Galgotias University Computer science and engineering, Gautam buddha nagar, India

<sup>a</sup>sahil91098singh@gmail.com, <sup>b</sup>tanuvermaonmail@gmail.com, <sup>c</sup>onlineankit1999@gmail.com

**Article History:** Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

**Abstract:** Image processing is an area consisting of different phases for manipulating pixels of an image. Image processing is an enhancement of images which affects an image's pictorial view. Image processing is the process of converting an image into digital form and then performing various operations on it, such as enhancing the image or extracting useful information. Image filtering is a technique for altering the scale, shape, color, depth, smoothness, and other aspects of images. Essentially, it modifies the pixels of an image to transform it into the desired form using various types of graphical editing methods via graphic design and editing software. In Image Improvement, the removal of noise is very necessary and filters are used for this. This paper introduces some filters that will give you a better version of your image. Using different kinds of filters provides you noise free images.

### 1. Introduction

Image filtering is a step in the image preprocessing process. The recovery of distinct signals from the correspondence channel with no noise is a difficult task. For the removal of noises from a digital signal, a variety of denoising methods have been suggested. Wavelet denoising with edge calculation is a useful technique for reducing noise in digital signals. Images and videos during the acquisition and transmission phase, impulse noises are often distorted. This impulse noise appears in a picture as a result of bit errors in transmission or as a result of signal acquisition errors. When it comes to detecting edges and contours, noise has a major effect on detection accuracy. As a result, noise must be removed and the pixel's intensity must be regulated. Image filtering removes noise and unwanted features from an image, resulting in a better and more enhanced version. In this paper, we are trying to solve a problem with the image filters which are used to increase or decrease noises present in images, also dealing with hue(s). There are different kinds of image noise, we are providing some filters that will help to reduce/increase the noise based on user requirements. So, why are we using OpenCV? OpenCV is a library designed to solve a wide number of tasks relating to computer vision. OpenCV supports a wide range of object-oriented languages, including Python, C++, and Java, and it is available on a variety of platforms, including iOS, Android, OS X, Linux, and Windows. Interfaces based on OpenCL and CUDA are also being dynamically developed for rapid Graphics Processing Unit functions. The OpenCV Python API is referred to as OpenCV-Python. It combines the Python programming language with the OpenCV C++ API. Furthermore, using Numpy to help with the assignment makes it easier to complete. Numpy is an advanced standard library for numerical activities. It is filled with several basic and advanced features, very simple to pick up and available for many languages of programming. It will be easy for us to provide different kinds of filters. In this we are dealing with some noises like salt and pepper noise, speckle noise. We can also enhance the edges of an image to detect the feature of that image. Mean filter, median filter, Gaussian filter and bilateral filter are some of the well-known filters.

### 2. Problem Statement

When we capture a photo through the camera, we might not get a clear or we can say noise free image. As a consequence of image capture, image noise is an unavoidable side-effect, more simply understood as inaudible, yet inevitable fluctuations. In a digital camera, it can produce image noise if the light that reaches the lens is misaligned with the sensors. Even if noise in an image is not so clearly noticeable, there is bound to be some kind of image noise. Any sort of electronic system receives some noise and transmits it and sends it on to what it makes. They are distorted with impulse noise due to noisy channels as the images are broadcast over channels. Thus, filters are required before processing to eliminate noises. We have plenty of filters to minimize noise.

#### OpenCV convolutions and Kernels

A convolution is the transformation and image process by breaking it into small parts called windows and applying a kernel operator for each component. Typically, the kernel is a set, small 2D array containing numbers. The middle, which is called the anchor point, is an essential part of the kernel.

0	-1	-1
---	----	----

1	0	-1
1	1	0

Kernel Example

By following these steps, a convolution is executed:

- Place the kernel at the top of an image, with the anchor point of the kernel at the top of a predefined pixel.
- Make a multiplication between the kernel numbers and the kernel- overlapping pixel values, count the results of the multiplication, and put the result on the pixel below the anchor point.
- Repeat the process for any possible image location by sliding the kernel on the top of the image.

### Image Processing Operations: -

Several basic image processing techniques will be discussed. Picture enhancement, reconstruction, coding, and comprehension are four common application areas for digital image processing. The aim of these techniques is typically to make the processed image look better than the original.

#### 1. Noise: -

Noise in image processing is a random variance in image intensity that appears as grains in the image. It may appear in the image as a result of fundamental physics, such as the photon nature of light or the thermal energy of heat inside the image sensors. Basically, noise occurs when the pixels in an image display different intensity values rather than the true pixel values derived from the image. Information theory, specifically the Nyquist–Shannon sampling theorem, sets the mathematical limits for noise removal.

The following are a few examples of different forms of digital image noise:

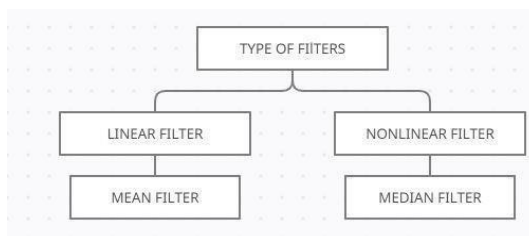
**1.1 Salt and pepper noise:** - Salt and pepper noise is a form of impulse noise that is generated by sprinkling random bright (255-pixel value) and random dark (0-pixel value) pixels in an image. Simply put, it's a random arrangement of black and white pixels on an image. It is caused by a sensor cell loss in the camera.

**1.2 Speckle noise:** - It's a kind of multiplicative noise that's popular in ultrasound medical images. Agranular noise is inherently present in Active Radar and Synthetic Aperture Radar (SAR) images, degrading their accuracy. It raises a local area's mean grey level.

**1.3 Gaussian Noise:** - Gaussian noise, also known as Gaussian Distribution, is a statistical noise with a probability density function equal to the normal distribution. Since it exists in amplifiers and detectors, it is also known as electronic noise.



**2. Image restoration:** - In image processing, image restoration is used to perform a variety of useful tasks. Noise reduction is one of the most critical activities. A filter is applied to an image to reduce the amount of unwanted noise and to enhance the edges of the image.



Types of filters: -

1. **Linear filter:** - To obtain the filtered image, linear filtering is performed by convolving the input with a filter function. A linear filter can be written as follows:

$$f(x, y) = \sum I(m, n) \cdot w(x - m, y - n)$$

In this case, I stand for the original image, f stands for the filtered pixel value, and w stands for the filter coefficients.

**1.1 Mean filter-** As the mean filter, the easiest filter to implement is known. Average smoothing is performed on an image by the mean filter. The name perfectly describes this filter's function. The average of the pixels that surround it is replaced by each pixel in the image. Noise, in particular, is blended into the rest of the picture. It is appropriate for a filter that performs average smoothing using a kernel for all non-negative entities. For example, if kernel A uses  $m(\text{size}) = 3$ .

$$A_{\text{avg}} = \frac{1}{3} [1 \ 1 \ 1]$$

Let I be an image of size N, let m be an odd number less than N, and let A be a linear filter kernel, that is, a mask of size m. Additionally, it is completely important to provide a sum of one for all entries in the kernel. The kernel must be divided by the number of the entries (hence the multiplication of (1/3) if the sum is not equal to one. If the requirement is not met, then the filtered image, together with the stated filtering effect, will become brighter than the original image. The second portion of the image filtering target A fulfils this constraint on the mean filter. Since averaging eliminates small differences, this filter is efficient at attenuating noise.

OpenCV function implementation: -

```
img = cv2.imread('name-of-img.jpg') # Load image
```

**blur = cv.blur(img,(5,5)) # Add mean filter to image**

2. **Nonlinear filter:** - One type of nonlinear filter works by ranking the pixels in a given neighborhood. It substitutes a value corresponding to the chosen rank for the target pixel. In the neighborhood pixels, the rank could be the highest, lowest, or middle value. It chose the value in the middle of the neighborhood of the target pixel.

**2.1 Median filter:** - The median filter overcomes one of the mean filter's drawbacks, namely its inability to minimize impulsive noise. The median filters are also useful for preserving image edges while reducing image noise. The median filter works by sorting pixel values within a neighborhood, selecting the median value from that ordered list, and replacing the original pixel in the output image with that median value.

$$f(x, y) = \text{median} \{I(m, n) | (m, n) \in W\}$$

Inside a window  $W$  of size  $m \times n$ , where the operation takes place,  $I$ = the noisy image,  $f$ = the restored image, and  $(m, n)$  = the row and column coordinates, respectively.

Instead of using an averaging operation, the median filter simply replaces each pixel with the middle value in the pixel's neighborhood, as opposed to the mean filtering technique. The median filter does not generate a new pixel value because it is one of the pixel values in the input image. This helps in the preservation of edges and demonstrates why these filters are useful in image processing.

OpenCV function implementation: -

**img = cv2.imread('name-of-img.jpg') # Load image**

**img\_median = cv2.medianBlur(img, 5) # Add median filter to image**

**3. Image Enhancement:** - Image enhancement techniques such as edge enhancement or image sharpening may be used instead of image smoothing for image restoration. If noise reduction is a low pass filtering application, image sharpening is a high pass filtering application. Any enhancement technique that highlights edges and fine details in an image is referred to as image sharpening. High pass filters are used to sharpen and improve the contrast of images because edges include the high frequency components. In printing and photography, image sharpening is commonly used. Image edge enhancement improves the appearance of the edges, making an image with highlighted edges more appealing than the original. The aim of image edge enhancement is to increase the processed image's visual appeal as compared to the unprocessed image.

**Canny edge detection:** - A four-step detection procedure is used to create the canny edge detector. The four steps are: -

1. Noise Reduction — 5x5 Gaussian filter
2. Using a calculator to calculate gradients — Finding the Image's Intensity Gradient
3. There is no maximum suppression— the upper threshold
4. Thresholding with hysteresis — upper/lower threshold

All of the above is combined into a single function, **cv.Canny()**, in OpenCV. Our input image is the first statement. Our `minVal` and `maxVal` arguments are the second and third arguments, respectively. Aperture size is the third statement. It refers to the size of the Sobel kernel that is used to locate image gradients. It is set to 3 by default. The last statement is `L2gradient`, which defines the gradient magnitude equation.

OpenCV function implementation: -

**img = cv2.imread('name-of-img.jpg') # Load image**

**canny=cv2.Canny(img,20,170) # Added canny edge detection filter**



### 3. Literature review

When a camera or other imaging system acquires an image, it is unable to explicitly use the vision system for which it is intended. The random changes in intensity, changes in lighting, or poor contrast that must be dealt with in the early stages of the picture may be damaging processing Vision. Images in Computer Vision are defined as number matrices representing the discrete values of color or intensity present in each pixel of the image. Each image is known to be displayable in several ways as input data, such as arrays of pixel values or multidimensional plots representing the pixel intensity distribution. In the image, a pixel is just a single point and is stored as a number within the  $[0, 256]$  range. The RGB model stands for Red-Green-Blue and informs us that we store the red, green and blue intensity as a number from 0 to 256 for every pixel. Images are transformed into multi-dimensional arrays in OpenCV, which simplifies their manipulation considerably. Image Filtering uses 2D Convolutions with different low and high pass filters to eliminate noise, blur images, etc. By modifying the intensity values of pixels, it is normal to enhance images. Most image processing software tools have many options for modifying the appearance of an image by converting the pixels into a new output value through a single feature that maps the input grey value. This can be easily generalized so that a user can indicate several different image regions and apply to each one a separate mapping function. Remapping the gray values is also called stretching, since stretching the gray values of an image that is too dark to the maximum set of usable gray values is normal. Sometimes, the elimination of small regions from an image is useful. A small region may just be the product of noise, or it may reflect low-level information that should be omitted from the construction of the image definition. You may delete small regions by modifying single pixels or by removing components after extracting related components. By computing intensity variations in local image regions, image points of high contrast can be observed. These points usually form the boundary between different objects or components of the scene. We illustrate how to do this using neighborhood templates or masks in this section. We begin with the use of one-dimensional signals: this helps to improve intuition and formalism, and is very important in its own right as well. The 1D signal could be just rows of a 2D image or columns. All filters are only used for getting a better image without any noise. Images have a RGB system if we change any of the color range the whole image would be changed. There are so many filters like linear, nonlinear, mean filter, frequency filter, Gaussian smoothing etc.

### 4. Working examples

#### 1. Simple thresholding

If the pixel value is greater than the threshold value, one value (white may be the value) is assigned, otherwise another value is assigned (may be black). The `cv2.threshold` function is the function used. The source image, which should be a grayscale image, is the first statement. The second statement is the threshold value that is used for pixel values to be graded. The third statement is `maxVal`, which is the value to be given if the pixel value is greater than the threshold value (sometimes less than that value). In thresholding we have binary thresholding.

OpenCV function implementation: -

```
img = cv2.imread('name-of-img.jpg')
```

**(This statement will load the image )**

```
ret,bw=cv2.threshold(img,127,255,
```

```
cv2.THRESH_BINARY)
```

(Through thresholding we will try to provide the value through which we can put below the particular value we assign the value 0 and above it will be white.)



## 2. Greyscaling

There is one simple way of gray scaling an image.

- 1) Determine a pixel's red, green, and blue values.
- 2) Make use of some fancy math to convert those numbers into a single gray value.
- 3) Substitute the current gray value for the original red, green, and blue values.
- 4) When explaining grayscale algorithms,

We'll concentrate on step 2 - using math to convert color values to grayscale values. So, if you come across a formula like this:

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3$$

For Each Pixel in Image {

Red = Pixel.Red

Green = Pixel.Green

Blue = Pixel.Blue

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3$$

Pixel.Red = Gray

Pixel.Green = Gray

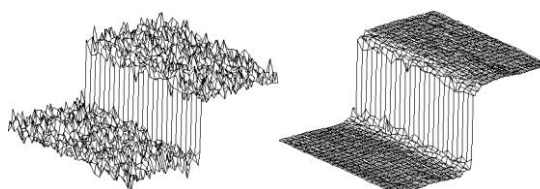
Pixel.Blue = Gray

}



## 3. Smoothing image

Picture smoothing is a primary image enhancement technology that can eliminate noise from images. As a result, it is a required functional module in various image-processing applications. A good smoothing algorithm can remove noise while preserving data. We use bilateral filter for smoothing the image. A bilateral filter is a non-linear, edge-preserving, noise-reducing image smoothing filter. It replaces each pixel's intensity with a weighted average of intensity values from neighboring pixels. A Gaussian distribution can be used to calculate this weight.





`bilateral=cv2.bilateralFilter(img,7,20,20)` #9 ,75 and 75 are sigma color value and sigma space value affects coordinates space and color space

```
cv2.imshow("bilateral",bilateral)
```



#### 4. Blurring the image

In order to filter a blur, we consider a rectangular group of pixels surrounding the pixel. This group of pixels, known as the kernel, goes along with the pixel being filtered. So that the filter pixel is always in the middle of the kernel, the kernel's width and height must be odd. The kernel in the preceding example is square and has a dimension of seven pixels. A weighted average of the color values of the pixels in the kernel is determined before applying this filter to the current pixel. In a Gaussian blur, the pixels closest to the kernel's core are given more weight than those farther away. This averaging is performed channel by channel, and the average channel values become the filtered pixel's new value. Because larger kernels have more values factored into the average, they will blur the image more than smaller kernels.

Open cv function implementation:-

```
gaussian=cv2.GaussianBlur(img,(7,7),0)
```

```
cv2.imshow("GaussianBlur",gaussian)
```



#### 5 Image saturation

Saturation is the level of shading or immersion of a tone. All grays from white dark to white and dark have an immersion of nothing. The higher the saturation, the bigger the part containing a specific tone. Indeed, it very well may be perceived that a strong shading mixes some dark or dark or white. Since the dim contains three segments of rgb, it causes the first rgb strong shading 100% extent to start to decrease. As indicated by the measure of dim added, the increment of the other two shading segments further lessens the immersion of the first strong tone.

Open cv function implementation:-

```
img_HSV=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
```

```
cv2.imshow("saturation",img_HSV[:, :,1])
```



#### 5. Summary

In this paper, as stated in the literature, we presented various noises and filtering techniques. Each filter has properties of its own. The appropriate filter is chosen based on the kind of noise and the amount of filtering necessary. Due to its edge preserving existence, filtering techniques are the most acceptable. Finally, various filters have different strengths and weaknesses. As a result, when solving image processing problems, various

types of filters are required. A nonlinear filter may perform better than a linear filter in some cases. In image restoration, the nonlinear filter outperforms the linear filter..

### References

1. Face Detection and Recognition using OpenCV, Article.
2. Image filters in python <https://towardsdatascience.com/image-filters-in-python-26ee938e57d2>
3. Facial Recognition using OpenCV
4. Image Enhancement on OpenCV based on the Tools: Python 2.7 February 2
5. Sweety Deswal, Shailender Gupta and Bharat Bhushan". A Survey of Various Bilateral Filtering Techniques" International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.8, No.3 (2015).
6. Image Filtering Techniques-A Review Ramanjeet Kaur 1, Er. Ram Singh 21 Student, Department of Computer Engineering, Punjabi University Patiala, Punjab, (India) 2 Assistant Professor, Department of Computer Engineering, Punjabi University Patiala, Punjab, (India)
7. MelikaMostaghim, Elnaz Ghodousi and Farshad "Image Smoothing Using Non- Linear Filters A Comparative Study" IEEE 2014
8. D. Z. P. F. L. I. Abdalla Mohamed Hambal, "Image Noise Reduction and Filtering Techniques, "International Journal of Science and Research, 2015
9. Image Enhancement on OpenCV based on the Tools: Python 2.7 V.Ravi1, Ch.Rajendra Prasad2, S.Sanjay Kumar3, P.Ramchandrar Rao4 Dept. of ECE, S R Engineering College, Warangal, Telangana, India.