

Image Enhancement Based on Opencv Using Python 2.7 – Review

A. P. Athane¹, Dr. S. R. Prasad²

Assistant Professor, Department of Textile, DKTE's TEI, Ichalkaranji, Maharashtra, India¹

Assistant Professor, Department of Textile, DKTE's TEI, Ichalkaranji, Maharashtra, India²

Abstract: In the vast area of digital image processing and high resolute transmission approach, this paper reviews the image enhancement techniques which are based OpenCV. OpenCV is the latest emerging approach in the area of image processing. There are different stages of image processing. The coding of all approaches is done by Python 2.7. [1] This paper gave you a basic understanding of image processing by looking at basic terms, then for each part, you'll saw the corresponding implementation of OpenCV, which is a library used for image processing.[4] This paper includes five most significant use cases, where the OpenCV library plays a key role. Also reviews different approaches such as Gamma transform, types of contrast enhancements, geometric transformation of image

Keywords OpenCV, image enhancements, gamma transform, contrast etc.

I. INTRODUCTION

OpenCV is used as an image processing library in many computer vision real-time applications. Digital image processing is a process which formulates the original image through various technologies. It applies many operations on an input image which result matches the desired image. These processing approaches implement some set of process which result an enhanced image or extract some useful data from the image. Image processing and enhancement techniques nowadays adapted mostly in industries like 3D mapping, object movement detection, radar targeting, color processing, face detection, restorations, and many others fields. All the above applications involve research and processing and implanting different filtering technologies to get the desired results. The objective of image enhancement is to restore the image that is distorted during the transmission from one form to another form. This paper reviews different approaches, all these approaches are processed through Open CV Library. First there are different stages of image processing also other simple openCV in image processing and Gamma transform techniques. This paper also includes image contrast and contrast enhancement algorithms and geometric transformation of image.. [1]

What is image processing and why do we need it?

When we look at the image processing or computer vision fields, the main purpose is to give the eye functions to machines. From this perspective, it's possible to state that image processing is the conversion of the human visual system (HVS) to digital images. To get the most realistic output from the digital images, we need to do some processing on them. There are various application areas where image processing is being used. One of the most common use cases is image enhancements, that is, as we defined image processing to improve the quality of the image to gain more information about it from 380 – 700 nanometers. This range is called visible light.[2]

II. STAGES OF IMAGE PROCESSING

OpenCV is an open source library for image processing and computer vision that was developed by Intel. OpenCV applications run on Windows™, Android, Linux®, Mac, and iOS systems because it is a platform-independent library. It's commonly used with Python and C++ languages, but we can develop applications by using it with the Java™ language, Perl, and Ruby. OpenCV is great for working in this field because it handles the workload of dealing with the implementation of mathematical formulas. Both Image Processing algorithms and Computer Vision (CV) algorithms take an image as input; however, in image processing, the output is also an image, whereas in computer vision the output can be some features/information about the image. [1].

What is the OpenCV Library?

To begin with, the OpenCV library is an open-source (hence its full name: Open Source Computer Vision and machine learning software library. [2]

.OpenCV is used as an image processing library in many computer vision real-time applications. There are thousands of functions available in OpenCV. These simple techniques are used to shape our images in our required format. As we know an image is a combination of pixels, for a color image we have three channels with pixels ranging from 0 to 255,

and for black & white-colored images has only one change ranging from 0 to 1. In this paper, we will demonstrate some important image processing steps and we will do modifications to an image by manipulating its pixels using OpenCV functions.

Converting a Color Image to a Grayscale Image:

This is the basic technique that is used to convert the color image into grey shades. A color image consists of 3 channel depth while using grayscaling it reduces the depth of the image to 1 channel. It reduces model complexity. For many applications like edge detection, photo sketch, cartooning image we use grayscale converted images.

In the below code snippet we are converting a color image to a grayscale image.

```
import cv2
image = cv2.imread('/content/Screenshot (14).png')
grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2_imshow(grayImage)
```

In the above output, we converted the colored image to grayscale.

Edge detection:

In edge detection, it gives an outline of the image by extracting edges from the image. We are using a canny filter to perform this task. Based on the threshold values, a canny filter detects the edges. The higher thresholds give cleaner images compared to lower thresholds gives a clumsy output.

```
import cv2
import numpy as np
img = cv2.imread('/content/Screenshot (14).png',0)
edge_det = cv2.Canny(img,100,200)
cv2_imshow(edge_det)
```

Here we had given a threshold value 100,200 and you can try with different threshold values. This edge detection is used for segmentation, restoration, picture enhancement, pattern recognition.

Smoothing Techniques:

The image smoothing technique is performed using a filter. By convolving the image, it reduces the noise in the image by adding a blurring effect on the edges. There are different types of smoothing techniques we perform depending on the input image.

1) Average using cv2.blur()

In this technique, depending on the size of the filter convolution is performed but it averages the pixels under the filter and assigns the value to the centre pixel under the filter.

```
2) import cv2
3) import numpy as np
4) from google.colab.patches import cv2_imshow
5) img = cv2.imread('/content/download (14).jpg')
6) blur = cv2.blur(img,(5,5))
7) cv2_imshow(blur)
8) cv2_imshow(img)
```

Morphological Techniques

These are the techniques used to manipulate the image sizes, these techniques are used only after converting the image into grayscale. These techniques are used for removing noise, correcting the imperfections in the data, and to make clear images. Erosion and Dilation are mostly used in morphological techniques.

Erosion:

This technique removes the boundary pixels from the input by just passing the filter on the image. Depending on the size of the kernel, it removes the boundary pixels from the input image. It performs similar to soil erosion so they named this technique as erosion.

```
img = cv2.imread('/content/Screenshot (14).png')
kernel = np.ones((5,5),np.uint8)
img_erosion = cv2.erode(img, kernel, iterations=1)
img_dilation = cv2.dilate(img, kernel, iterations=1)
cv2_imshow(img)
cv2_imshow(img_erosion)
```

Dilation:

The above technique removes the boundary pixels but in Dilation. It adds the additional pixels to the input. It is used when the pixels are missing in the image. In general practices, we apply erosion to shrink the image to remove noises, and then by applying the dilation, there will be no loss of pixels.

```
img = cv2.imread('/content/Screenshot (14).png')  
kernel = np.ones((5,5),np.uint8)  
img_dilation = cv2.dilate(img, kernel, iterations=1)  
cv2_imshow(img)  
cv2_imshow(img_dilation). [10]
```

Examples Of Using OpenCV In Image Processing

Five most significant use cases, where the OpenCV library plays a key role.

Image Enhancement:

Depending on the use case, there are various methods available within OpenCV, which could be applied to enhance your images. For instance, the OpenCV algorithms can help you in

Histogram equalization. It's a technique used to improve contrast in images. This method usually increases the contrast level in images, particularly when a picture has many areas of lower local contrast, and you want them to gain a higher contrast in order to make them more readable.

Image noise reduction. Generally speaking, image noise is a random variation of brightness or color information in images. In most cases, not only is it useless, but it also deteriorates the image. It originates from the camera itself and usually happens when pictures are taken in dark places, and not enough light reaches the object-glass. The OpenCV library has appropriate algorithms to help you with the image denoising.

Information Extraction:

Sometimes, you need to extract some information from your image, for instance, available camera functions, flash use, aperture, zoom position, and quality settings. In many cases, it's done to analyze a given picture thoroughly. Moreover, from the metadata information included in the picture, you can find information about the geo-location of the image, especially when the picture was taken from a smartphone. With the OpenCV algorithms, any information extraction can be done with ease!

Text Extraction:

Let's get back to our parking space example. Let's say that the camera in your parking area can scan the license plate of every car that enters your car park. To make it useful, you need to extract the text (which in this case is the combination of letters and digits) from each plate. The text extraction methods are applied with the help of machine learning algorithms.

The extracted text is collected from the image and transferred to the given application or a specific file type. This way, you can easily send them to your ticket machine software. As a result, your customers can go to the ticket machine, choose their license plate number, and pay for their stopover.

Background Removal:

This finds applications, particularly in the eCommerce sector. Naturally, every online store wants to present their products as favorably as possible. That's why every picture before it can be published online, has to go through a background removal stage. Within this stage, the background is removed and replaced with transparency or solid color (particularly white). As a result, you have a picture where only the product in question is visible, and nothing distracts the viewer. It's another vital and commonly-used image processing method often done with the assistance of the OpenCV algorithms.[2]

III. OTHER SIMPLE OPENCV IN IMAGE PROCESSING TECHNIQUES

We have other simple OpenCV image processing techniques. They comprise:

- **Rotating**
- **Cropping**
- **Resizing and several others**

Why are they helpful? One might argue that these processes are so straightforward, they can be easily done even in MS Paint, not to mention more advanced graphic software. And it's true, these techniques raise no difficulties

But what happens when you want to resize or rotate 10,000 pictures? It's a different story, isn't it? That's why you want an ML algorithm to do it for you. OpenCV has the ready-made algorithms that will help you speed up this process. Image processing algorithms are not necessarily only about executing complicated and sophisticated processes. In many instances, they do simple corrections or modifications, which are time-consuming when done by hand. OpenCV In Image

Processing – Conclusion To sum up, if you're after image processing, the OpenCV library is your best bet. It's a multi-purpose, open-source platform that can be used in almost any image-related endeavor..[2]

What Are Color Spaces?

In the most common color space, RGB (Red Green Blue), colors are represented in terms of their red, green, and blue components. In more technical terms, RGB describes a color as a tuple of three components. Each component can take a value between 0 and 255, where the tuple (0, 0, 0) represents black and (255, 255, 255) represents white.

RGB is considered an “additive” color space, and colors can be imagined as being produced from shining quantities of red, blue, and green light onto a black background.

Table -1 Examples of colors in RGB

Color	RGB value
Red	255, 0, 0
Orange	255, 128, 0
Pink	255, 153, 255

RGB is one of the five major color space models, each of which has many offshoots. There are so many color spaces because different color spaces are useful for different purposes.

In the printing world, CMYK is useful because it describes the color combinations required to produce a color from a white background. While the 0 tuple in RGB is black, in CMYK the 0 tuple is white. Our printers contain ink canisters of cyan, magenta, yellow, and black.

In certain types of medical fields, glass slides mounted with stained tissue samples are scanned and saved as images. They can be analyzed in HED space, a representation of the saturations of the stain types—hematoxylin, eosin, and DAB—applied to the original tissue.

HSV and HSL are descriptions of hue, saturation, and brightness/luminance, which are particularly useful for identifying contrast in images. These color spaces are frequently used in color selection tools in software and for web design.

In reality, color is a continuous phenomenon, meaning that there are an infinite number of colors. Color spaces, however, represent color through discrete structures (a fixed number of whole number integer values), which is acceptable since the human eye and perception are also limited. Color spaces are fully able to represent all the colors we are able to distinguish between. Now that we understand the concept of color spaces, we can go on to use them in OpenCV.[6]

We can use relatively simple image processing and computer vision techniques in Python, using the `skimage`

library. With careful experimental design, a digital camera or a flatbed scanner, in conjunction with some Python code, can be a powerful instrument in answering many different kinds of problems. Consider the following two types of problems that might be of interest to a scientist.

Morphometrics: involves counting the number of objects in an image, analyzing the size of the objects, or analyzing the shape of the objects. For example, we might be interested automatically counting the number of bacterial colonies growing in a Petri dish.

Colorimetrics: involves analyzing the color of objects in an image. We could use image processing to look at the color of the solution, and determine when the titration is complete.[3]

IV. GAMMA TRANSFORM

Image enhancement mainly solves the problem of low contrast due to the small gray scale range of the image. The purpose is to enlarge the gray level of the output image to a specified extent, so that the details in the image appear to be sharper. There are several commonly used methods for contrast enhancement, such as linear transformation, piecewise linear transformation, gamma transformation, histogram normalization, histogram equalization, and local adaptive histogram equalization.

Assuming that the input image is I , the width is w , the height is h , first, the gray value is normalized to the range $[0, 1]$, and for the 8-bit image, it is divided by 255. $I(r, c)$ represents the gray value of the normalized r -th row and c -th column, and the output image is recorded as O . At the time, the image did not change. If the image as a whole or the region of interest is dark, it is possible to increase the image contrast; on the contrary, if the image as a whole or the region of interest is brighter, it is possible to reduce the image contrast. The gamma transformation of the image is essentially a power operation on each value in the image matrix. The power function power provided by Numpy implements this function. The code is implemented as follows:

1. `img = cv.imread("../testImages/4/img8.jpg", 0)`
2. `# image normalization`

3. $f_i = \text{img} / 255.0$
4. #
5. $\text{gamma} = 0.4$
6. $\text{out} = \text{np.power}(f_i, \text{gamma})$
7. $\text{cv.imshow}(\text{"img"}, \text{img})$
8. $\text{cv.imshow}(\text{"out"}, \text{out})$
9. cv.waitKey

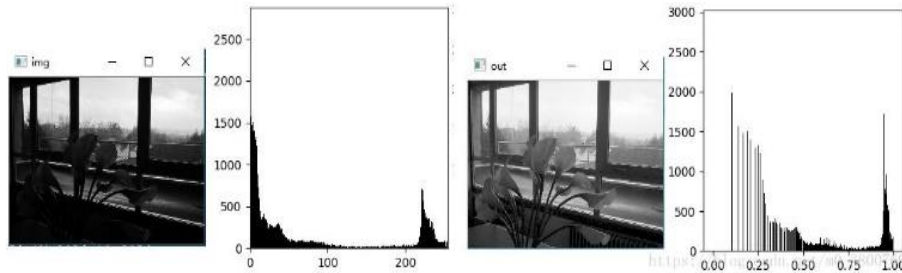


Fig.1 Gamma transform has a better effect in improving contrast, but requires manual adjustment γ value. [5]

V. CONTRAST AND CONTRAST ENHANCEMENT

Contrast refers to the amount of differentiation that is there between the various image features. Images having a higher contrast level generally display a greater degree of color or gray-scale variation than those of lower contrast.



Fig.2 An example of a low contrast image and a high contrast image

In Section 3 you will learn how convert the low contrast image to a high contrast image yourself using not one but three algorithms! Contrast Enhancement refers to the sharpening of image features to remove the noisy feature such as edges and contrast boundaries. Contrast Enhancement Algorithms aim to improve the perception of the image by human eye.

Types of Contrast Enhancement Algorithms:

Histogram Equalization:

Histogram equalization is an image processing technique that adjusts image intensities to improve contrast. Histogram Equalization is one of the simplest and commonly used method in low level image enhancement using the histogram. The logic behind Histogram Equalization is that the image with the best visual appearance, is the one whose histogram looks like the regular distribution. A Cumulative Distribution Function(CDF) of a histogram is the fraction of pixels in with an pixel value is less than or equal to the specified pixel value. Histogram Equalization is particularly useful in cases where both backgrounds and foregrounds are both bright or dark. Histogram Equalization can lead to better views of bone structure in x-ray images, and to better detail in over-exposed or under-exposed photos. A quintessential advantage of Histogram Equalization method is that it is a fairly straight forward image processing technique.

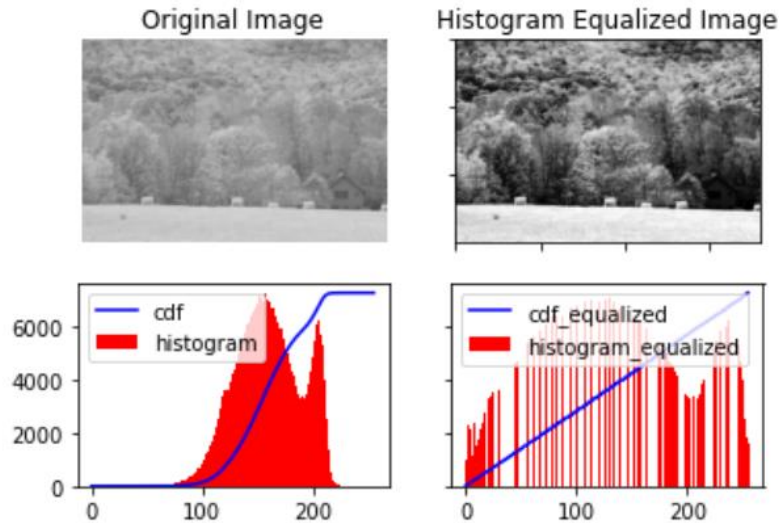


Fig.3 As we can see from the image above, the image after Histogram Equalization is much easier to interpret.

The Cumulative Distribution Function(CDF) of the image after Histogram Equalization ideally should be a straight line. Contrast Enhancement of a Black and White image is fairly straight forward. Histogram Equalization of color images is a little complicated. OpenCV loads color images in BGR (Blue Green Red) color space. In BGR, it is not possible to perform histogram equalization without affecting the color information as all channels contain color information, therefore we have to convert the BGR image into YCrCb. In YCrCb (Luminance; Chroma: Blue; Chroma: Red) color space, the Y channel of the image only contains intensity information where as Cr and Cb channels contain all the color information of the image. The operation must be performed only on the Y channel to get the Histogram Equalized output without changing any color related details.

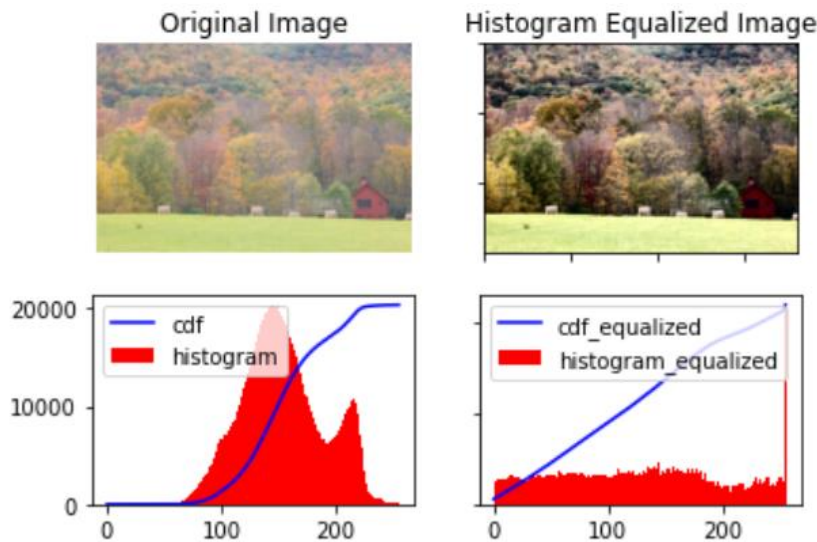


Fig. 4 The image before and after Histogram Equalization are shown above

Adaptive Histogram Equalization:

Adaptive Histogram Equalization computes many histograms for each of the separate part of the image, and uses them to redistribute the lightness values of the image, hence it differs from Histogram Equalization. Hence it is suitable for bettering the local contrast in images.

CLAHE:

Contrast Limited Adaptive Histogram Equalization(CLAHE) is a variant of Adaptive Histogram Equalization. CLAHE has one additional step over Adaptive Histogram Equalization and that is clipping of the histogram.

The 5 steps in CLAHE are mentioned below:

- Divide the image into tiny regions.
- Decide the mapping functions of local histogram.
- Choose the clipping point of histogram.

- Apply the function to every region.
- Reduce the noise by the background subtraction method.
-

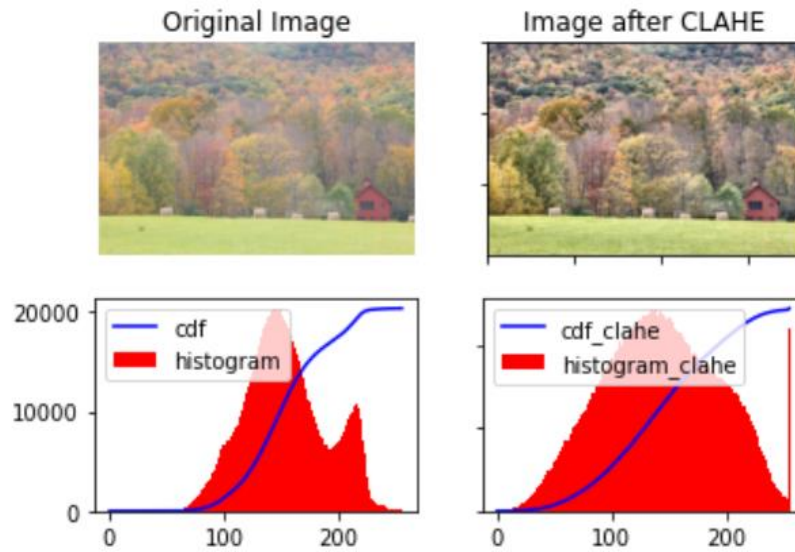


Fig.5 The image before and after CLAHE are shown above

Contrast Stretching:

In Contrast Stretching the contrast in an image is stretched from the range of intensity values it contains to span a desired range of values. It is also called Normalization.

Some Contrast Stretching techniques include: Minimum-Maximum, Percentage, and Piecewise Contrast Enhancement. Min-Max Contrast Stretching:

In Min-Max Contrast Stretching for each pixel:

$$\text{pixel} = ((\text{pixel} - \text{min}) / (\text{max} - \text{min})) * 255$$

Where min and max are the maximum and minimum pixel values in the image.

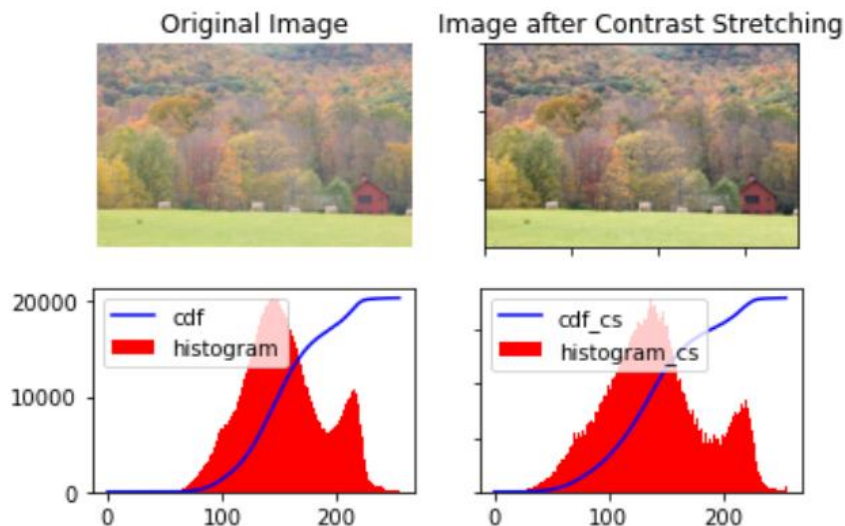


Fig.6 An image before and after Min-Max Contrast Stretching: [7]

Geometric Transformations of Images:

Transformations

OpenCV provides two transformation functions, cv2.warpAffine and cv2.warpPerspective, with which you can have all kinds of transformations. cv2.warpAffine takes a 2x3 transformation matrix while cv2.warpPerspective matrix takes a 3x3 transformation as input.[11]

**VI. CONCLUSION**

Using OpenCV and support of inbuilt functions in OpenCV, which performed image enhancement by just writing a few lines of codes and different filters used for image processing. This paper reviews different approaches, all these approaches are processed through Open CV Library. First there are different stages of image processing also other simple openCV in image processing and Gamma transform techniques, examples of use of openCV. Also reviews image contrast and contrast enhancement algorithms and geometric transformation of image.

REFERENCES

- [1]. <https://www.ijraset.com/files/serve.php?FID=8952>
- [2]. <https://addepto.com/the-use-of-opencv-in-image-processing-5-examples/>
- [3]. <https://datacarpentry.org/image-processing/aio/index.html>
- [4]. <https://developer.ibm.com/technologies/artificial-intelligence/articles/learn-the-basics-of-computer-vision-and-object-detection>
- [5]. <https://www.programmingsought.com/article/6367364560/>
- [6]. <https://realpython.com/python-opencv-color-spaces/>
- [7]. <https://iq.opengenus.org/contrast-enhancement-algorithms/>
- [8]. <https://stackabuse.com/introduction-to-image-processing-in-python-with-opencv/>
- [9]. <https://www.toptal.com/opencv/python-image-processing-in-computational-photography>
- [10]. <https://analyticsindiamag.com/guide-to-opencv-functions-for-image-processing/>
- [11]. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html