Contrast Limited Adaptive Histogram Equalization (CLAHE)

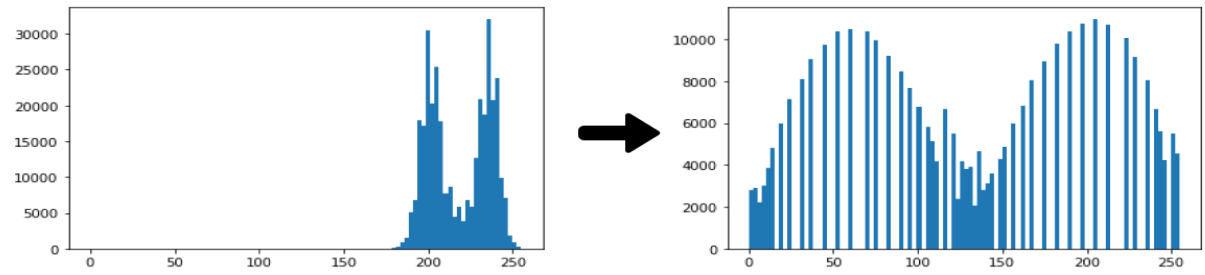
Introduction

Contrast enhancement algorithms have evolved over the last few decades to meet the needs of its objectives. There are two main goals in enhancing an image's contrast: (i) Improving its appearance for visual interpretation and (ii) facilitating/increasing the performance of subsequent tasks (e.g., image analysis, object detection, and image segmentation). Most contrast enhancement techniques rely on histogram modifications, which can be applied globally or locally. The Contrast Limited Adaptive Histogram Equalization (CLAHE) method overcomes the limitations of global approaches by enhancing local contrast.

CLAHE – Contrast Limited Adaptive Histogram Equalization is an Image Equalization method. CLAHE is a variation of Adaptive histogram equalization (AHE) that prevents contrast over-amplification.

Adaptive histogram equalization

Consider an image in which the pixel values are limited to a specified range. A brighter image, for example, will have all pixels restricted to high values. On the other hand, an excellent image will contain pixels from all sections of the image. So you need to stretch this histogram to either end, which is what Histogram Equalization does (in simple words). This usually increases the image's contrast.



Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE works on small areas of an image called tiles rather than the complete image. The surrounding tiles are blended using bilinear interpolation to remove the false boundaries. This algorithm can be used to improve image contrast.

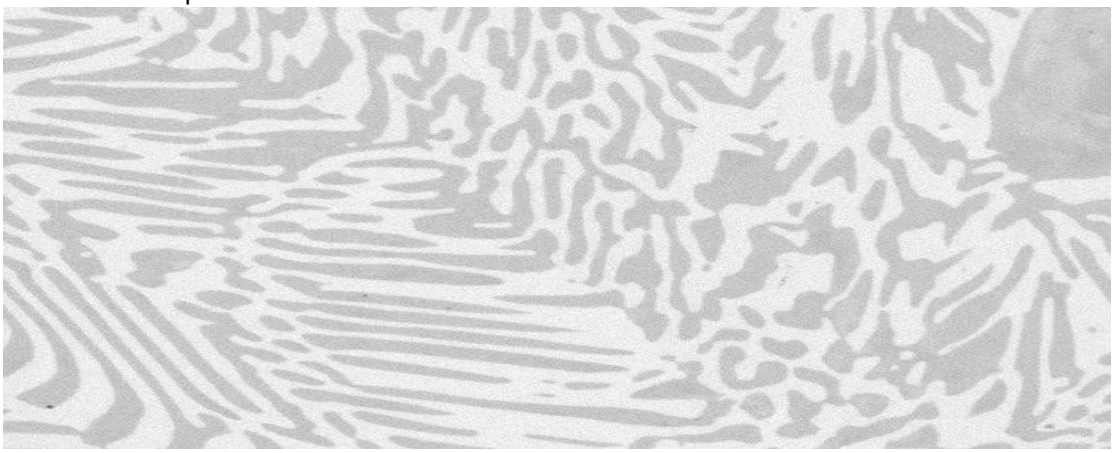
CLAHE can also be applied to color images, often to the luminance channel. The results of equalizing only the luminance channel of an HSV image outperform equalizing all channels of a BGR image.

CLAHE Algorithm

CLAHE was initially used to improve low-contrast medical images.CLAHE differs from ordinary AHE in that it limits contrast. To address the issue of noise amplification, the CLAHE implemented a clipping limit. Before computing the Cumulative Distribution Function, the CLAHE limits the amplification by clipping the histogram at a predefined value (CDF). The CLAHE technique divides an input original image into non-overlapping contextual regions known as sub-images, tiles, or blocks.

The CLAHE is defined by two parameters: Block Size (BS) and Clip Limit (CL). These two parameters primarily govern improved image quality. When CL is increased, the image becomes brighter because the input image has a very low intensity and a larger CL makes its histogram flatter.
As the BS increases, the dynamic range expands, and the image contrast increases. The two parameters determined at the point of maximum entropy curvature produce subjectively good image quality when using image entropy.

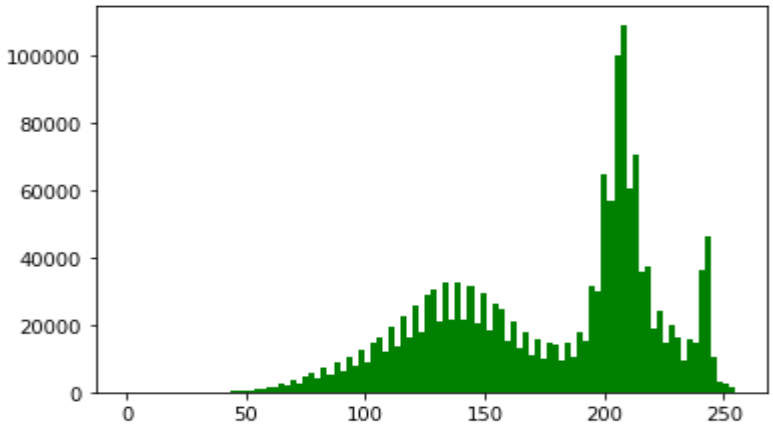
Let's look at an example



The division may be seen in the above image, although it is not as clear as it could be. So, let's look at the histogram and utilize equalization to stretch it to the threshold.

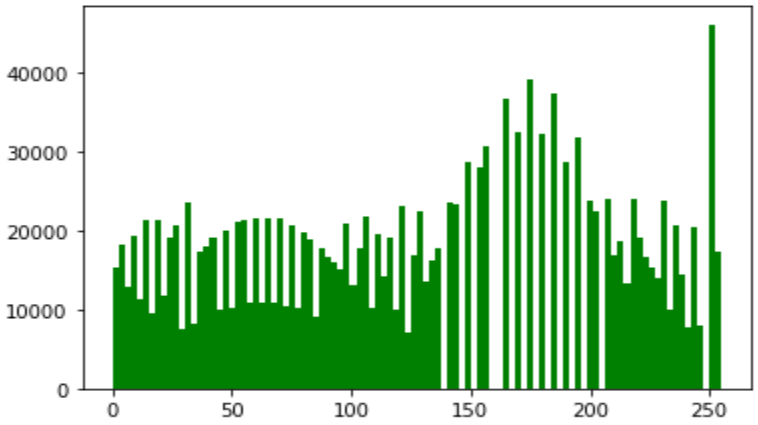
```
# Import the necessary libraries for CLAHE
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread("testing_image.jpeg", 0)
equ = cv2.equalizeHist(img)
print(img)
print(equ)
```

First, I read my greyscale image and assigned it to the variable image. We can use cv2.equalizeHist to equalize histograms (Image).
Let's have a look at the histogram of our test photograph. And you can see that it is tilted to the right.
plt.hist(img.flat, bins=100, range=(0, 255))

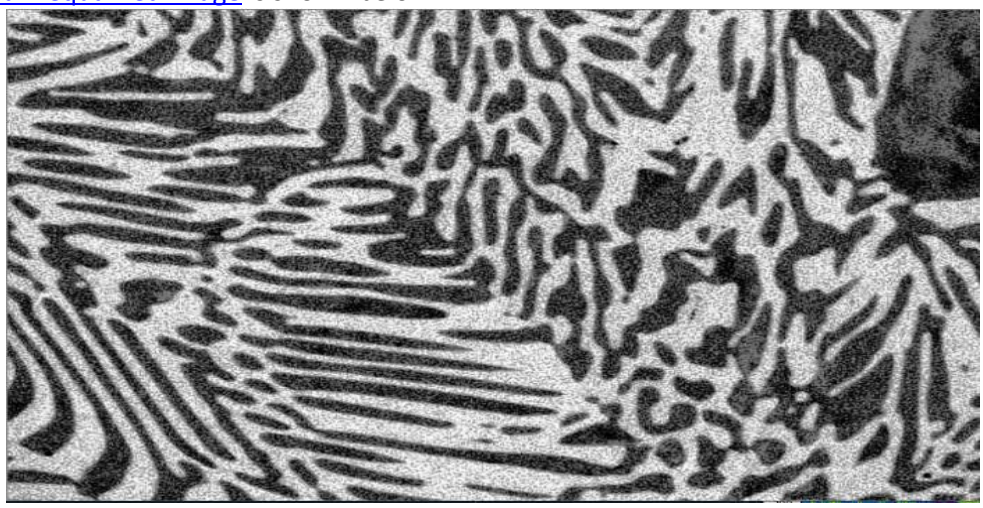


Let's have a look at the histogram of the equalized image. The histogram is also expanded to 255.

plt.hist(equ.flat, bins=100, range=(0, 255))



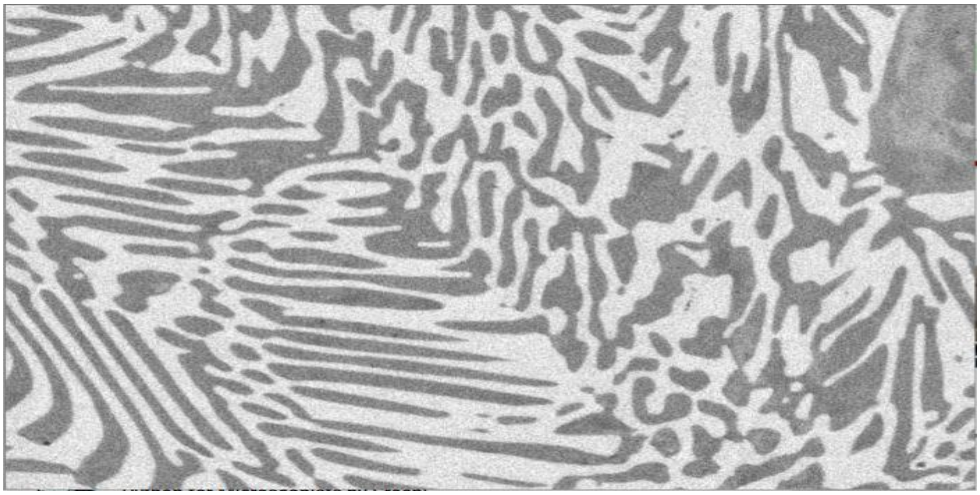
The histogram equalized image is shown below.



The histogram e result is shown below. As you can see, the above image has a lot of noise because it considers the image's global contrast rather than just the local contrast. As a result, performing global equalization on your image may not work very well. In such cases, we can use Adaptive Histogram Equalization, also known as CLAHE.

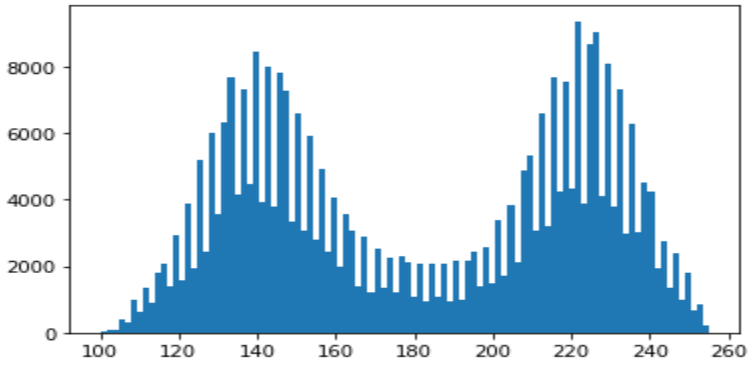
Contrast Limited AHE (CLAHE) is a variant of adaptive histogram equalization that limits contrast amplification to reduce noise amplification. CLAHE, in a nutshell, performs histogram equalization in small patches or small tiles with high accuracy and contrast limiting.

clahe = cv2.createCLAHE(clipLimit =2.0, tileGridSize=(8,8))
cl_img = clahe.apply(img)



As shown in the image above, CLAHE produces significantly better results than the standard equalized image. However, there is still a lot of noise. Let's see how thresholding works to get better results. Use.tiff file format instead of.jpeg file format for better image results. Before we begin thresholding, we must first examine the CLAHE image's histogram.

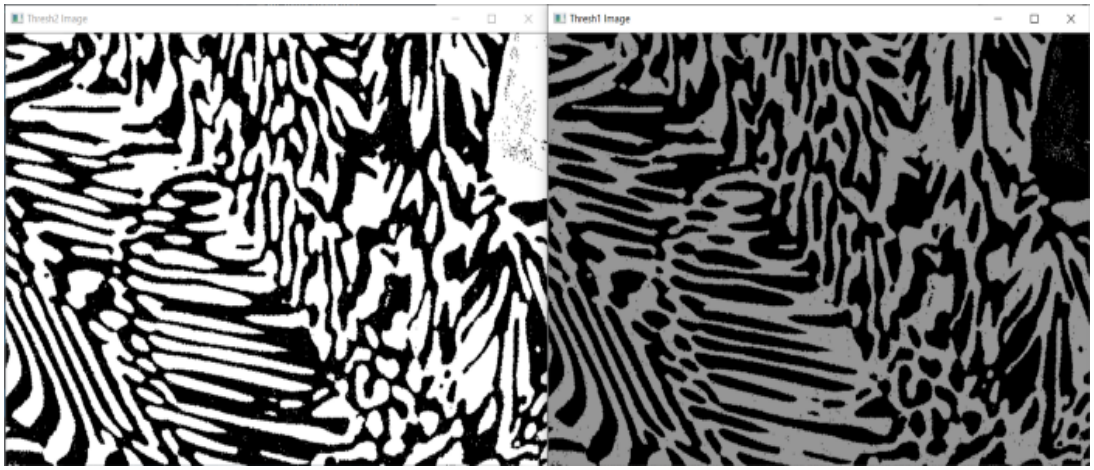
plt.hist(cl_img.flat, bins=100, range=(100, 255))



As shown in the histogram above, there is a dip between 160 and 200, and we can choose a close number to separate those two peaks. We can do the thresholding after we've decided on a close number (I've chosen 190).

ret, thresh1 = cv2.threshold(cl_img, 190, 150, cv2.THRESH_BINARY)
ret, thresh2 = cv2.threshold(cl_img, 190, 255, cv2.THRESH_BINARY_INV)

Ignore the first ret argument. We can get the thresholded image to variablethresh1 and variablethresh2. The image is the first argument in the above code section, followed by the threshold value we chose, a value for all the thresholded pixels, and finally, a method. I separated the variables THRESH BINARY and THRESH BINARY INV.



The grey level in the first threshold image (thresh1) is 150, and the grey level in the second threshold image (thresh2) is 255. This is simply histogram-based thresholding.
Using the histogram, we determined that 190 is the best value in the preceding example. However, there is an easier way to find the best value with OTSU.

ret, thresh3 = cv2.threshold(cl_img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
We can automatically segment it using OTSU.

https://amroamroamro.github.io/mexopencv/opencv/clahe_demo_gui.html
https://en.wikipedia.org/wiki/Adaptive_histogram_equalization
https://www.analyticsvidhya.com/blog/2022/08/image-contrast-enhancement-using-clahe/