

▼ Contents:

Plotting Histogram

Histogram Equalisation

```
1 import numpy as np
2 import cv2
3 from google.colab import files
4 uploaded=files.upload()
```

Choose Files flower2.jpg

- **flower2.jpg**(image/jpeg) - 36251 bytes, last modified: 6/22/2021 - 100% done
Saving flower2.jpg to flower2.jpg

```
1 img=cv2.imread("flower2.jpg")
2 from google.colab.patches import cv2_imshow
3 cv2_imshow(img)
```

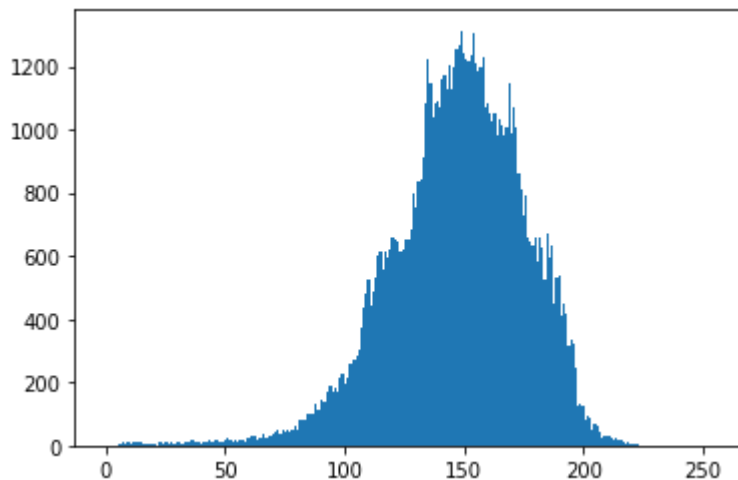


▼ Plotting Histogram

Using MATPLOTLIB and OPENCV

```
1 from matplotlib import pyplot as plt
2 height = img.shape[0]
3 width = img.shape[1]
4 max_intensity = 255
5 # convert image to grayscale
6 img_gs = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
7 cv2_imshow(img_gs)
```

```
8 # plot histogram of image
9 plt.hist(img_gs.ravel(), 256, [0, 255])
10 plt.show()
```



▼ **Histogram Equalisation**

Syntax: `cv.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])`

images : it is the source image of type `uint8` or `float32`. it should be given in square brackets, ie, "`[img]`".

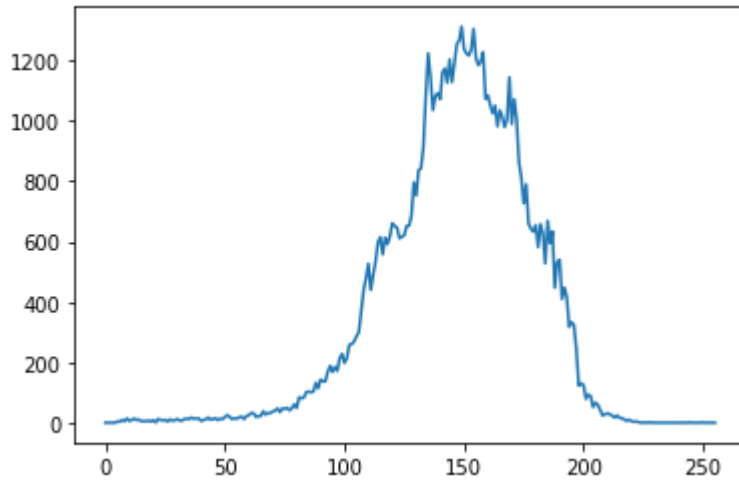
channels : it is also given in square brackets. It is the index of channel for which we calculate histogram. For example, if input is grayscale image, its value is `[0]`. For color image, you can pass `[0]`, `[1]` or `[2]` to calculate histogram of blue, green or red channel respectively.

mask : mask image. To find histogram of full image, it is given as `"None"`.

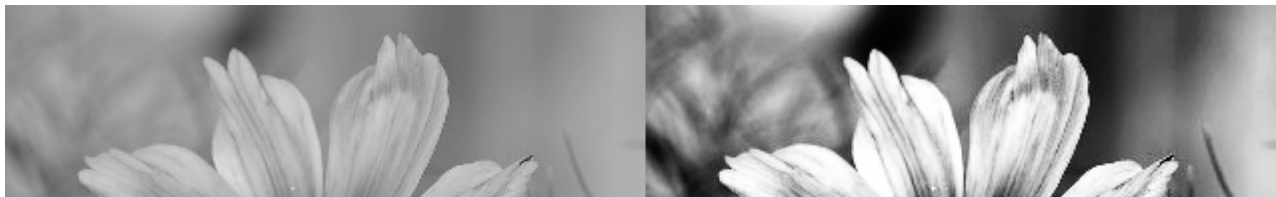
histSize : this represents our BIN count. Need to be given in square brackets. For full scale, we pass `[256]`.

ranges : this is our RANGE. Normally, it is `[0,256]`.

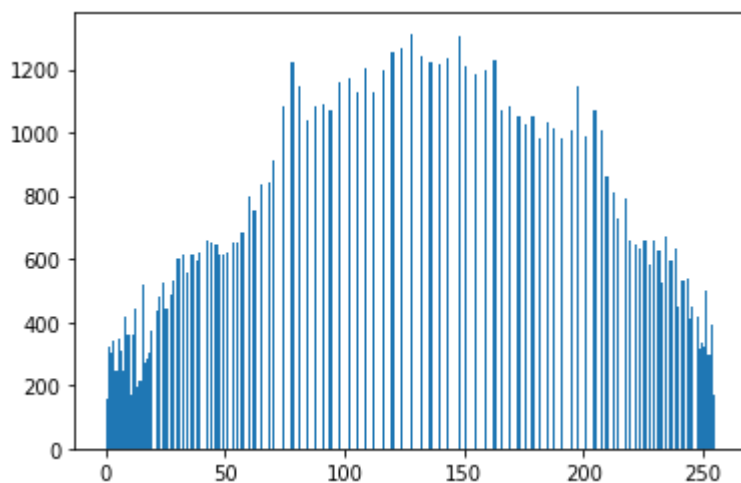
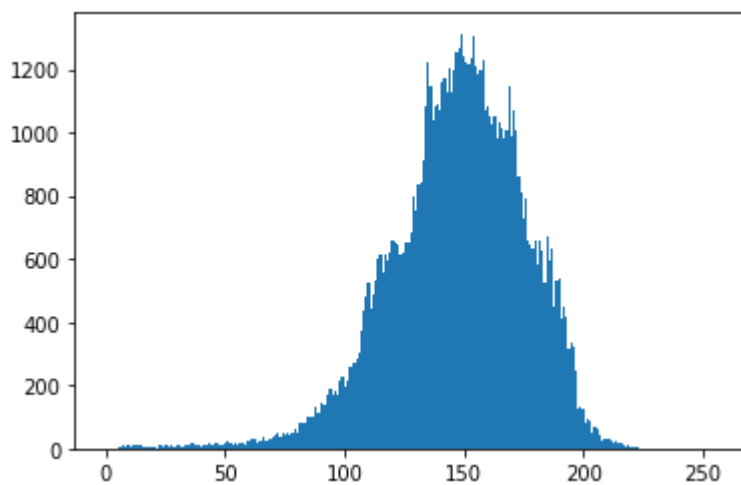
```
1 histg = cv2.calcHist([img_gs],[0],None,[256],[0,256]) # Calculate histogram
2 cum_hist = histg.copy()
3 #img2=img.copy()
4 plt.plot(histg)
5 plt.show()
```



```
1 equ = cv2.equalizeHist(img_gs)
2
3 # stacking images side-by-side
4 res = np.hstack((img_gs, equ))
5 res1= np.hstack((res, img_gs))
6 cv2_imshow(res)
7 cv2_imshow(res1)
```



```
1 plt.hist(img_gs.ravel(), 256, [0, 255])  
2 plt.show()  
3 plt.hist(equ.ravel(), 256, [0, 255])  
4 plt.show()
```



✓ 0s completed at 12:01

● ✕