```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow

img=cv2.imread("/content/tiger.jpg")
cv2_imshow(img)
```


```
img1=cv2.imread("/content/tiger.jpg",0)
cv2_imshow(img1)
```


```
# Check the image matrix data type (could know the bit depth of the image)
print(img.dtype)
# Check the height of image
print('Hight',img.shape[0])
# Check the width of image
print(img.shape[1])
# Check the number of channels of the image
print(img.shape[2])
```

```
uint8
Hight 682
1024
3
```

```
resized_image = cv2.resize(img, (500,200))
cv2_imshow(resized_image)
```

```
cv2.imwrite("output3.png",img)
```

```
True
```

```
import matplotlib.pyplot as plt

total_number_of_elements= img.size
print(total_number_of_elements)
```

```
2095104
```

```
# To get the value of the pixel (x=50, y=50), we would use the following code
(b, g, r) = img[50, 50]
print("Pixel at (50, 50) - Red: {}, Green: {}, Blue: {}".format(r,g,b))
```

```
Pixel at (50, 50) - Red: 1, Green: 1, Blue: 1
```

BGR stands for Blue(255, 0, 0), Green(0, 255, 0), Red(0, 0, 255). **OpenCV uses BGR color as a default color **space to display images, when we open a image in openCV using cv2. imread() it display the image in BGR format. And it provides color-changing methods using cv2.

```
# Using indexing we modified a whole region rather than one pixel
# For the top-left corner of the image, we can rewrite
# the color channels in folowing way:
resized_image[0:150, 0:300] = [0,255,0]

cv2_imshow(resized_image)
```

```
img_gray = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
cv2_imshow(img_gray)
```

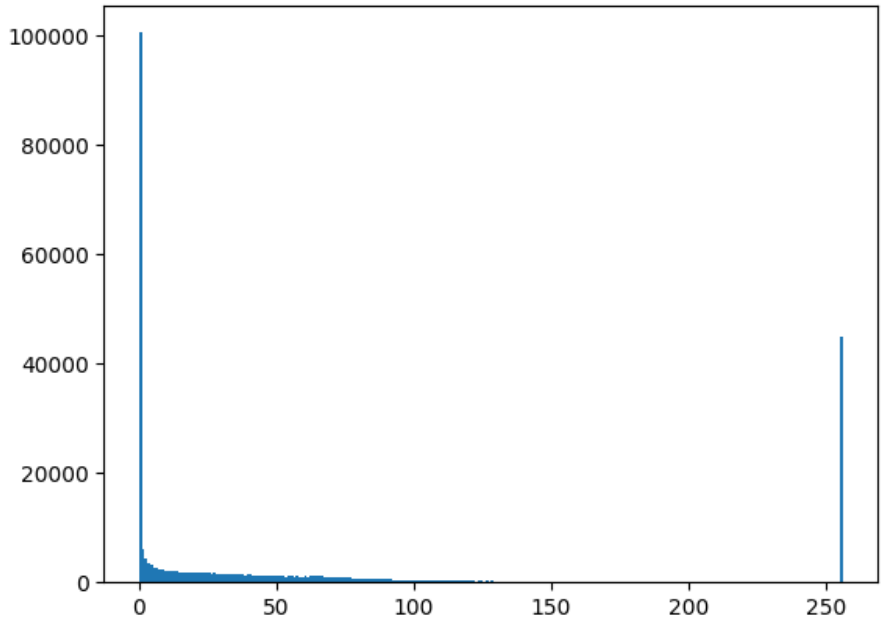
Grayscale Transform This section provides some examples of conducting mathematical transformations of the grayscale image.

```
# This is an inverse operation of the grayscale image, you could see that the
# bright pixels become dark, and the dark pixels become bright
im2 = 255 - img_gray
cv2_imshow(im2)
```

```
# Another transform of the image, after adding a constant,
# all the pixels become brighter and a hazing-like effect of the image is generated
im3 = (100.0/255)*img_gray + 100
cv2_imshow(im3)
```

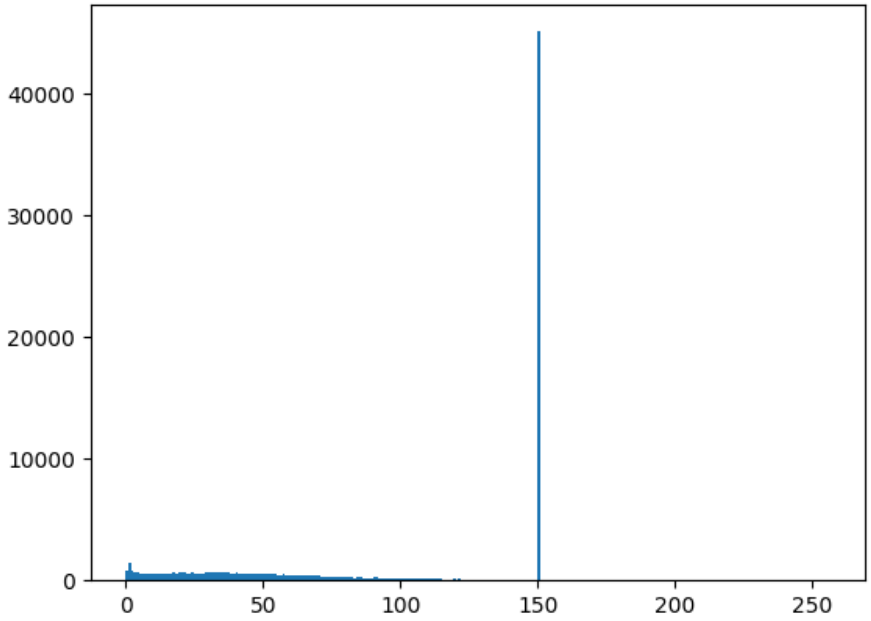
```
# The lightness level of the gray_image decreases after this step
im4 = 255.0*(img_gray/255.0)**2
cv2_imshow(im4)
```

```
import matplotlib.pyplot as plt
plt.hist(resized_image.ravel(),bins = 256, range = [0,256])
plt.show()
```



Plot the histogram of the gray image. We could observe that the frequency of the image hist has decreased ~ 1/3 of the histogram of color image

```
plt.hist(img_gray.ravel(),bins = 256, range = [0,256])
plt.show()
```



Colab paid products - Cancel contracts here

✓ 0s completed at 11:50 PM