```
import cv2
import matplotlib.pyplot as plt
from skimage.color import rgb2gray
from skimage.filters import threshold_otsu, gaussian
```

isr

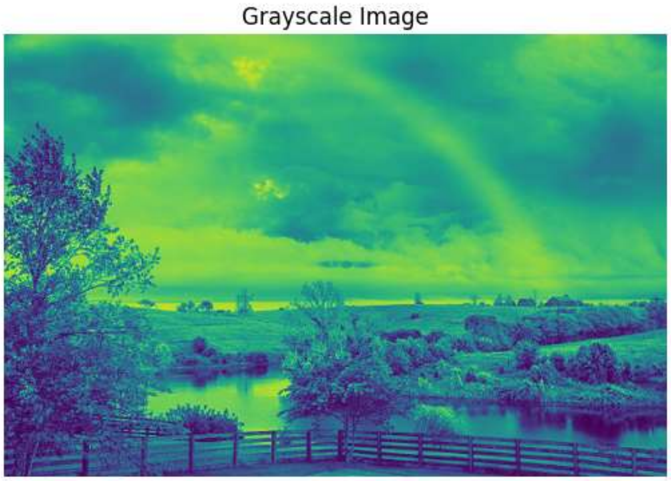
```
Start coding or generate with AI.
```

Double-click (or enter) to edit

```
image = cv2.imread('/content/pic.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

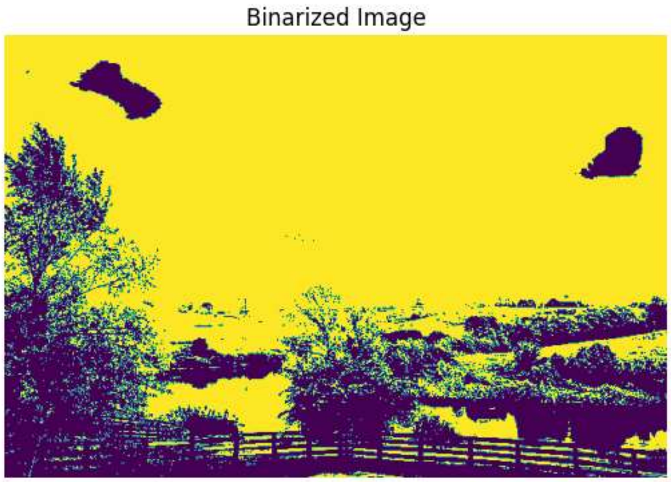
```
def show_img(img, title="Image"):
    plt.imshow(img)
    plt.title(title)
    plt.axis('off')
    plt.show()

show_img(image, title="Actual Image")
```

```
from skimage.color import rgb2gray
gray_image = rgb2gray(image)
show_img(gray_image, title="Grayscale Image")
print("Colored image shape:", image.shape)
print("Grayscale image shape:", gray_image.shape)
```

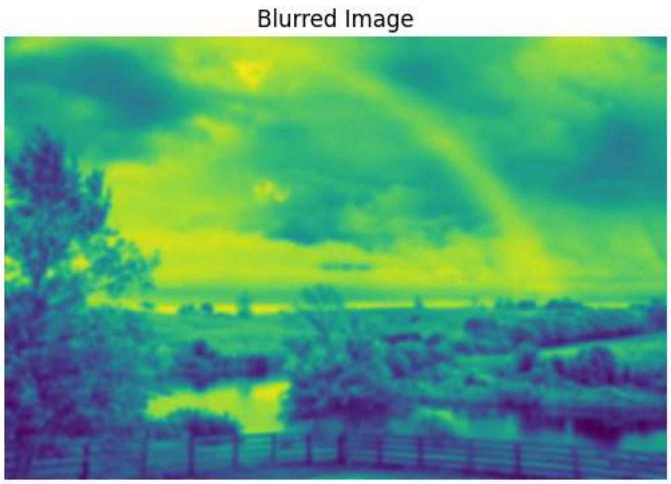


```
Colored image shape: (408, 612, 3)
Grayscale image shape: (408, 612)
```

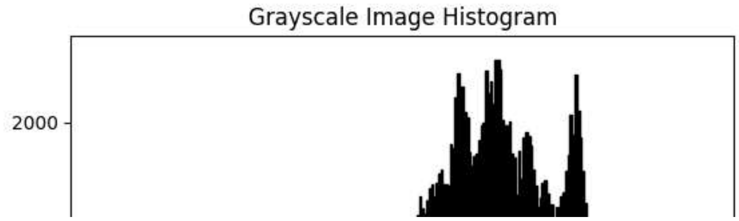
```
from skimage.filters import threshold_otsu
thresh = threshold_otsu(gray_image)
binary_image = gray_image > thresh
show_img(binary_image, title="Binarized Image")
```



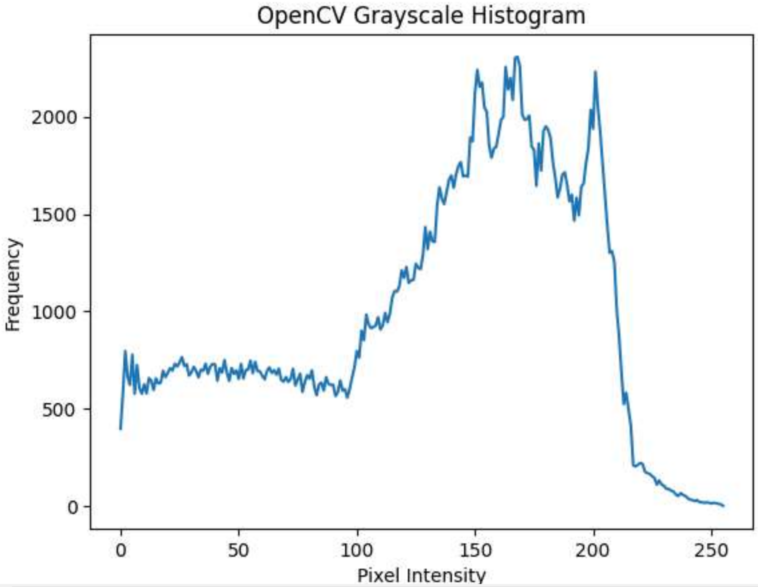
```
from skimage.filters import gaussian
blurred_image = gaussian(gray_image, sigma=2)
show_img(blurred_image, title="Blurred Image")
```



```
import numpy as np

flattened_gray_image = gray_image.ravel()
plt.hist(flattened_gray_image, bins=256, range=(0.0, 1.0), fc='k', ec='k')
plt.title('Grayscale Image Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
```



```
hist = cv2.calcHist([gray_image.astype('float32')], [0], None, [256], [0, 1])
plt.plot(hist)
plt.title('OpenCV Grayscale Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
```



Start coding or generate with AI.