

#Note this code is mainly for COLAB users as they cannot import their own image from disk,
 #This takes imgs from the net as a link.

#For NORMAL procedure while doing it on JUPYTER using cv2.imread visit--
<https://colab.research.google.com/drive/1PUMdKxTVdCzVfcDIk1A5d6wCiPds4Gu2#scrollTo=uyTCKR>
 #Do not execute code on the above link as it will not work... use on jupyter and provide p
 #People using colab may also refer to this link to know what on earth does the code say as
 #I've covered every block of code there over here, so peace.
 #The code on the above page is different than this one

#Following is the code for ppl who are using imgs from net-

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
from skimage import io
from skimage.color import rgb2gray
```

```
#https://images.fineartamerica.com/images-medium-large-5/1-peacock-feather-ithinksky.jpg #
#Copy the above link for a diff pic and change the link for img. If ya wanna use ur own li
img = io.imread('https://www.industrialempathy.com/img/remote/ZiClJf-640w.jpg')
img1 = rgb2gray(img) #For grayscale
```

```
#RGB
plt.imshow(img)
plt.title('Original')
plt.axis('off')
plt.show()
```

```
#BGR
plt.imshow(cv2.cvtColor(img, cv2.COLOR_RGB2BGR))
plt.title('BGR')
plt.axis('off')
plt.show()
```

```
#Grayscale
plt.imshow(img1, cmap = plt.cm.gray)
plt.title('Grayscale')
plt.axis('off')
plt.show()
```

```
print('')
#Size of img
print(img.shape)
print('')
print(img1.shape)#When loaded as grayscale note there are no color channels for grayscale
print('')
#minimum and maximum pixel value present in the image
print (img.min())
print (img.max())
print('')
```

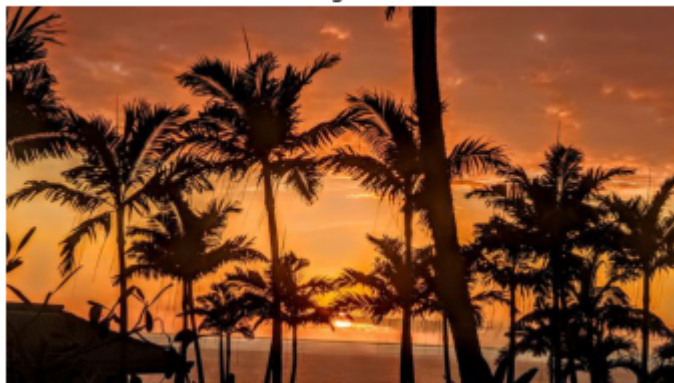
```
#Splitting Color Channel
plt.imshow(img):plt.axis('off'):plt.title('RGB')
```

```
plt.imshow(img, cmap=plt.cm.bgr) #plt.imshow(img, cmap=plt.cm.bgr)
b = img[:, :, 0]
g = img[:, :, 1]
r = img[:, :, 2]
fig, bgr = plt.subplots(1, 3)
bgr[0].imshow(cv2.cvtColor(b, cv2.COLOR_BGR2RGB));bgr[0].axis('off');bgr[0].set_title('blue')
bgr[1].imshow(cv2.cvtColor(g, cv2.COLOR_BGR2RGB));bgr[1].axis('off');bgr[1].set_title('green')
bgr[2].imshow(cv2.cvtColor(r, cv2.COLOR_BGR2RGB));bgr[2].axis('off');bgr[2].set_title('red')
plt.show()
print('')

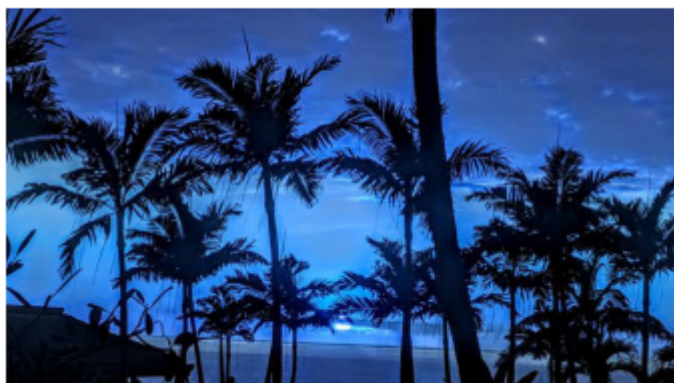
#Cropping
plt.imshow(img)
plt.title('Pic')
plt.axis('on')
plt.show()
print('')
roi = img[100:200, 300:640] #img[range of y, range of x]
plt.imshow(roi)
plt.title('Cropped_Pic')
plt.axis('off')
plt.show()
```



Original



BGR



Grayscale



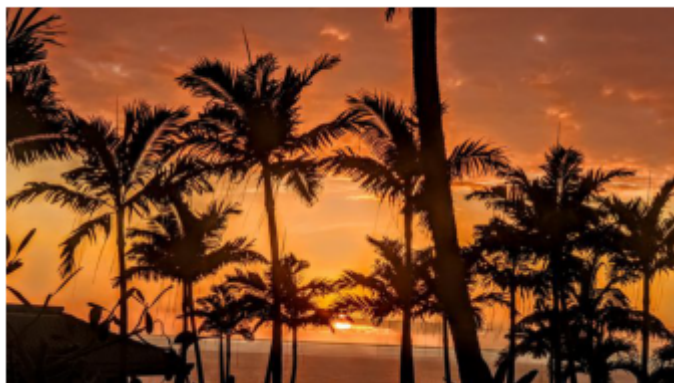
(360, 640, 3)

(360, 640)

0

255

RGB



blue

green

red

