Digital image processing Lab

CEL 444

Lab Journal 2

Name: M Muneeb Ahmed Kiani
Enrollment : 01-135212063
BSCS 7B

Department of Computer Science
BAHRIA UNIVERSITY, ISLAMABAD

Task # 1:

Practice loading images using different methods and paths in Python with OpenCV. You will use the following approaches to load images:

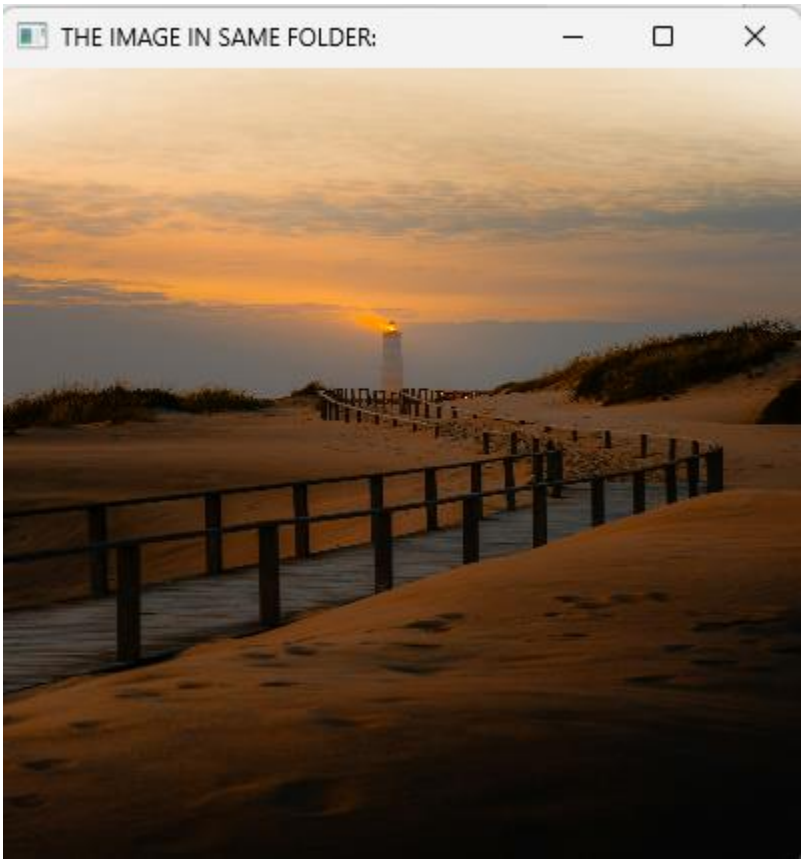
Instructions

1. **Setup:**
    - Create a folder named image_task.
    - Inside image_task, create two subfolders: images and data.
    - Place an image file (e.g., example.jpg) in both the images and data folders.
    - Ensure you have OpenCV installed. You can install it using pip install opencv-python.

1. **Image in the Same Folder**

```
import cv2
```

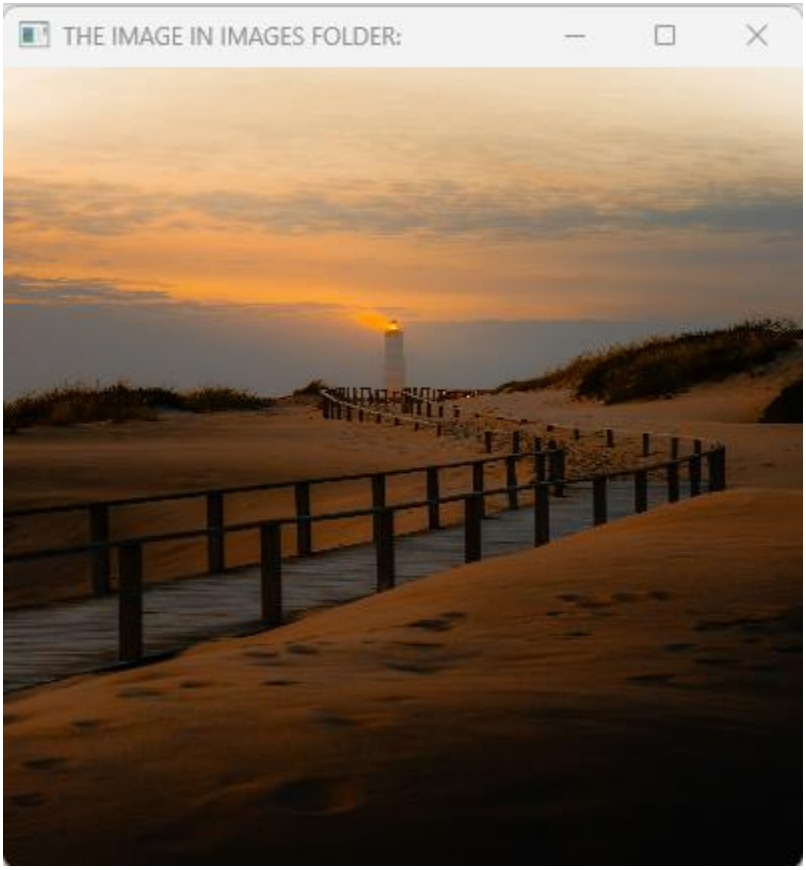
```
image=cv2.imread("image_task/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
cv2.imshow("THE IMAGE IN SAME FOLDER:",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



2.Image in a Different Folder

```
import cv2
```

```
image=cv2.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
cv2.imshow("THE IMAGE IN IMAGES FOLDER:",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



3.Image Using an Absolute Path

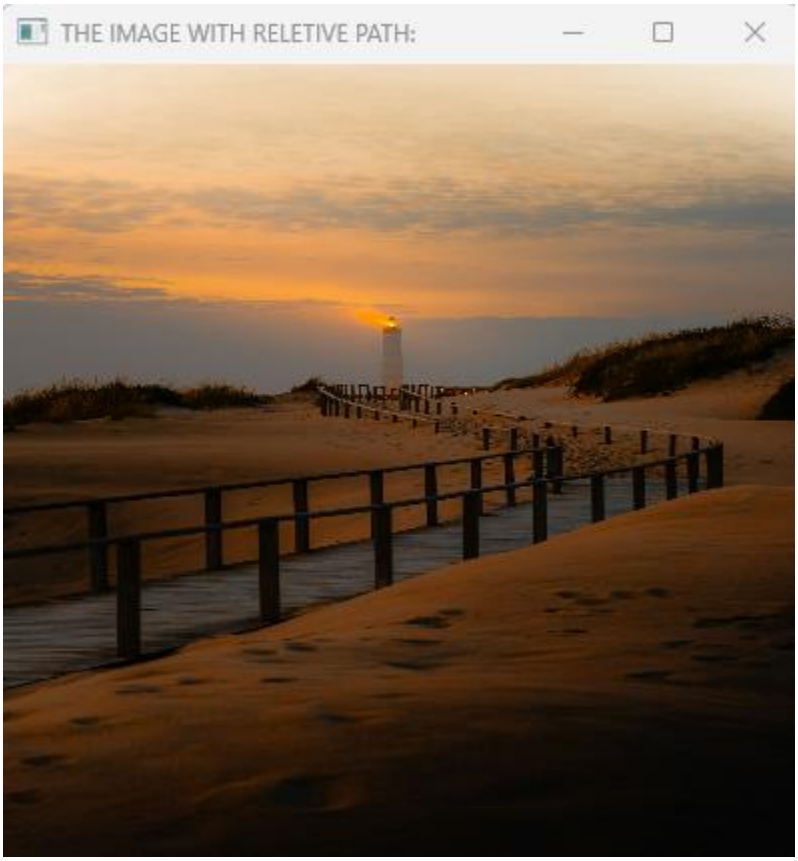
```
import cv2
```

```
image=cv2.imread("C:/Users/Muneeb Kiyani/image_task/data/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
cv2.imshow("THE IMAGE IN DATA FOLDER USING ABSOLUTE PATH:",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

4.Image Using a Relative Path

```
import cv2
```

```
image=cv2.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
cv2.imshow("THE IMAGE WITH RELETIVE PATH:",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



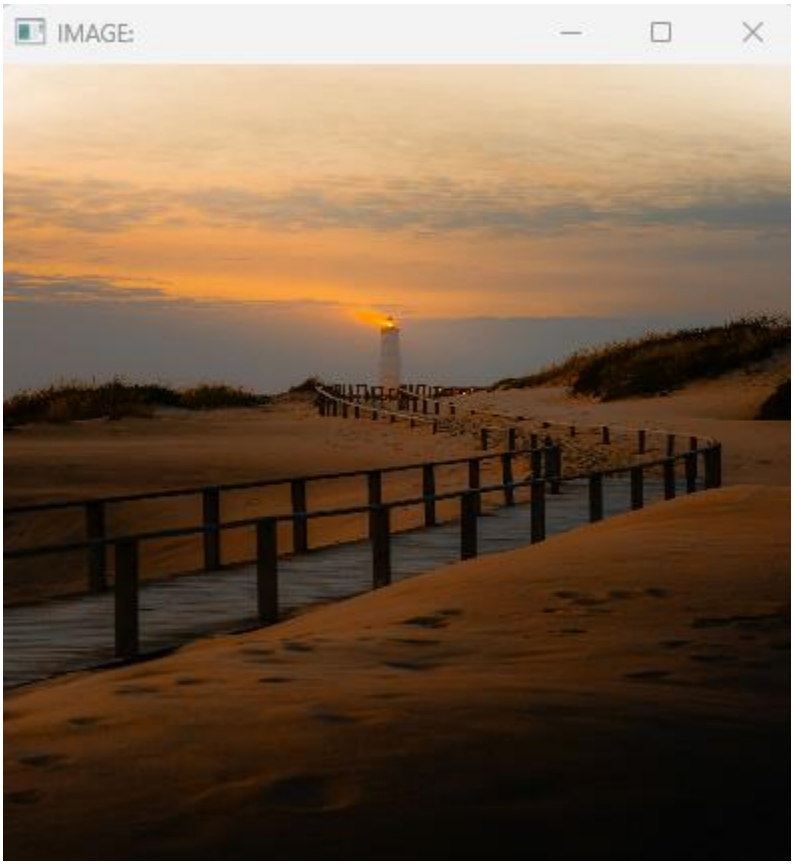
Task # 2:

Learn how to find and display the dimensions of an image using Python and OpenCV. This task will help you understand how to retrieve image dimensions and use them in image processing.

```
import cv2
```

```
image=cv2.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
width,height,channel=image.shape
print(f"width:{width} pixels")
print(f"height:{height} pixels")
print(f"No of channels:{channel}")
cv2.imshow("IMAGE:",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
width:6240 pixels
height:4160 pixels
No of channels:3
```



Task # 3:

Learn how to load and display an image using three different libraries: PIL (Pillow), Matplotlib, and OpenCV. This task will help you understand how to handle images in various Python libraries.

IMAGE USING OPENCV

```
import cv2
```

```
image=cv2.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
cv2.imshow("IMAGE USING OPENCV:",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

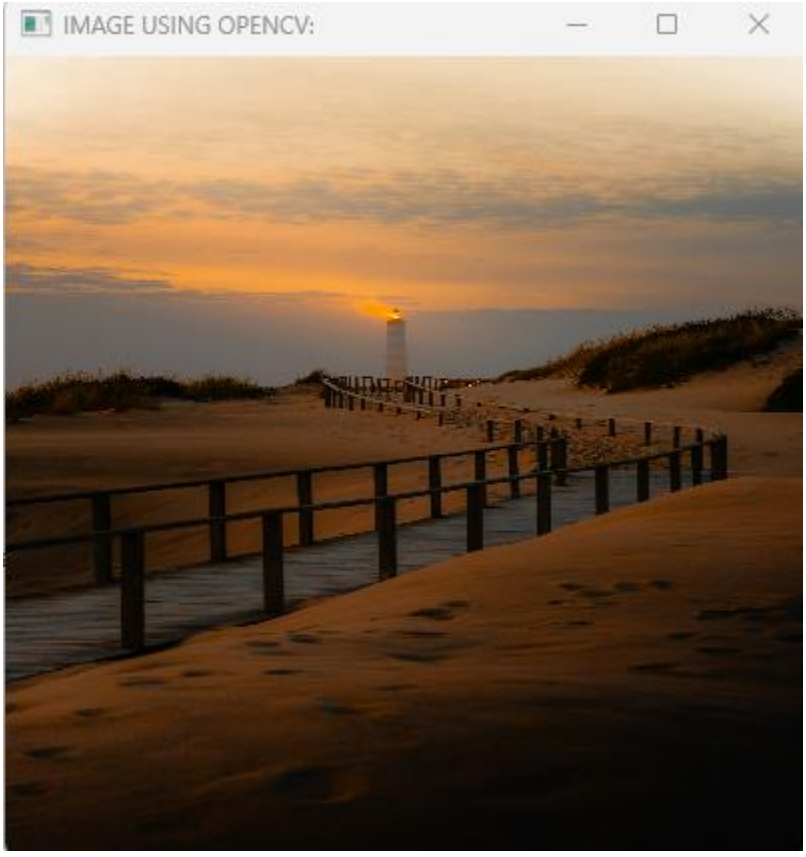


IMAGE USIG MATPLOT:

```
import matplotlib.pyplot as plt
image=plt.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
plt.title("THE IMAGE WITH MATPLOT")
plt.imshow(image)
plt.axis('off')
plt.show()
```

IMAGE WITH PILLOW

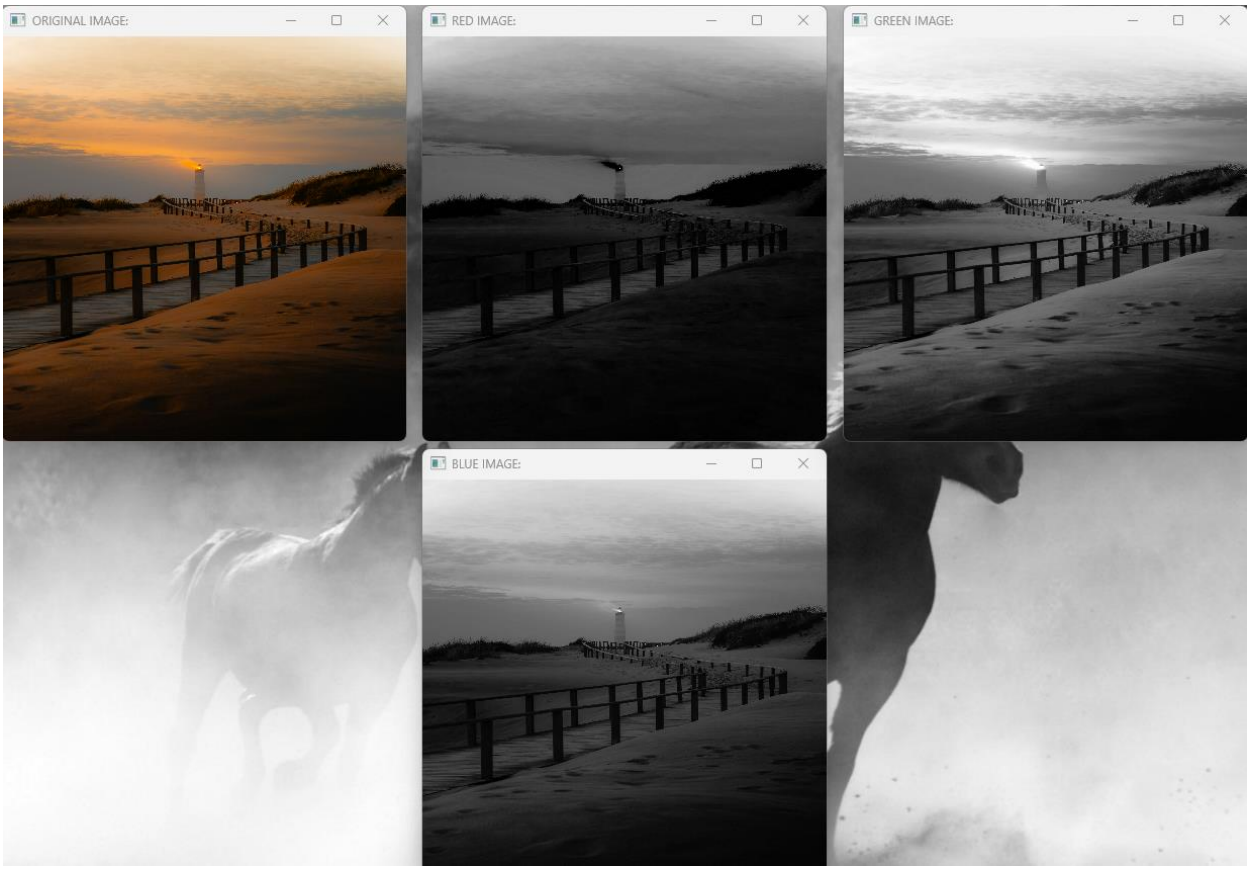
```python
from PIL import Image
import matplotlib.pyplot as plt
image=Image.open("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
plt.title("THE IMAGE USING PILLOW:")
plt.imshow(image)
plt.axis('off')
plt.show()
```

Task # 4:

Learn how to split an image into its individual color channels (Blue, Green, and Red) using Python and OpenCV. This task will help you understand how to access and manipulate different color components of an image.

```
import cv2
```

```
image=cv2.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
R,B,G=cv2.split(img)
cv2.imshow("ORIGINAL IMAGE:",img)
cv2.imshow("RED IMAGE:",R)
cv2.imshow("BLUE IMAGE:",B)
cv2.imshow("GREEN IMAGE:",G)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Task # 5:

Write a program that loads iris data set:

Explanation: The iris data set comprises data of 3 different flower species of the Iris plant. The first four columns of the data set represent sepal length, sepal width, petal length and petal width respectively. The last column is a label which tells the type of flower (1,2 or 3). The first 50 rows correspond to type 1, the next 50 rows to type 2 and the last 50 rows to type 3.

```
import pandas as pd
```

```
irisdata=pd.read_csv('iris.data',header=None)
iris_columns=['sepal length','sepal width','petal length','petal width']
print("IRIS DATASET:")
print(irisdata)
print("IRIS DATASET DESCRIBE:")
print(irisdata.describe())
```

```
IRIS DATASET:
       0    1    2    3               4
0    5.1  3.5  1.4  0.2     Iris-setosa
1    4.9  3.0  1.4  0.2     Iris-setosa
2    4.7  3.2  1.3  0.2     Iris-setosa
3    4.6  3.1  1.5  0.2     Iris-setosa
4    5.0  3.6  1.4  0.2     Iris-setosa
..   ...  ...  ...  ...             ...
145  6.7  3.0  5.2  2.3  Iris-virginica
146  6.3  2.5  5.0  1.9  Iris-virginica
147  6.5  3.0  5.2  2.0  Iris-virginica
148  6.2  3.4  5.4  2.3  Iris-virginica
149  5.9  3.0  5.1  1.8  Iris-virginica

[150 rows x 5 columns]
IRIS DATASET DESCRIBE:
                0           1           2           3
count  150.000000  150.000000  150.000000  150.000000
mean     5.843333    3.054000    3.758667    1.198667
std      0.828066    0.433594    1.764420    0.763161
min      4.300000    2.000000    1.000000    0.100000
25%      5.100000    2.800000    1.600000    0.300000
50%      5.800000    3.000000    4.350000    1.300000
75%      6.400000    3.300000    5.100000    1.800000
max      7.900000    4.400000    6.900000    2.500000
```

Task # 6:

Learn how to convert a color image to grayscale manually using Python and OpenCV. This task will help you understand the underlying process of grayscale conversion.

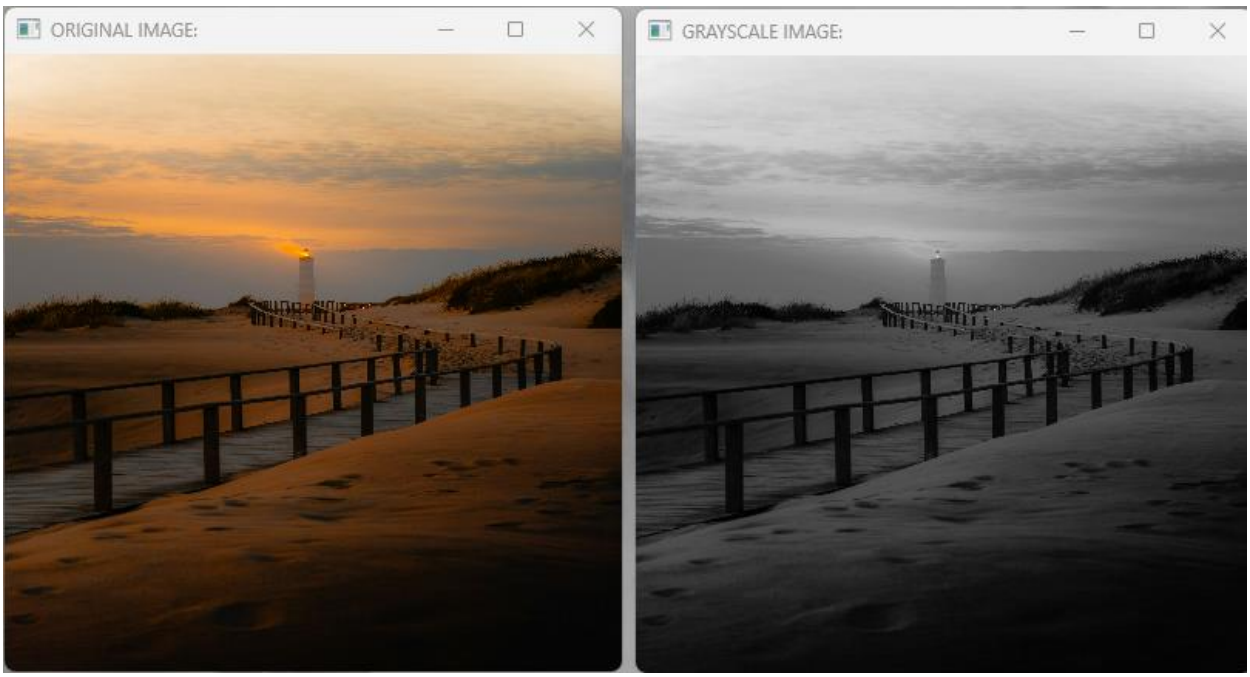
Algorithm for Grayscale Conversion

To convert an image to grayscale, you can use the following algorithm:

1. **Read the Image:** ○ Load the color image using OpenCV.
2. **Extract Color Channels:**
    - Split the image into its Blue, Green, and Red color channels.
3. **Apply Grayscale Conversion Formula:**
    - Use the formula to compute the grayscale value for each pixel: Gray=0.299×Red+0.587×Green+0.114×Blue\text{Gray} = 0.299 \times \text{Red} + 0.587 \times \text{Green} + 0.114 \times \text{Blue}Gray=0.299×Red+0.587×Green+0.114×Blue
    - The coefficients 0.299, 0.587, and 0.114 are based on the luminosity method, which weights the Red, Green, and Blue channels differently according to human perception.
4. **Combine Channels:**
    - Create a new image where all three channels (Red, Green, and Blue) have the same grayscale value.
5. **Display the Grayscale Image:**
    - Use OpenCV to display the grayscale image.

```
import cv2
```

```
import numpy as np
image=cv2.imread("image_task/images/daniel-j-schwarz-domPjJc0HzQ-unsplash.jpg")
img=cv2.resize(image,(400,400))
R,B,G=cv2.split(img)
grayimage=(0.299*R+0.587*G+0.114*B).astype(np.uint8)
gray_image=cv2.merge([grayimage])
cv2.imshow("ORIGINAL IMAGE:",img)
cv2.imshow("GRAYSCALE IMAGE:",gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```


ORIGINAL IMAGE:
GRAYSCALE IMAGE: