

Experiment-1

Basics of OpenCV, Matplotlib, Numpy – Image Read and Display Operations

Name : N U Praneeth Reddy

Reg.No: 21BAI1500

Aim: To create a program in Python to read and display images and perform simple image manipulation operations.

Outcome: Understand different image libraries in Python.

Resources Used: Anaconda Python Environment, Google Collab ,Jupyter Notebook

Theory :

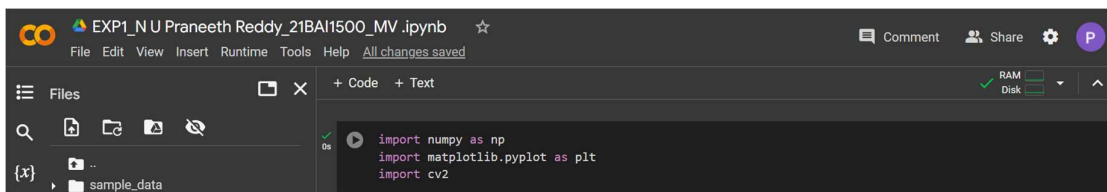
OpenCV stands as an open-source library designed for computer vision and machine learning applications. Its primary goal is to offer a unified foundation for computer vision projects and to facilitate the integration of machine perception into various commercial products.

On the other hand, NumPy serves as a Python library, enabling support for large, multi-dimensional arrays and matrices, accompanied by an extensive array of high-level mathematical functions for manipulating these arrays.

Additionally, Matplotlib functions as a Python plotting library, directly connected to the numerical mathematics capabilities of NumPy. It delivers an object-oriented API for seamlessly embedding plots within applications.

Procedure :

- Open Google Colab and create a new Jupyter Notebook.
- Import important libraries namely OpenCV, Numpy and Matplotlib.



```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

- Read the image and find its type and its resolution in pixels.

```
EXP1_N_U Praneeth Reddy_21BA11500_MV.ipynb
File Edit View Insert Runtime Tools Help All changes saved

Files
- sample_data
- fruits.png

+ Code + Text
[2] bgr_fruits = cv2.imread("fruits.png")
    if(bgr_fruits.ndim == 3):
        print("Image Type : Colour Image")
    else:
        print("Image Type : Greyscale Image")

Image Type : Colour Image

[3] shape = bgr_fruits.shape
    print("Image Resolution is :", f"{shape[0]} x {shape[1]} pixels ")

Image Resolution is : 512 x 512 pixels
```

- Show the image in BGR (Blue-Green-Red).

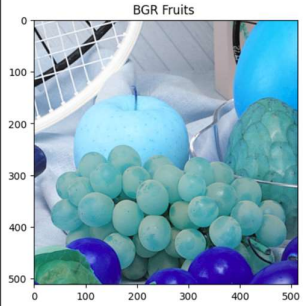
```
EXP1_N_U Praneeth Reddy_21BA11500_MV.ipynb
File Edit View Insert Runtime Tools Help All changes saved

Files
- sample_data
- fruits.png

+ Code + Text
[4] #BGR Image of Fruits
    plt.imshow(bgr_fruits)
    plt.title("BGR Fruits ")

Text(0.5, 1.0, 'BGR Fruits ')

BGR Fruits
```



- Print the maximum and minimum values of the array for the BGR image of the Fruits

```
EXP1_N_U Praneeth Reddy_21BA11500_MV.ipynb
File Edit View Insert Runtime Tools Help All changes saved

Files
- sample_data
- fruits.png

+ Code + Text
[5] print ("Maximum value of the array for the BGR Image :", np.max(bgr_fruits))
    print ("Minimum value of the array for the BGR Image :", np.min(bgr_fruits))

Maximum value of the array for the BGR Image : 255
Minimum value of the array for the BGR Image : 0
```

- Using the imread convert the image from BGR (Blue-Green-Red) to RGB (Red-Green-Blue).

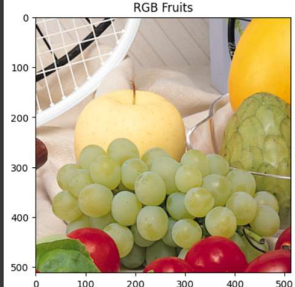
```
EXP1_N_U Praneeth Reddy_21BA11500_MV.ipynb
File Edit View Insert Runtime Tools Help All changes saved

Files
- sample_data
- fruits.png

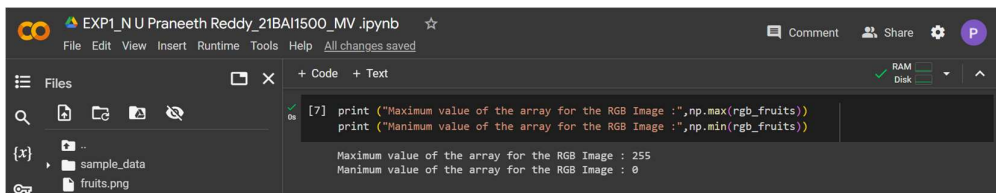
+ Code + Text
[6] # BGR to RGB Conversion
    rgb_fruits = cv2.cvtColor(bgr_fruits, cv2.COLOR_BGR2RGB)
    plt.imshow(rgb_fruits)
    plt.title("RGB Fruits")

Text(0.5, 1.0, 'RGB Fruits ')

RGB Fruits
```



- Print the maximum and minimum values of the array for the RGB image of the Fruits.



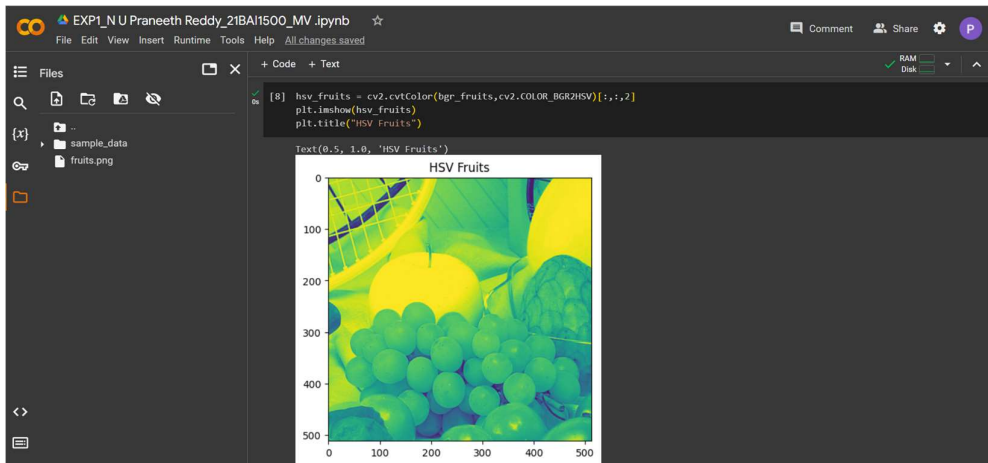
A Jupyter Notebook interface with a file explorer on the left showing 'sample_data' and 'fruits.png'. The code cell contains the following Python code:

```
[7] print ("Maximum value of the array for the RGB Image :",np.max(rgb_fruits))
     print ("Manimum value of the array for the RGB Image :",np.min(rgb_fruits))
```

The output below the code cell shows:

```
Maximum value of the array for the RGB Image : 255
Manimum value of the array for the RGB Image : 0
```

- Using the imread convert the image to the HSV version of it.

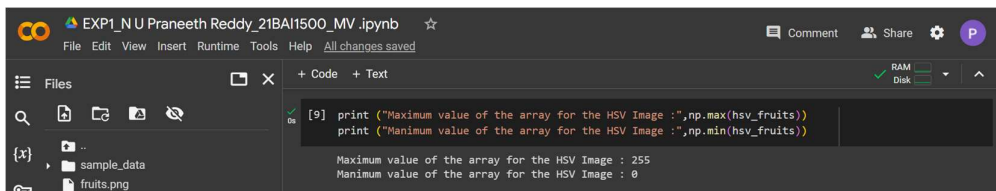


A Jupyter Notebook interface showing the conversion of an image to HSV. The code cell contains:

```
[8] hsv_fruits = cv2.cvtColor(bgr_fruits,cv2.COLOR_BGR2HSV)
     plt.imshow(hsv_fruits)
     plt.title("HSV Fruits")
```

The output is a plot titled "HSV Fruits" showing the HSV representation of the fruit image. The plot has axes ranging from 0 to 500 on both the x and y dimensions.

- Print the maximum and minimum values of the array for the HSV image of the Fruits.




A Jupyter Notebook interface showing the calculation of maximum and minimum values for the HSV image. The code cell contains:

```
[9] print ("Maximum value of the array for the HSV Image :",np.max(hsv_fruits))
     print ("Manimum value of the array for the HSV Image :",np.min(hsv_fruits))
```

The output shows:

```
Maximum value of the array for the HSV Image : 255
Manimum value of the array for the HSV Image : 0
```

- Using the imread convert the image to the gray scale version of it.

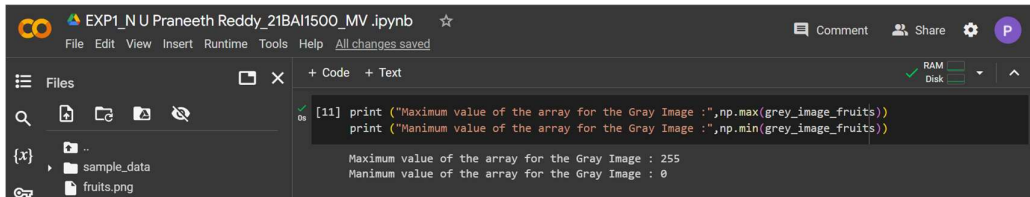


A Jupyter Notebook interface showing the conversion of an image to grayscale. The code cell contains:

```
[10] # Grey Image of the Image
      grey_image_fruits =cv2.cvtColor(bgr_fruits, cv2.COLOR_BGR2GRAY)
      plt.imshow(grey_image_fruits, "gray")
      plt.title("Grey Image of Fruits")
```

The output is a plot titled "Grey Image of Fruits" showing the grayscale version of the fruit image. The plot has axes ranging from 0 to 500 on both the x and y dimensions.

- Print the maximum and minimum values of the array for the gray Fruits image.

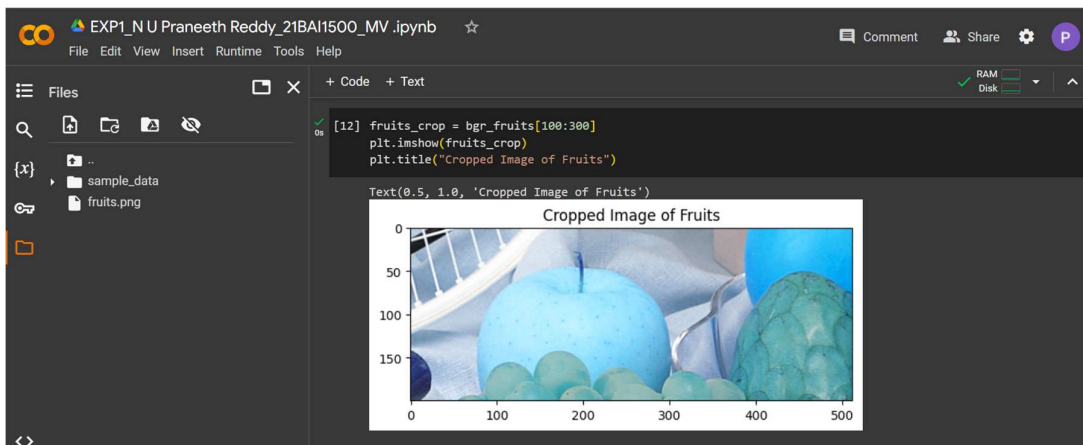


```

EXP1_N U Praneeth Reddy_21BAI1500_MV.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[11]: print ("Maximum value of the array for the Gray Image :",np.max(grey_image_fruits))
      print ("Manimum value of the array for the Gray Image :",np.min(grey_image_fruits))

Maximum value of the array for the Gray Image : 255
Manimum value of the array for the Gray Image : 0
  
```

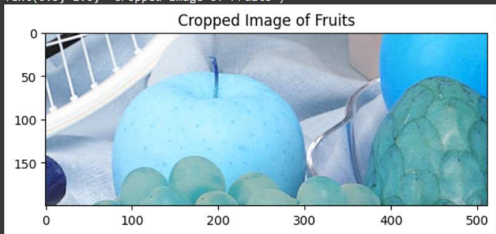
- Print the cropped image of the Fruits with only apple displayed.



```

EXP1_N U Praneeth Reddy_21BAI1500_MV.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
[12]: fruits_crop = bgr_fruits[100:300]
      plt.imshow(fruits_crop)
      plt.title("Cropped Image of Fruits")

Text(0.5, 1.0, 'Cropped Image of Fruits')
  
```



Results: The image has been converted to various forms and the maximum and minimum values of the arrays have been printed.

For BGR Image, the maximum value of the array is 255 and the minimum value is 0

For RGB Image, the maximum value of the array is 255 and the minimum value is 0

For the Blue Channel Image, the maximum value of the array is 255 and the minimum value is 0.

For the Gray Image, the maximum value of the image is 255 and the minimum value of the array is 0.

The min and Max doesn't value doesn't change for the fruits image as the values of pixel arrays can change based on the image's pixel encoding, such as in high dynamic range (HDR) or float32 images.

Conclusion: A Python program has been created to read, manipulate and display images using matplotlib, OpenCV and Numpy libraries.

Google Collab Link : <https://colab.research.google.com/drive/1mHKddvMzuH3IdLGOWSj-fSir0Sm9zyIj?usp=sharing>