Experiment-1 Worksheet

Question:

Write the python program to implement the different modules of reading and displaying the images in with all the techniques that we have learnt during the session **im.read(),im.show() and others** of any image.(format is given below)

Experiment Title :Read and Display Images

Student Name: Mulla Thanuj Kumar Reddy **UID:18BCS6189**

Branch: CSE(AIT) **Section/Group:AI&ML-2**

Semester: 5 **Date of Performance:**

Subject Name: Digital Image Processing lab **Subject Code: CSF-336**

1. Aim/Overview of the practical:

To implement the different modules of reading and Displaying image with im.read(), im.show() and with different formats.

2. The task to be done:

We need import all the required libraries for performing the tasks in different modules and we need have images in different formats.

We will read the image, show and perform task with the image in differesnt modules.

3. required libraries or software

Import these all libraries to perform tasks we will work with these in jupyter notebook.

- PIL
- Matplotlib

- Imageio
- CV2

4. Algorithm/Flowchart :

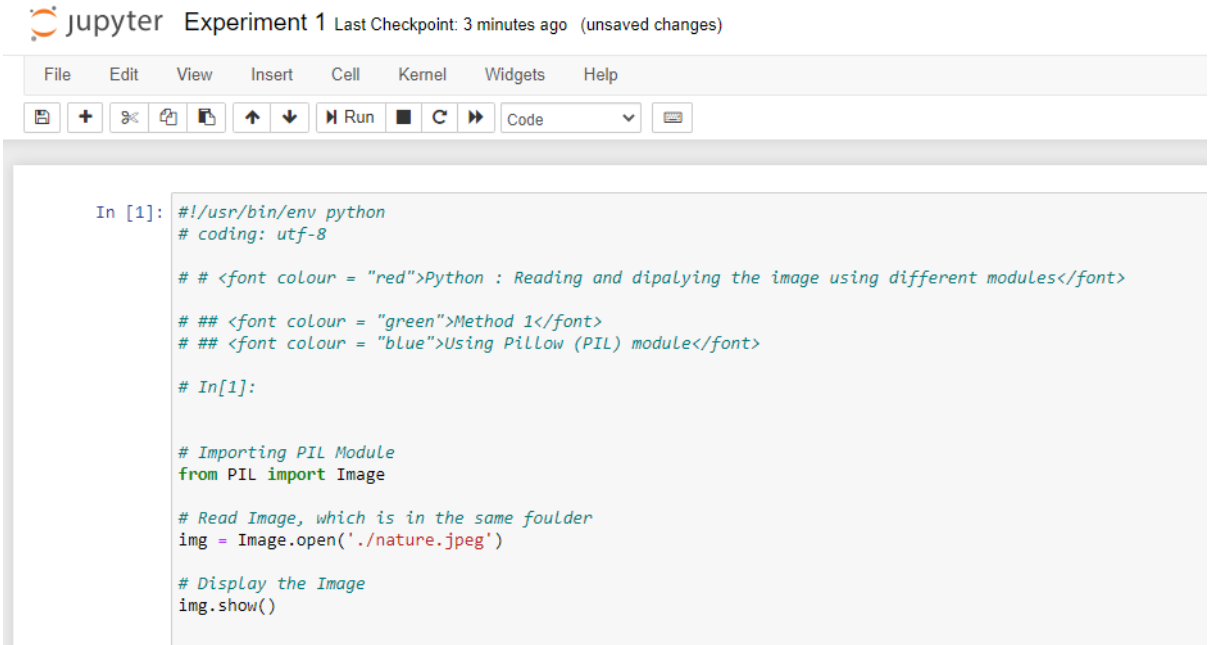
1. Import library
2. Read the image
3. Show image
4. Convert grayscale

5. Theme/Interests definition(For creative domains):

6. Steps for experiment/practical: (Step by step):

1.import PIL libraray

Get path for image and then read and show image



2. **import PIL , malplotlib libraray**

Get path for image and then read and show image

In [2]:

```
# ## <font colour = "green">Mentod 1.2</font>
# ## <font colour = "blue">Using Pillow (PIL) with matplotlib module</font>
#

# In[2]:


# Importing PIL Module
from PIL import Image
# Importing Matplotlib module
import matplotlib.pyplot as plt

# Read Image, which is in the same foulder
img = plt.imread('./nature.jpg')
# using matplotlib to display the image
plt.imshow(img)
```

Out[2]: <matplotlib.image.AxesImage at 0x26ed0546340>

3. **import matplotlib libraray**

Get path for image and then read and show image

In [3]:

```
# ## <font colour = "green">Mentod 2</font>
# ## <font colour = "blue">Using Matplotlib module</font>

# In[20]:


#Importing Matplotlib Module
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

#Read the Image , which is in the same foulder
img = mpimg.imread('./nature.jpg')

# Displaying the image using matplotlib
plt.imshow(img)
```

Out[3]: <matplotlib.image.AxesImage at 0x26ed05f2f10>

4,import imageio libraray

Get path for image and then read and show image

In [4]:
```
# ## <font colour = "green">Mentod 3</font>
# ## <font colour = "blue">Using imageio with matplotlib module</font>

# In[4]:


#Importing imageio module
import imageio
# Importing matplotlib module
import matplotlib.pyplot as plt

# Read the image using imageio
img = imageio.imread('./nature.jpg')

# Display the image using imageio
plt.imshow(img)
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x26ed074d280>
```

5.import cv2 library

Get path for image and then read and show image

In [5]:
```
# ## <font colour = "green">Mentod 4</font>
# ## <font colour = "blue">Using OpenCV module</font>

# In[16]:


# Import OpenCV-Python (cv2) Module
import cv2 as cv
# Read the Image
img = cv.imread('./nature.jpg',1)
# NB: 1 IMREAD_COLOUR IMAGE, NB:0 IMREAD_ GREYSCALE IMAGE, NB:-1 IMREAD_UNCHANGE IMAGE

# Display the image using openCV
cv.imshow('windowTitle', img)

# Display the image until you press any key
cv.waitKey(0)
```

Out[5]:
```
-1
```

6.import cv2, matplolib libraray

Get path for image and then read and show image

In [6]:
```
# ## <font colour = "green">Mentod 5</font>
# ## <font colour = "blue">Using OpenCV with Matplotlib module</font>

# In[19]:


# Import OpenCV-Python (cv2) Module
import cv2 as cv
# Importing Matplotlib Module
import matplotlib.pyplot as plt

# Read the Image in greyscale
img = cv.imread('./nature.jpg',-1)
# NB: 1 IMREAD_COLOUR IMAGE, NB:0 IMREAD_ GREYSCALE IMAGE, NB:-1 IMREAD_UNCHANGE IMAGE

# Convert GBR colour mode to RGB colour mode
RGBimg = cv.cvtColor(img, cv.COLOR_BGR2RGB)
# using matplotlib to display the image
plt.imshow(RGBimg)


# In[15]:
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x26ed0aeafd0>
```

7. import cv2, matplotlib library

Get path for image and then read and show image

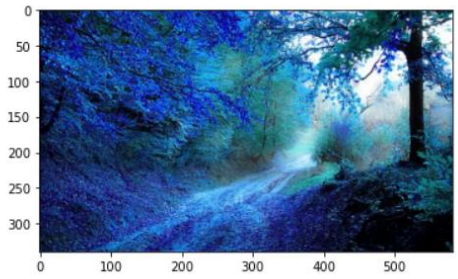
We convert into grayscale

```
In [7]: # Import OpenCV-Python (cv2) Module
        import cv2 as cv
        # Importing Matplotlib Module
        import matplotlib.pyplot as plt

        # Read the Image in greyscale
        img = cv.imread('./nature.jpg',1)
        # NB: 1 IMREAD_COLOUR IMAGE, NB:0 IMREAD_ GREYSCALE IMAGE, NB:-1 IMREAD_UNCHANGE IMAGE
        #using matplotlib to display the image
        plt.imshow(img)


        # In[ ]:

Out[7]: <matplotlib.image.AxesImage at 0x26ed0da0e50>
```



7. Observations/Discussions(For applied/experimental sciences/materials based labs):

(if the same concept had to be applied in the real-life where would you choose to apply)

This can be applied to your own images for reading and showing images.

We can use grayscale on your images and see how it looks and some more operations we can do and watch what happens to image when we apply grayscales.

8. Percentage error (if any or applicable):

NO error.

9. The command that we have learned today in the program :

- How to import libraries ex import matplotlib,pyplot as plt

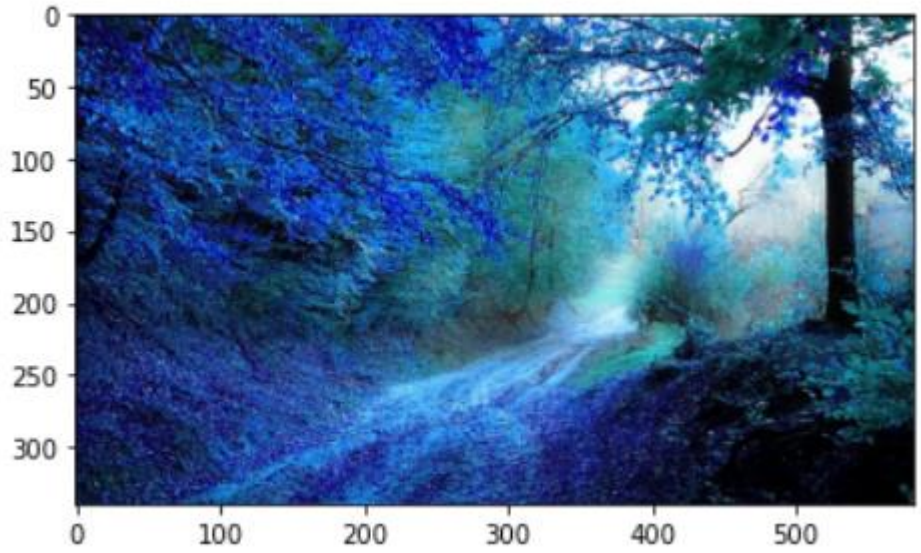
- How to read image example img = cv.imread('./nature.jpg')
- How to show image example plt.imshow(img)

10. Result/Output/Writing Summary of the concept behind the experiment:

These are the some outputs of experiment where we can see image show window page, matplotlib, grayscale images.

```
<matplotlib.image.AxesImage at 0x26ed0546340>
```

```
Out[7]: <matplotlib.image.AxesImage at 0x26ed0da0e50>
```



11. Graphs (If Any): Image /Soft copy of graph paper to be attached here

NO graphs.