

Question:

Write the python program to implement the different modules of reading and displaying the image and video using openCV in with all the techniques that we have learnt during the session.(format is given below)

Experiment Title: 2**Student Name: Pratibha****UID:18BCS6093****Branch:AIML-1****Section/Group:A****Semester:5****Date of Performance:****Subject Name:DIP LAB****Subject Code: CSF-336****1. Aim/Overview of the practical:**

WAP to convert one file format to other

2. The task to be done:

Converting from one file format to another

3. required libraries or software

```
im = Image.open("../images/parrot.png")
print(im.mode)

im.save("../images/parrot.jpg")

RGB

im = Image.open("../images/hill.png")
print(im.mode)

# RGBA

im.convert('RGB').save("../images/hill.jpg") # first convert to RGB mode

RGBA
```

4. Algorithm/Flowchart :

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

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

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# # Experiment-2 (OpenCV with Python for Image and Video Analysis)
```

```
# ## Method 1.0: Reading an image using cv2
```

```
# In[1]:
```

```
# Importing required modules
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import cv2 as cv
```

```
# Defining and reading an Image using opencv module
```

```
img = cv.imread('./moral.jpg', cv.IMREAD_GRAYSCALE)
```

```
# we can use IMREAD_GRAYSCALE for colour image instead of 0
```

```
# we can use IMREAD_COLOR for colour image instead of 1
```

```
# we can use IMREAD_UNCHANGED for colour image instead of -1
```

```
# Displaying the image
```

```
cv.imshow('Image window', img)
```

```
cv.waitKey(0)

cv.destroyAllWindows
```

Method 1.1: Reading , displaying and plotting the image using matplotlib in cv2.

```
# In[8]:
```

```
# Importing required modules
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import cv2 as cv
```

```
# Defining and reading an Image using opencv module
```

```
img = cv.imread('./moral.jpg', cv.IMREAD_GRAYSCALE)
```

```
# we can use IMREAD_GRAYSCALE for colour image instead of 0
```

```
# we can use IMREAD_COLOR for colour image instead of 1
```

```
# we can use IMREAD_UNCHANGED for colour image instead of -1
```

```
# Displaying the image using matplotlib module
```

```
plt.imshow(img, cmap='gray', interpolation='bicubic')
```

```
plt.show()
```

```
# Convert BGR colour mode to RGB colour mode
```

```
RGBimg = cv.cvtColor(img, cv.COLOR_BGR2RGB)
```

```
# To plot on the image
```

```
plt.plot([120,700] , [230,700] , 'c' , linewidth=5)
```

```
# using matplotlib to display the image
```

```
plt.imshow(RGBimg)
```

```
# Save the image
```

```
cv.imwrite('./flute.png', img)
```

```
# # Method 1.2: Reading the image and saving it.
```

```
# In[18]:
```

```
# Importing required modules
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import cv2 as cv
```

```
# Defining and reading an Image using opencv module
```

```
img = cv.imread('./moral.jpg', cv.IMREAD_GRAYSCALE)
```

```
# we can use IMREAD_GRAYSCALE for colour image instead of 0
```

```
# we can use IMREAD_COLOR for colour image instead of 1
```

```
# we can use IMREAD_UNCHANGED for colour image instead of -1
```

```
# Displaying the image
```

```
cv.imshow('Image window', img)
```

```
cv.waitKey(0)
```

```
cv.destroyAllWindows
```

```
# Save the image
```

```
cv.imwrite('./flute.png', img)
```

```
# # Working with video Source
```

```
#
```

```
# ## Method 2.0: Capture the video
```

```
# In[4]:
```

```
# Importing required modules
```

```
import numpy as np
```

```
import cv2 as cv
```

```
# To capture the video
```

```
cap = cv.VideoCapture('./video.mp4')
```

```
while(True):
```

```
    #infinite loop, ret_repeat
```

```
    ret, frame = cap.read()
```

```
    # To display the video
```

```
    cv.imshow('frame Window',frame)
```

```
    # To be broken later by a break statement
```

```
    if cv.waitKey(1) & 0xFF == ord('1'):
```

```
        break
```

```
cap.release()
```

```
cv.destroyAllWindows()
```

```
# ## Method 2.1: Capture and displaying the video in gray scale
```

```
# In[7]:
```

```
# Importing required modules
```

```
import numpy as np
```

```
import cv2 as cv
```

```
cap = cv.VideoCapture('./video.mp4')
```

```
while(True):
```

```
    #infinite loop, ret_repeat
```

```
    ret, frame = cap.read()
```

```
    #converting to gray
```

```
    grayimg = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
```

```
    # To display the video
```

```
    cv.imshow('frame Window',frame)
```

```
    cv.imshow('Gray Window', grayimg)
```

```
    # To be broken later by a break statement
```

```
    if cv.waitKey(1) & 0xFF == ord('q'):
```

```
        break
```

```
cap.release()
```

```
cv.destroyAllWindows()
```

```
# In[ ]:
```

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)

8. Percentage error (if any or applicable):

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

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

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