```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import cv2
from google.colab.patches import cv2_imshow
```

Name: Aparna Iyer PRN: 22070126017 Division: AI-ML A1 Batch: 2022-2026

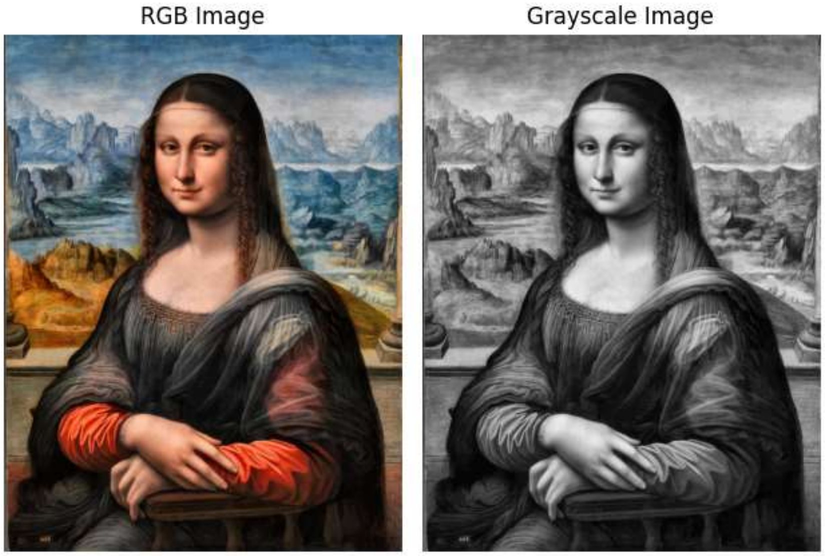
1. Reading and displaying RGB image and convert to gray-scale and display. Write programs using any two libraries. a) OpenCV b) SciKit image c) PIL

```
img=cv2.imread('/content/MonaLisa.png')
```

```
gray_img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

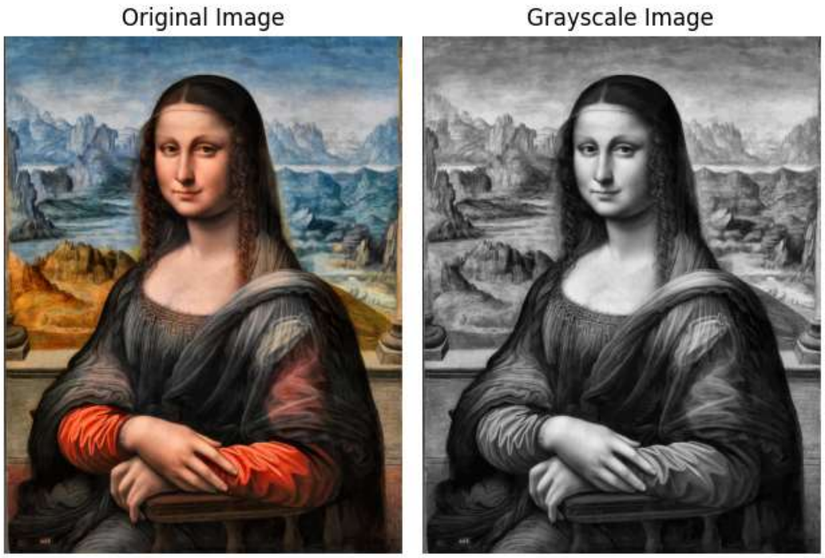
```
rgb_img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

```
plt.subplot(1,2,1)
plt.imshow(rgb_img)
plt.title('RGB Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(gray_img,cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.tight_layout()
plt.show()
```



```
#Using PIL Image Library

img=Image.open('/content/MonaLisa.png')
gray_img=img.convert('L')
```

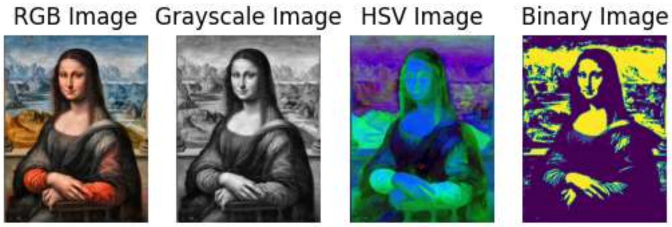
```
plt.subplot(1,2,1)
plt.imshow(img)
plt.title('Original Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(gray_img,cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.tight_layout()
plt.show()
```



Read and display RGB image and corresponding gray scale, binary image and HSV image. Use OpenCV library.

```
img=cv2.imread('/content/MonaLisa.png')

rgb_img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

gray_img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

hsv_img=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)

binary_img=cv2.threshold(gray_img,127,255,cv2.THRESH_BINARY)[1]


plt.subplot(1,4,1)
plt.imshow(rgb_img)
plt.title('RGB Image')
plt.axis('off')

plt.subplot(1,4,2)
plt.imshow(gray_img,cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')

plt.subplot(1,4,3)
plt.imshow(hsv_img)
plt.title('HSV Image')
plt.axis('off')

plt.subplot(1,4,4)
plt.imshow(binary_img)
plt.title('Binary Image')
plt.axis('off')
```

```
(-0.5, 982.5, 1279.5, -0.5)
```

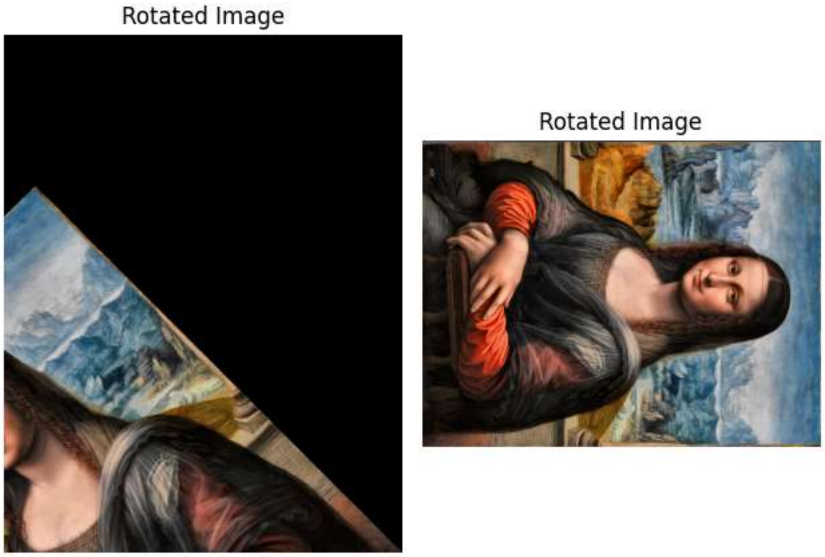


3. Write program to perform any two operations on image. a) Image Translation b) Image Rotation c) Image scaling and resizing

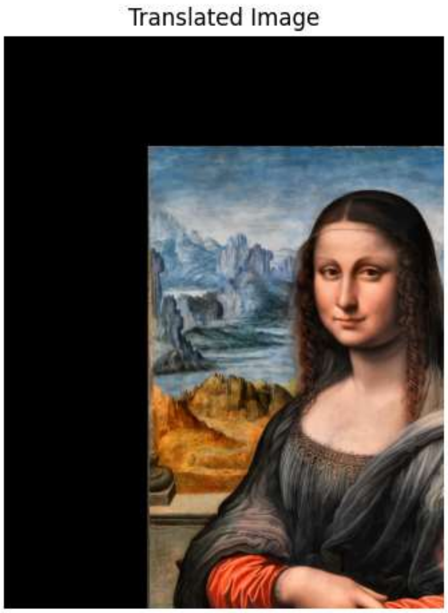
```
img=cv2.imread('/content/MonaLisa.png')
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

#Image Rotation
height,width=img.shape[0:2]


rotation_matrix=cv2.getRotationMatrix2D(center=(width,height),angle=45,scale=1)


print(rotation_matrix,"\n")
```

```
[[ 7.07106781e-01  7.07106781e-01 -6.17182646e+02]
 [-7.07106781e-01  7.07106781e-01  1.06998929e+03]]
```

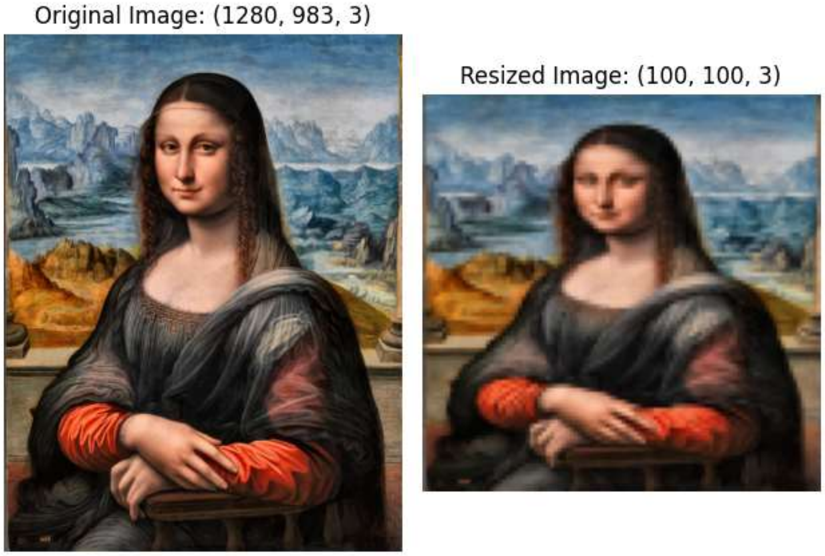
```
rotated_img=cv2.warpAffine(img,rotation_matrix,(width,height))


rotated_img2=cv2.rotate(img,cv2.ROTATE_90_CLOCKWISE)


plt.subplot(1,2,1)
plt.imshow(rotated_img)
plt.title('Rotated Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(rotated_img2)
plt.title('Rotated Image')
plt.axis('off')

plt.tight_layout()
plt.show()
```



```
#Image Translation

translation_matrix=np.array([[1,0,(height/4)],[0,1,(width/4)]],dtype=float)


print(translation_matrix,"\n")

     [[  1.     0.   320.  ]
      [  0.     1.   245.75]]



translated_img=cv2.warpAffine(img,translation_matrix,(width,height))


plt.imshow(translated_img)
plt.title("Translated Image")
plt.axis('off')
plt.tight_layout()
plt.show()
```

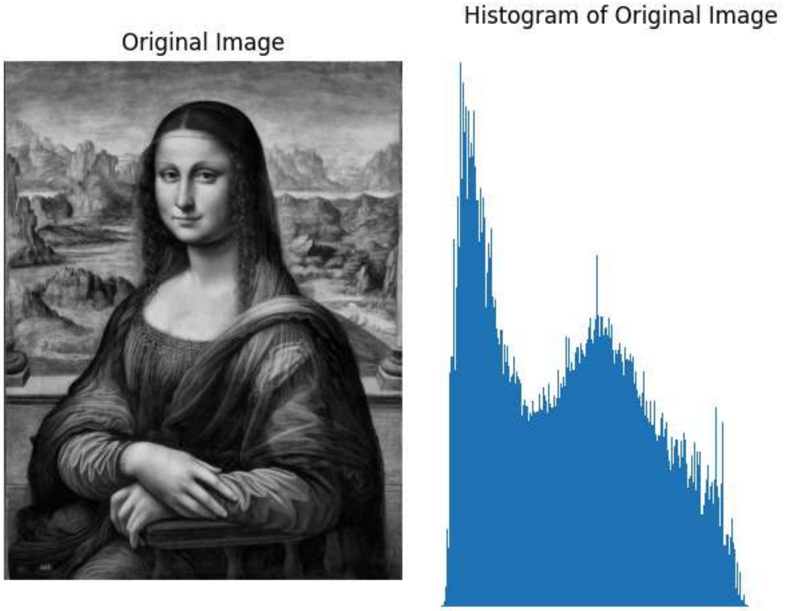


```
#Image Resizing

resized_img=cv2.resize(img,(100,100),interpolation=cv2.INTER_AREA)


plt.subplot(1,2,1)
plt.imshow(img)
plt.title(f'Original Image: {img.shape}')
plt.axis('off')

plt.subplot(1,2,2)
plt.imshow(resized_img)
plt.title(f'Resized Image: {resized_img.shape}')
plt.axis('off')

plt.tight_layout()
plt.show()
```

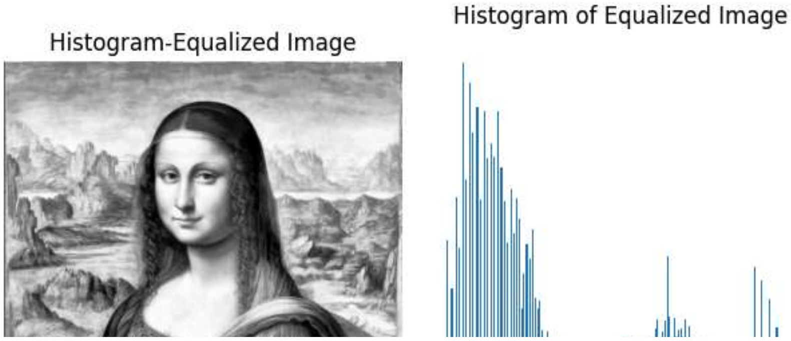


4. Write a program to plot histogram of original image. Perform histogram equalization and display histogram equalized image and histogram.

```
img=cv2.imread('/content/MonaLisa.png')
img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
i2=cv2.equalizeHist(img) #Histogram-Equalized Image


plt.subplot(1,2,1)
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(1,2,2)
plt.hist(img.ravel(),256,[0,256])
plt.title('Histogram of Original Image')
plt.axis('off')

plt.tight_layout()
plt.show()
```



```
plt.subplot(1,2,1)
plt.imshow(cv2.cvtColor(i2,cv2.COLOR_BGR2RGB))
plt.title('Histogram-Equalized Image')
plt.axis('off')
plt.subplot(1,2,2)
plt.hist(i2.ravel(),256,[0,256])
plt.title('Histogram of Equalized Image')
plt.axis('off')

plt.tight_layout()
plt.show()
```


Histogram-Equalized Image
Histogram of Equalized Image