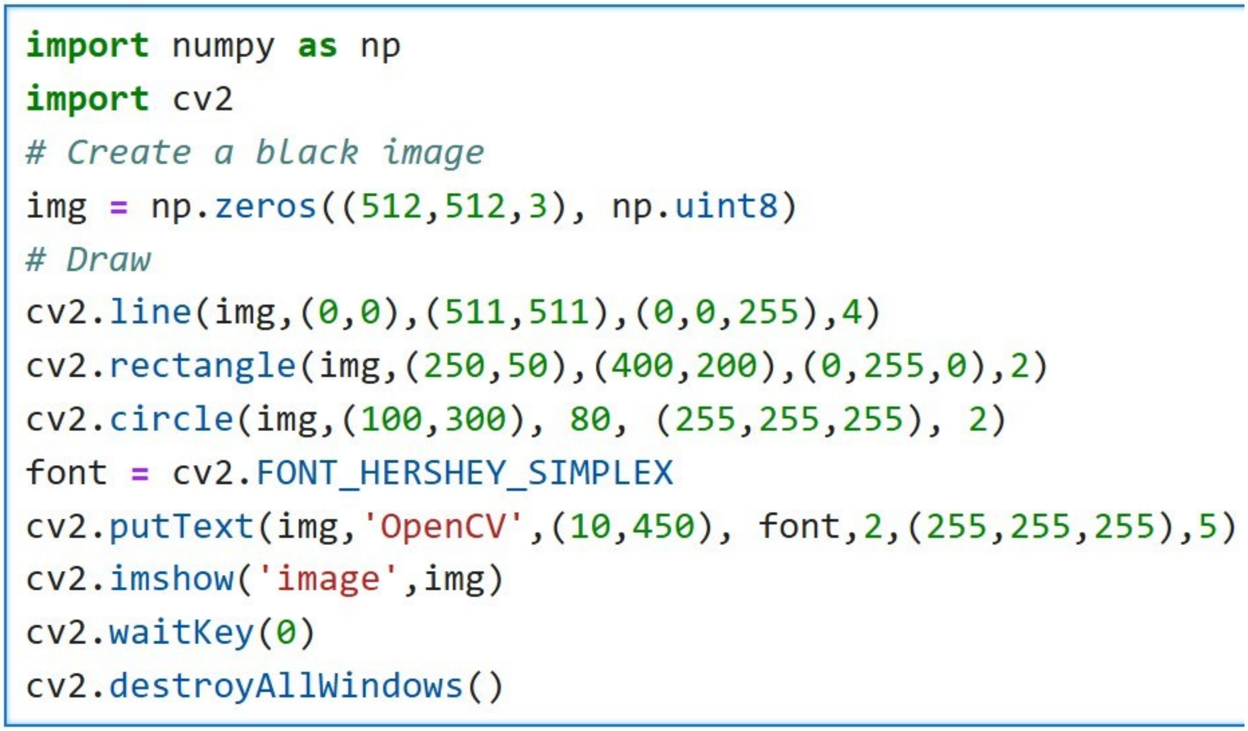
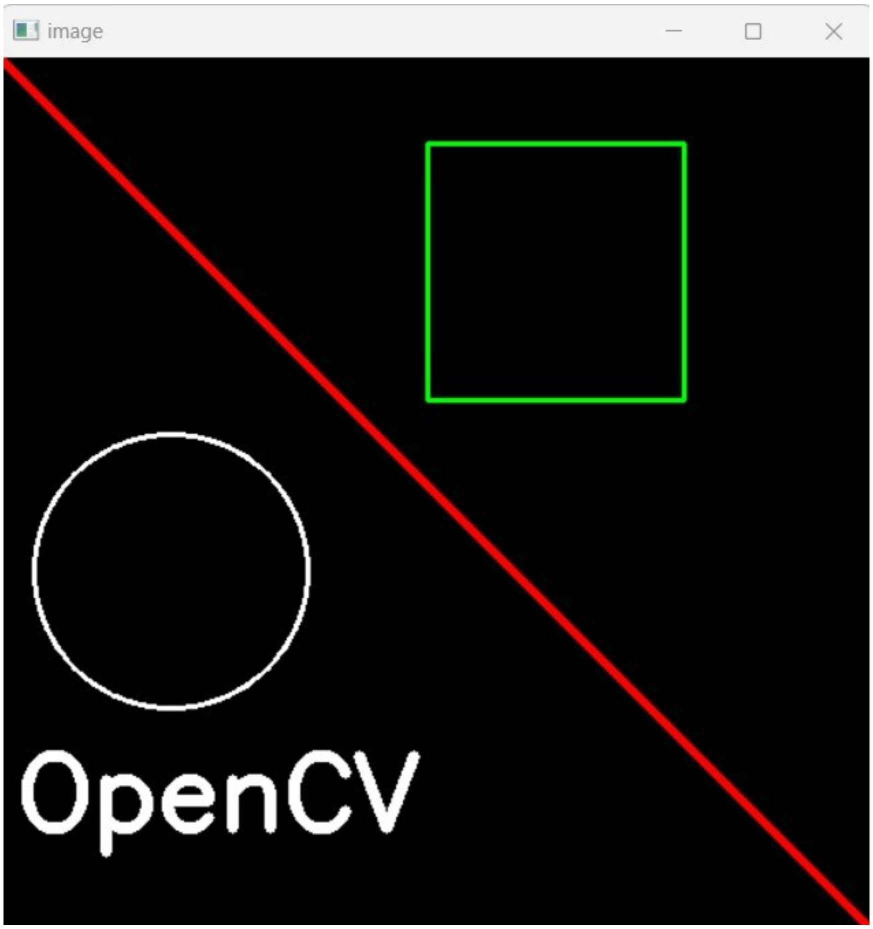
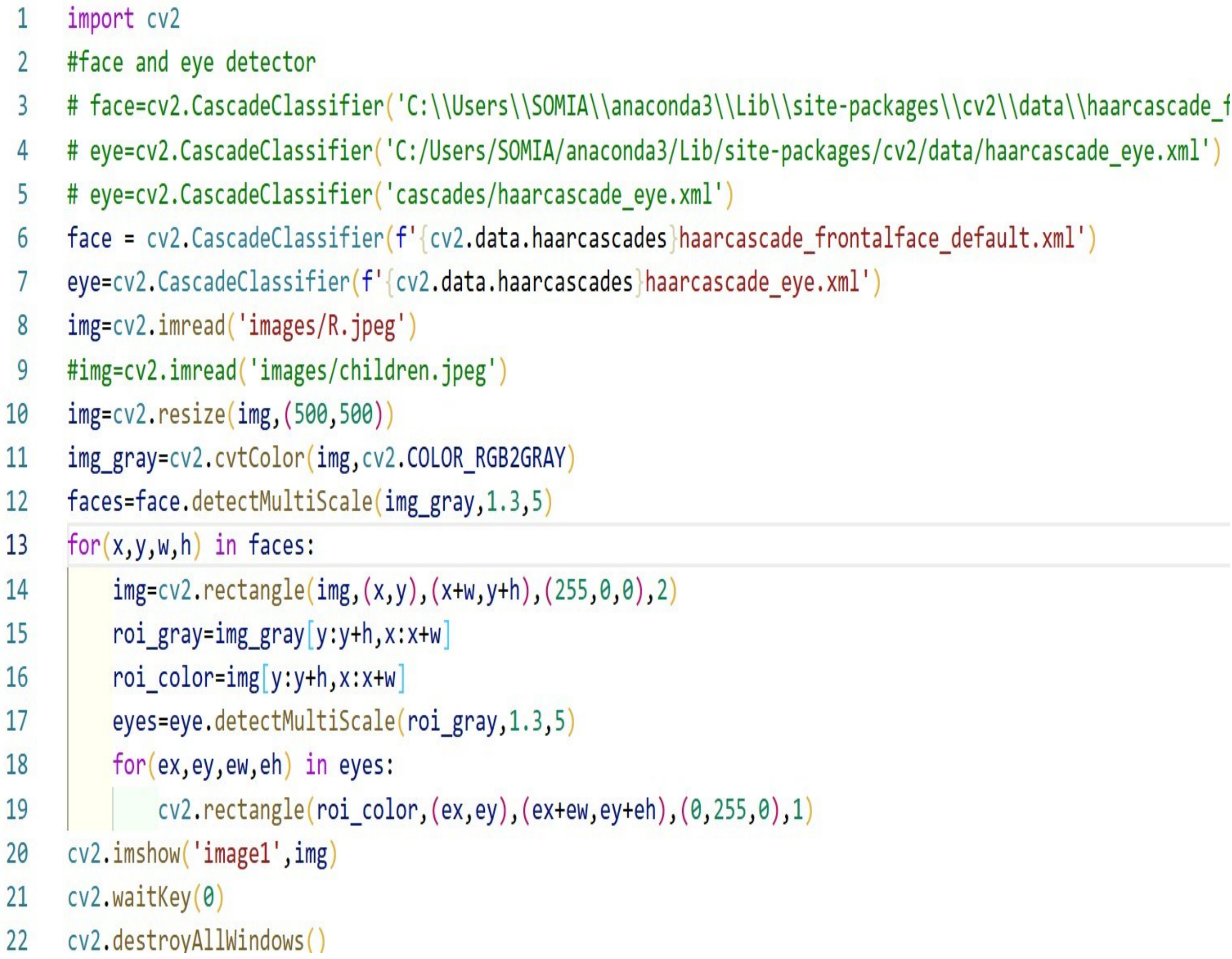
Draw- Image Algebra



```python
import numpy as np
import cv2
# Create a black image
img = np.zeros((512,512,3), np.uint8)
# Draw
cv2.line(img,(0,0),(511,511),(0,0,255),4)
cv2.rectangle(img,(250,50),(400,200),(0,255,0),2)
cv2.circle(img,(100,300), 80, (255,255,255), 2)
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img,'OpenCV',(10,450), font,2,(255,255,255),5)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

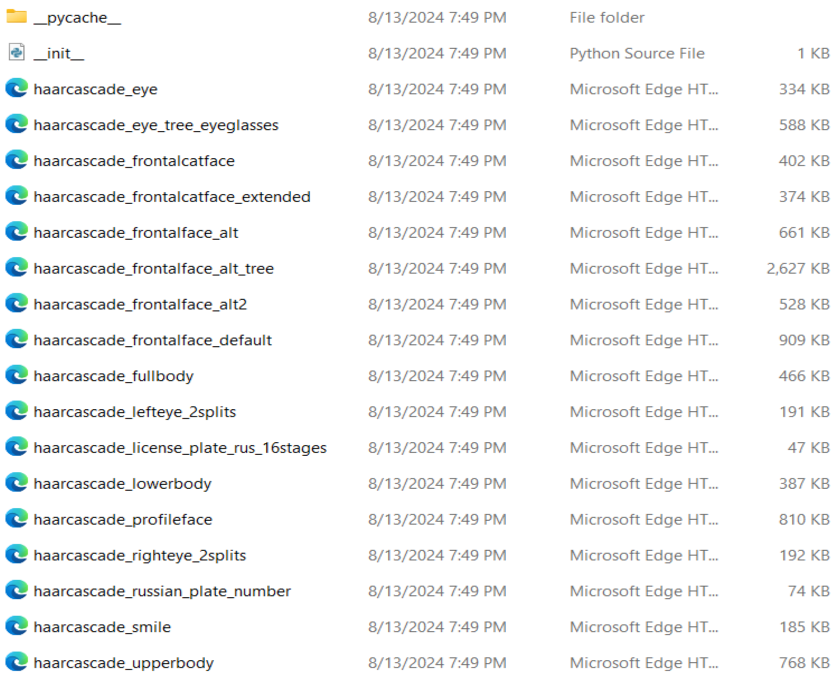


Face and Eye Detectors

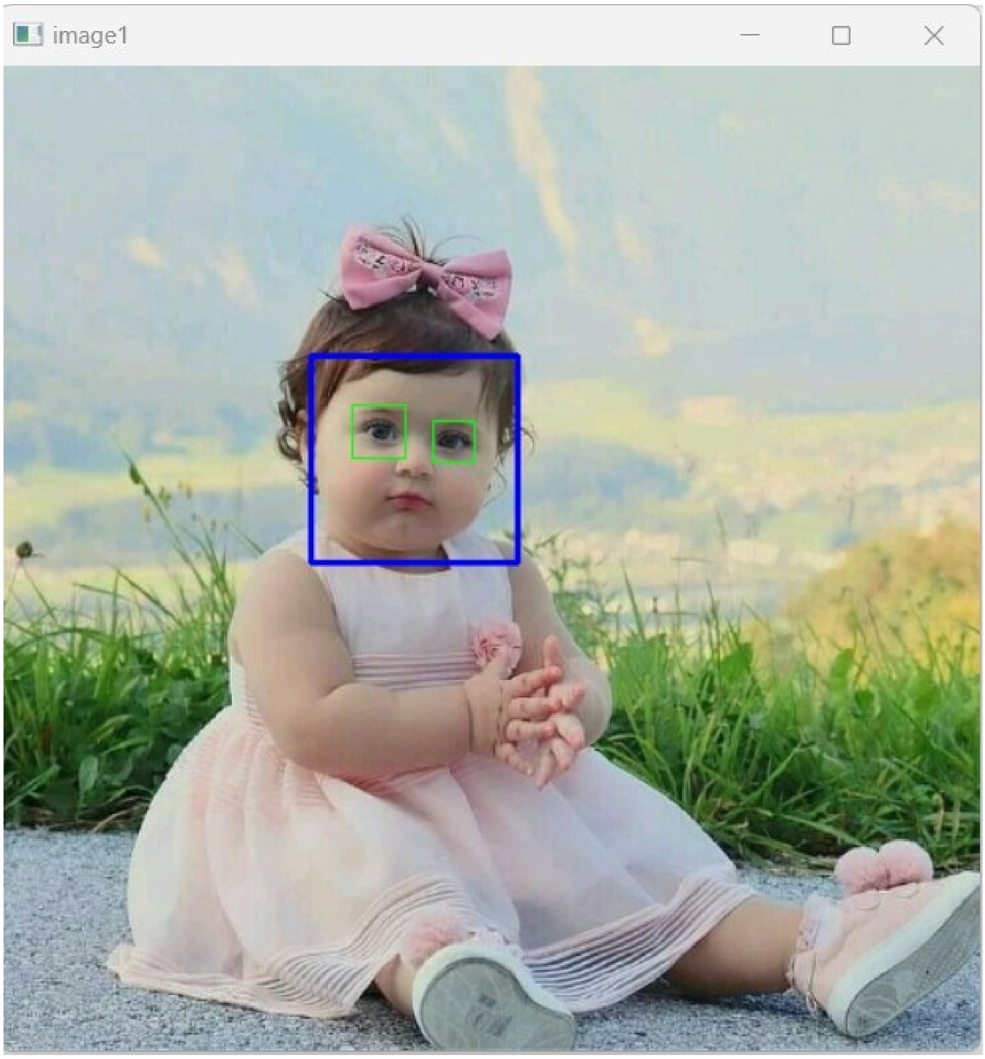


```python
import cv2
#face and eye detector
# face=cv2.CascadeClassifier('C:\\Users\\SOMIA\\anaconda3\\Lib\\site-packages\\cv2\\data\\haarcascade_f
# eye=cv2.CascadeClassifier('C:/Users/SOMIA/anaconda3/Lib/site-packages/cv2/data/haarcascade_eye.xml')
# eye=cv2.CascadeClassifier('cascades/haarcascade_eye.xml')
face = cv2.CascadeClassifier(f'{cv2.data.haarcascades}haarcascade_frontalface_default.xml')
eye=cv2.CascadeClassifier(f'{cv2.data.haarcascades}haarcascade_eye.xml')
img=cv2.imread('images/R.jpeg')
#img=cv2.imread('images/children.jpeg')
img=cv2.resize(img,(500,500))
img_gray=cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
faces=face.detectMultiScale(img_gray,1.3,5)
for(x,y,w,h) in faces:
    img=cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray=img_gray[y:y+h,x:x+w]
    roi_color=img[y:y+h,x:x+w]
    eyes=eye.detectMultiScale(roi_gray,1.3,5)
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),1)
cv2.imshow('image1',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

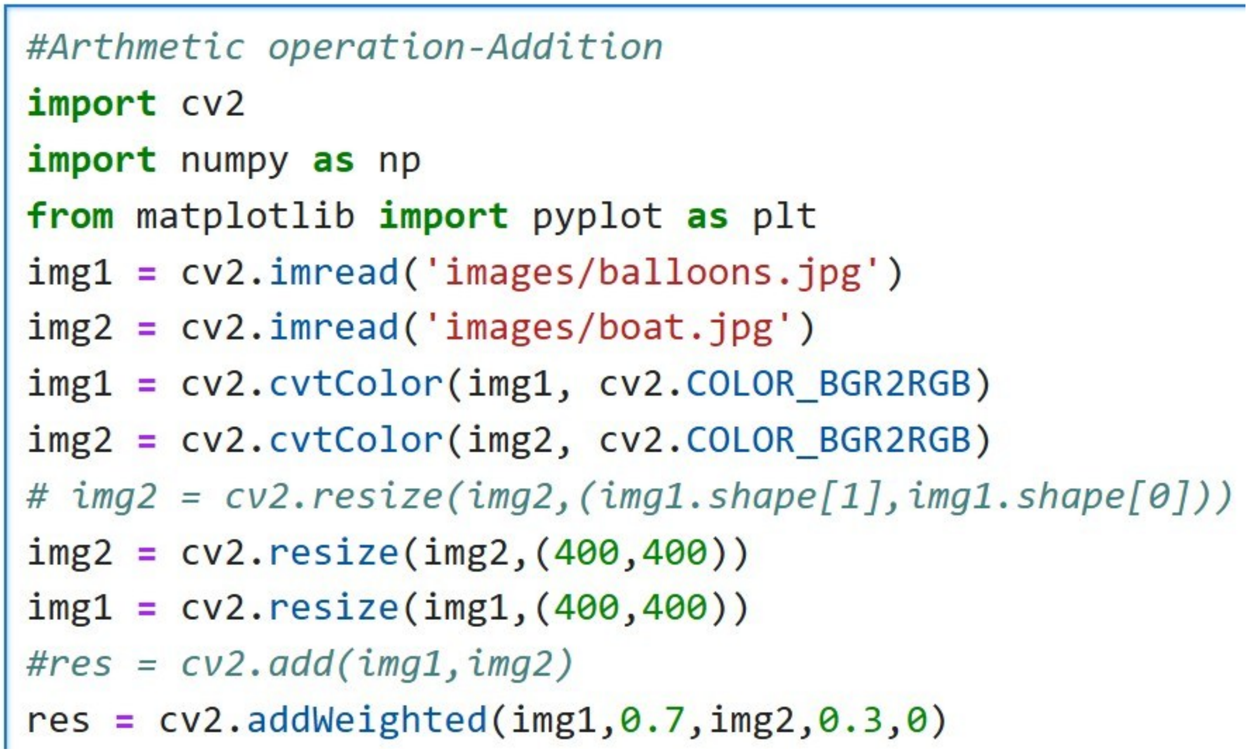
Image processing – OpenCV : LAB3



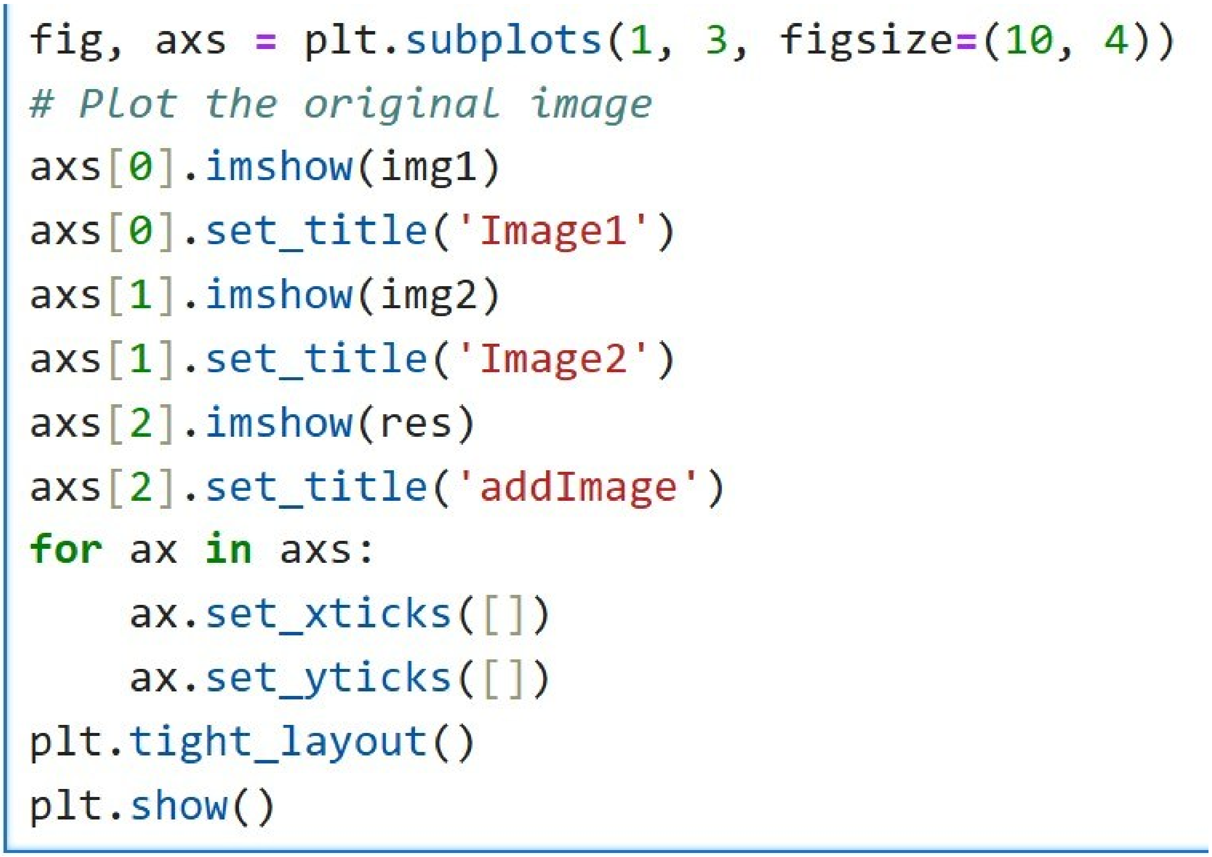
__pycache__	8/13/2024 7:49 PM	File folder	
__init__	8/13/2024 7:49 PM	Python Source File	1 KB
haarcascade_eye	8/13/2024 7:49 PM	Microsoft Edge HT...	334 KB
haarcascade_eye_tree_eyeglasses	8/13/2024 7:49 PM	Microsoft Edge HT...	588 KB
haarcascade_frontalcatface	8/13/2024 7:49 PM	Microsoft Edge HT...	402 KB
haarcascade_frontalcatface_extended	8/13/2024 7:49 PM	Microsoft Edge HT...	374 KB
haarcascade_frontalface_alt	8/13/2024 7:49 PM	Microsoft Edge HT...	661 KB
haarcascade_frontalface_alt_tree	8/13/2024 7:49 PM	Microsoft Edge HT...	2,627 KB
haarcascade_frontalface_alt2	8/13/2024 7:49 PM	Microsoft Edge HT...	528 KB
haarcascade_frontalface_default	8/13/2024 7:49 PM	Microsoft Edge HT...	909 KB
haarcascade_fullbody	8/13/2024 7:49 PM	Microsoft Edge HT...	466 KB
haarcascade_lefteye_2splits	8/13/2024 7:49 PM	Microsoft Edge HT...	191 KB
haarcascade_license_plate_rus_16stages	8/13/2024 7:49 PM	Microsoft Edge HT...	47 KB
haarcascade_lowerbody	8/13/2024 7:49 PM	Microsoft Edge HT...	387 KB
haarcascade_profileface	8/13/2024 7:49 PM	Microsoft Edge HT...	810 KB
haarcascade_righteye_2splits	8/13/2024 7:49 PM	Microsoft Edge HT...	192 KB
haarcascade_russian_plate_number	8/13/2024 7:49 PM	Microsoft Edge HT...	74 KB
haarcascade_smile	8/13/2024 7:49 PM	Microsoft Edge HT...	185 KB
haarcascade_upperbody	8/13/2024 7:49 PM	Microsoft Edge HT...	768 KB



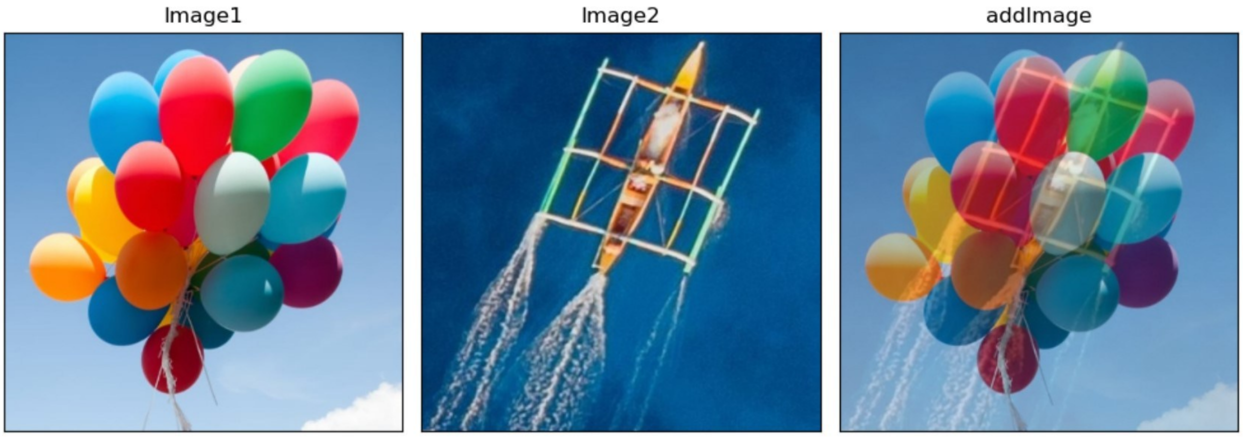
T: Somia AL-Shibah

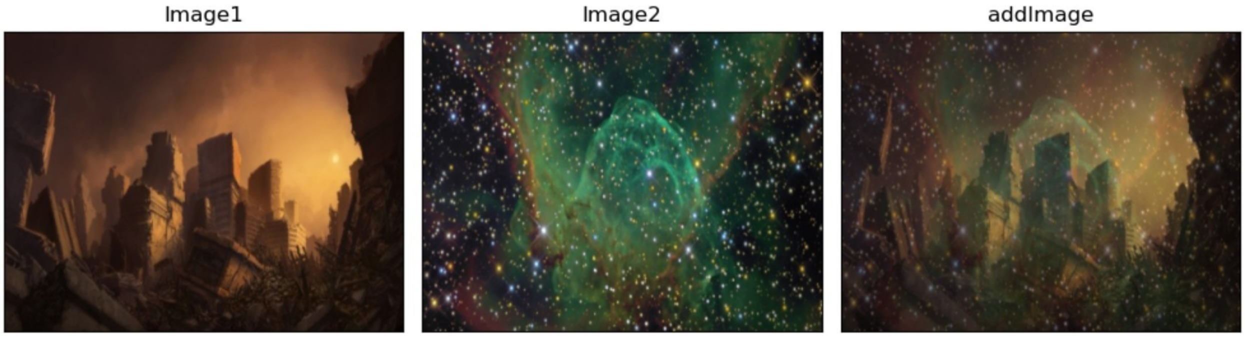


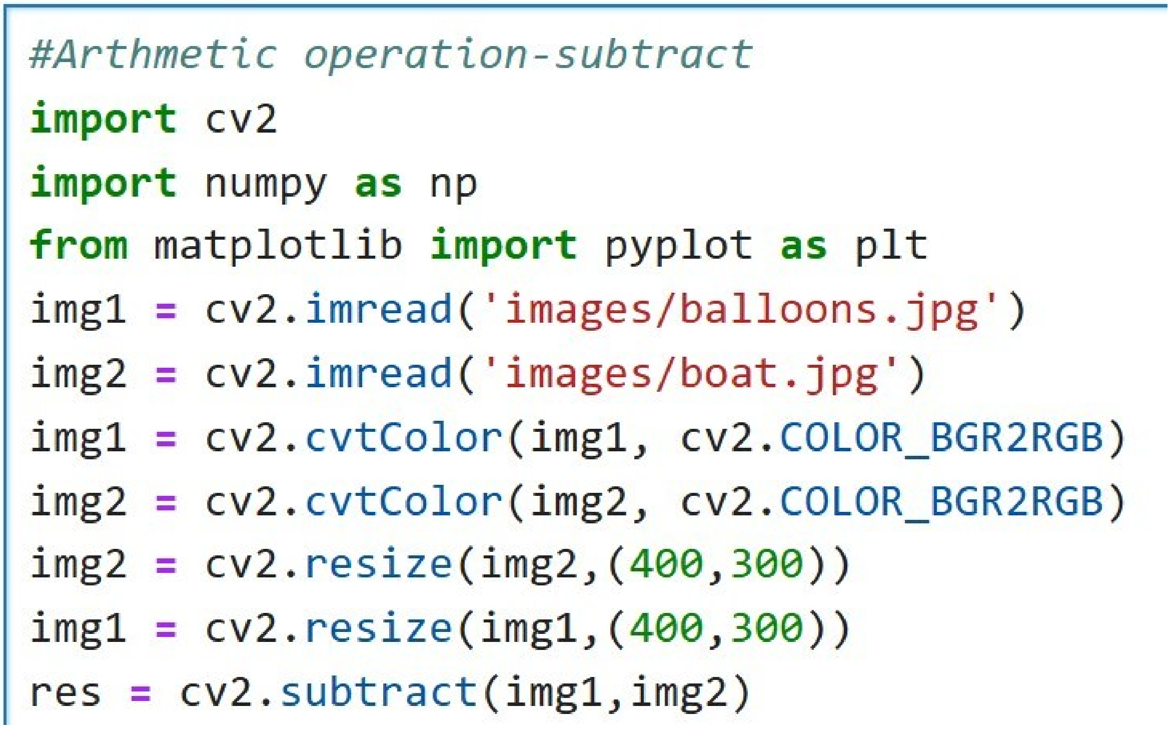
```
#Arthmetic operation-Addition
import cv2
import numpy as np
from matplotlib import pyplot as plt
img1 = cv2.imread('images/balloons.jpg')
img2 = cv2.imread('images/boat.jpg')
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
# img2 = cv2.resize(img2,(img1.shape[1],img1.shape[0]))
img2 = cv2.resize(img2,(400,400))
img1 = cv2.resize(img1,(400,400))
#res = cv2.add(img1,img2)
res = cv2.addWeighted(img1,0.7,img2,0.3,0)
```



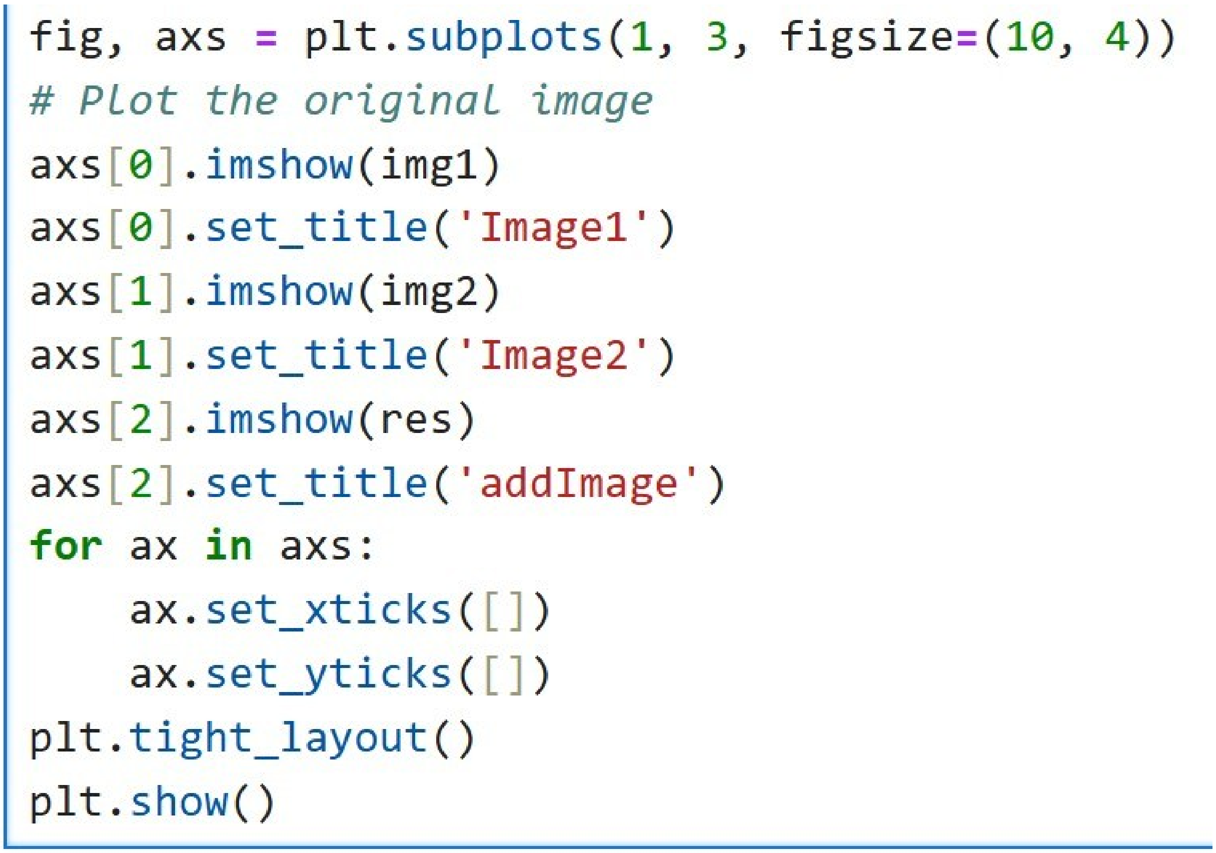
```
fig, axs = plt.subplots(1, 3, figsize=(10, 4))
# Plot the original image
axs[0].imshow(img1)
axs[0].set_title('Image1')
axs[1].imshow(img2)
axs[1].set_title('Image2')
axs[2].imshow(res)
axs[2].set_title('addImage')
for ax in axs:
    ax.set_xticks([])
    ax.set_yticks([])
plt.tight_layout()
plt.show()
```



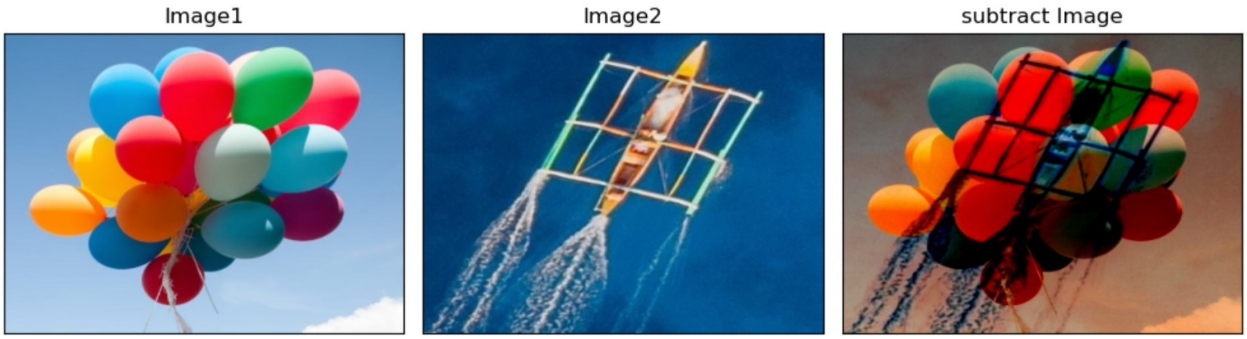


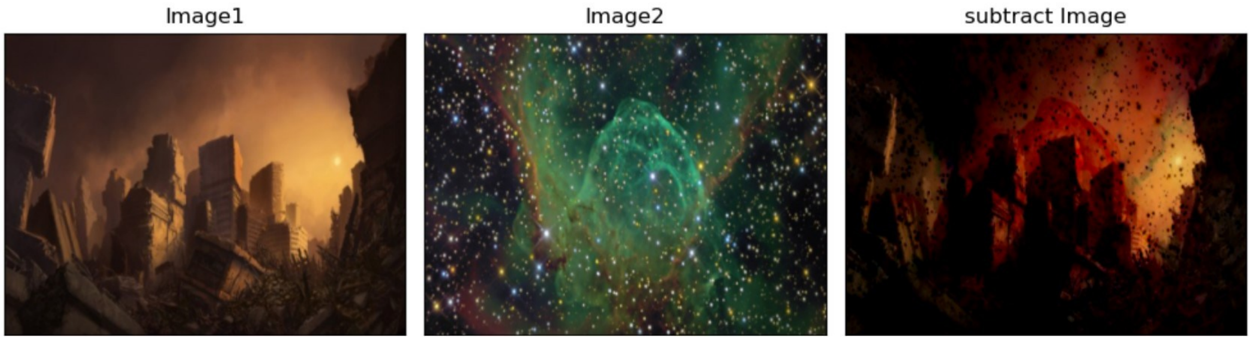


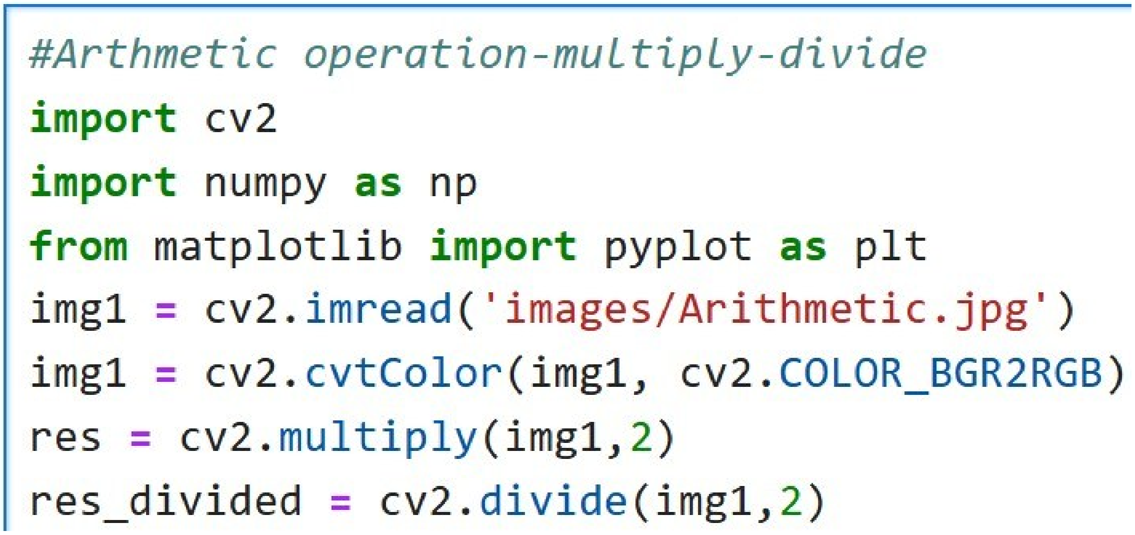
```
#Arthmetic operation-subtract
import cv2
import numpy as np
from matplotlib import pyplot as plt
img1 = cv2.imread('images/balloons.jpg')
img2 = cv2.imread('images/boat.jpg')
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
img2 = cv2.resize(img2,(400,300))
img1 = cv2.resize(img1,(400,300))
res = cv2.subtract(img1,img2)
```



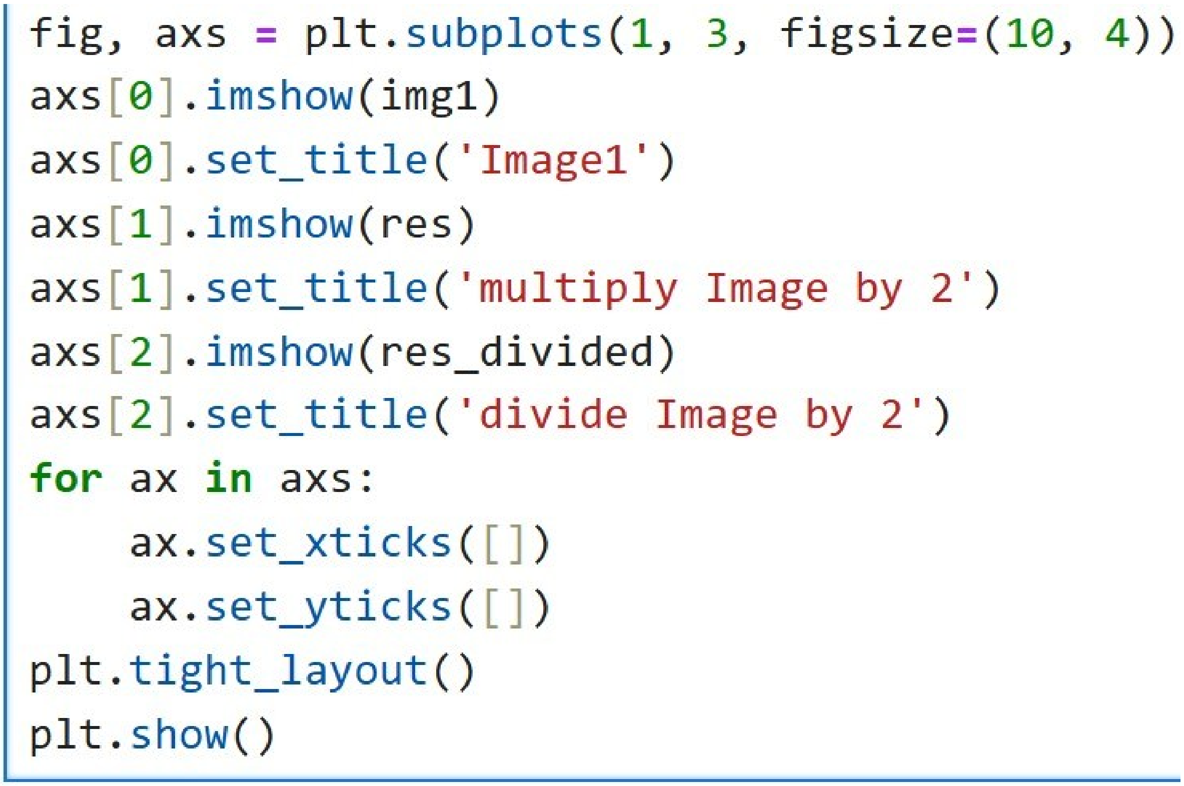
```
fig, axs = plt.subplots(1, 3, figsize=(10, 4))
# Plot the original image
axs[0].imshow(img1)
axs[0].set_title('Image1')
axs[1].imshow(img2)
axs[1].set_title('Image2')
axs[2].imshow(res)
axs[2].set_title('addImage')
for ax in axs:
    ax.set_xticks([])
    ax.set_yticks([])
plt.tight_layout()
plt.show()
```



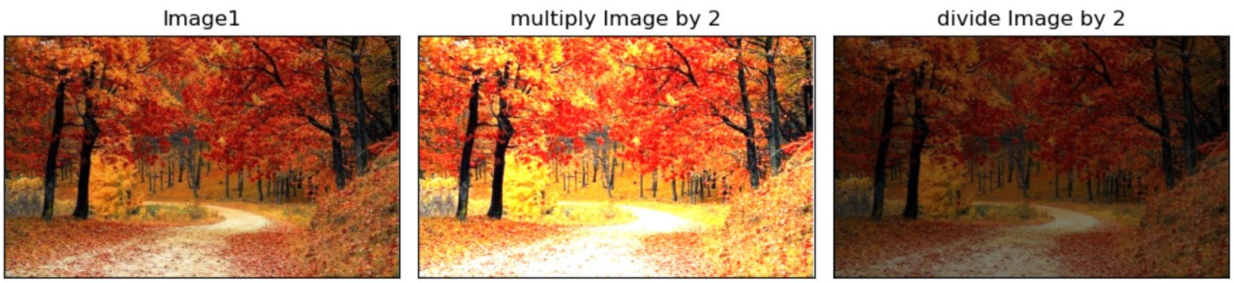


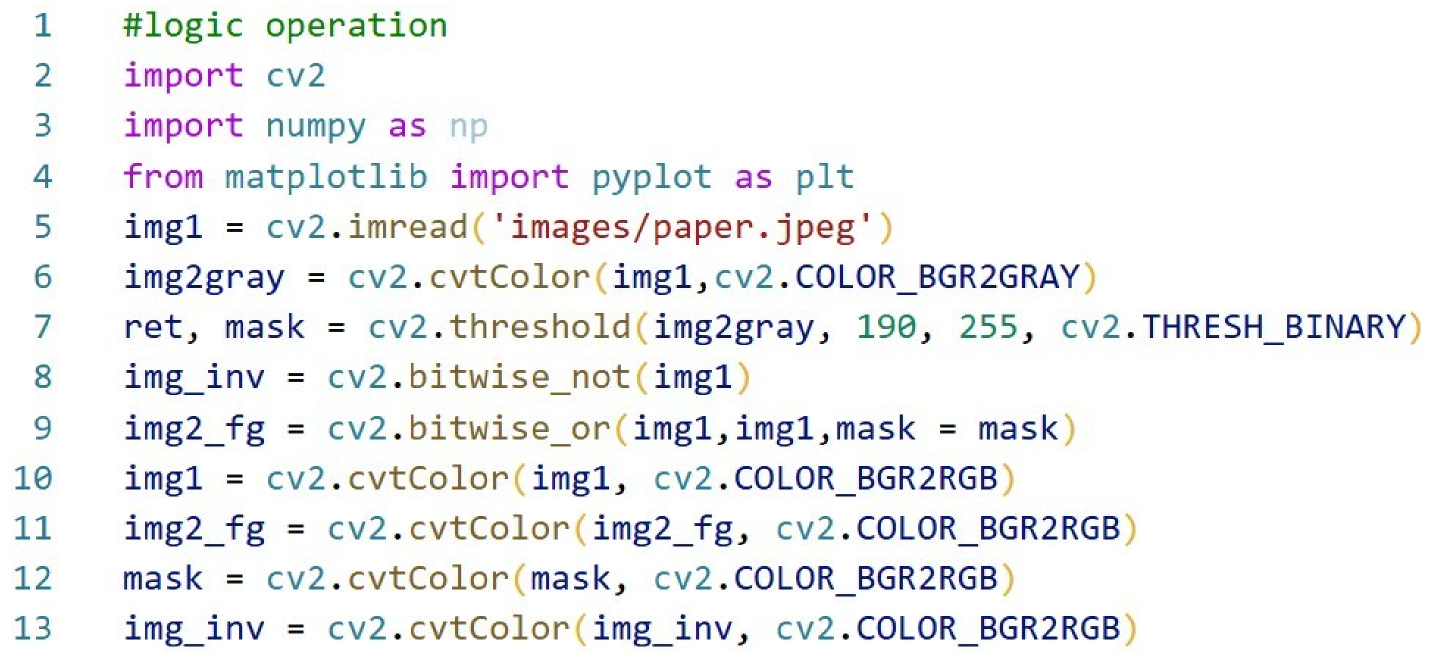


```
#Arthmetic operation-multiply-divide
import cv2
import numpy as np
from matplotlib import pyplot as plt
img1 = cv2.imread('images/Arithmetic.jpg')
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
res = cv2.multiply(img1,2)
res_divided = cv2.divide(img1,2)
```

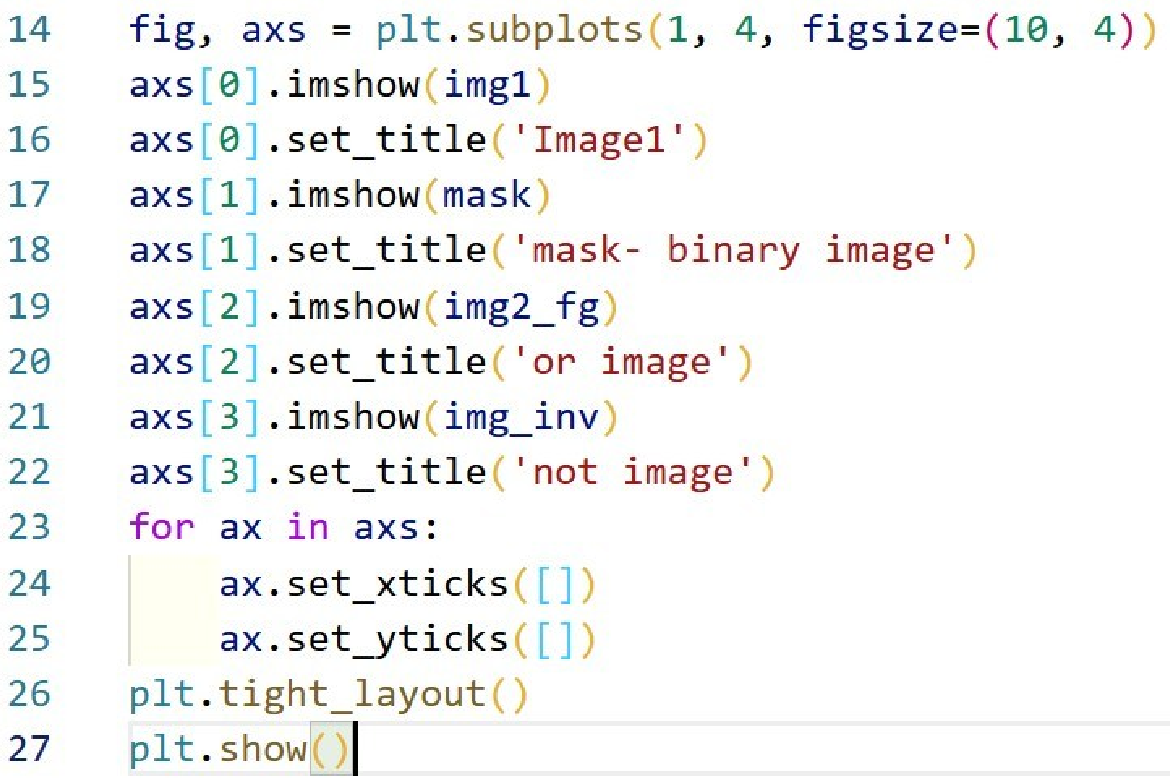


```
fig, axs = plt.subplots(1, 3, figsize=(10, 4))
axs[0].imshow(img1)
axs[0].set_title('Image1')
axs[1].imshow(res)
axs[1].set_title('multiply Image by 2')
axs[2].imshow(res_divided)
axs[2].set_title('divide Image by 2')
for ax in axs:
    ax.set_xticks([])
    ax.set_yticks([])
plt.tight_layout()
plt.show()
```





```python
#logic operation
import cv2
import numpy as np
from matplotlib import pyplot as plt
img1 = cv2.imread('images/paper.jpeg')
img2gray = cv2.cvtColor(img1,cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(img2gray, 190, 255, cv2.THRESH_BINARY)
img_inv = cv2.bitwise_not(img1)
img2_fg = cv2.bitwise_or(img1,img1,mask = mask)
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img2_fg = cv2.cvtColor(img2_fg, cv2.COLOR_BGR2RGB)
mask = cv2.cvtColor(mask, cv2.COLOR_BGR2RGB)
img_inv = cv2.cvtColor(img_inv, cv2.COLOR_BGR2RGB)
```



```python
fig, axs = plt.subplots(1, 4, figsize=(10, 4))
axs[0].imshow(img1)
axs[0].set_title('Image1')
axs[1].imshow(mask)
axs[1].set_title('mask- binary image')
axs[2].imshow(img2_fg)
axs[2].set_title('or image')
axs[3].imshow(img_inv)
axs[3].set_title('not image')
for ax in axs:
    ax.set_xticks([])
    ax.set_yticks([])
plt.tight_layout()
plt.show()
```