|  |  |
| --- | --- |
| **EXP NO:01** |  |
| **DATE:** |

AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
  
  
def main():  
 image = cv2.imread("C:\\Users\\CSE\\Pictures\\Screenshots\\hi.png")  
 if image is None:  
 print("error:could not read the image")  
 return  
 res\_image = cv2.resize(image, (500, 500))  
 cv2.imwrite("resized\_image.jpg", res\_image)  
 print("resized image saved")  
 cv2.imshow("original image", image)  
 cv2.imshow("resized image", res\_image)  
 cv2.waitKey()  
 cv2.destroyAllWindows()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| **EXP NO:02** |  |
| **DATE:** |

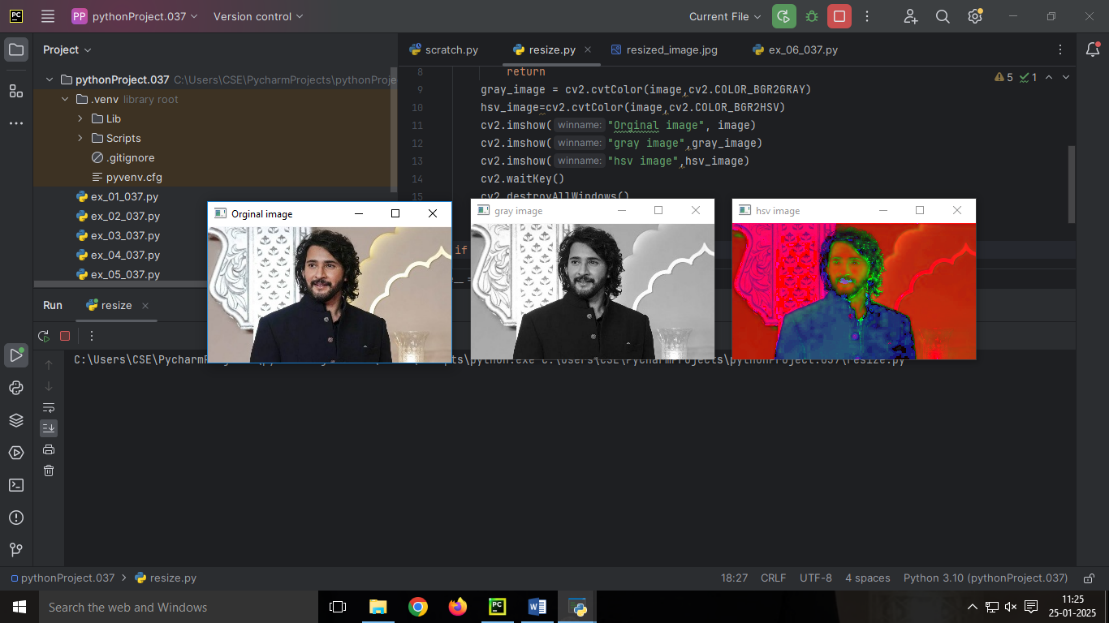
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
  
  
def main():  
 image = cv2.imread("resized\_image.jpg")  
 if image is None:  
 print("error:could not read the image")  
 return  
 gray\_image = cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)  
 hsv\_image=cv2.cvtColor(image,cv2.COLOR\_BGR2HSV)  
 cv2.imshow("Orginal image", image)  
 cv2.imshow("gray image",gray\_image)  
 cv2.imshow("hsv image",hsv\_image)  
 cv2.waitKey()  
 cv2.destroyAllWindows()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| **EXP NO:3** |  |
| **DATE:** |

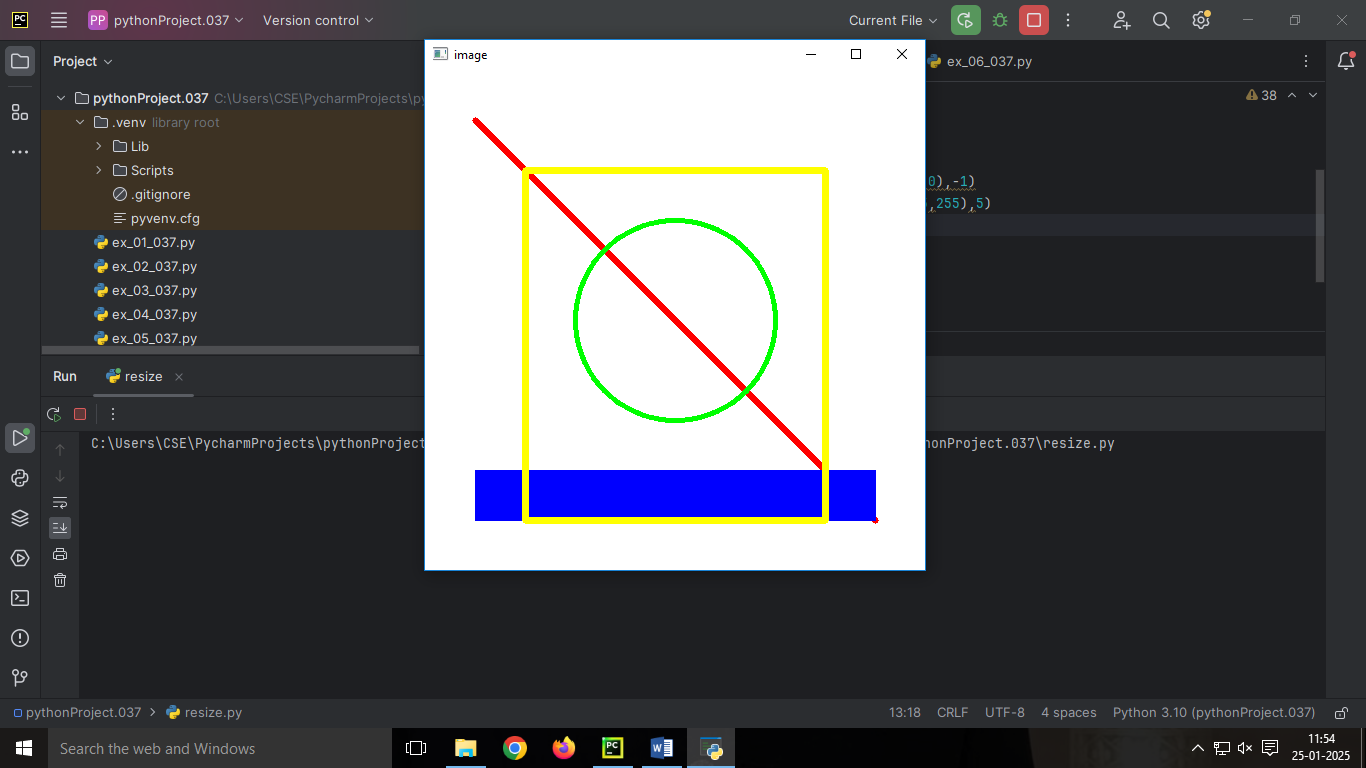
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
import numpy as np  
image = np.ones((500,500,3),dtype='uint8')\* 255  
cv2.line(image,(50,50),(450,450),(0,0,255),5)  
cv2.circle(image,(250,250),100,(0,255,0),3)  
cv2.rectangle(image,(50,400),(450,450),(255,0,0),-1)  
cv2.rectangle(image,(100,100),(400,450),(0,255,255),5)  
cv2.imshow('image',image)  
cv2.waitKey()  
cv2.destroyAllWindows()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| **EXP NO:4** |  |
| **DATE:** |

AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
from PIL import Image  
image = cv2.imread('resized\_image.jpg')  
print("IMAGE HEIGHT:",image.shape[0])  
print("IMAGE WIDTH:",image.shape[1])  
print("NUMBER OF CHANNELS",image.shape[2])  
print("IMAGE SIZE:(IN PIXELS)",image.size)  
print("IMAGE DATA TYPE:",image.dtype)  
image=Image.open('resized\_image.jpg')  
print("image Format",image.format)

OUTPUT:

IMAGE HEIGHT: 168

IMAGE WIDTH: 300

NUMBER OF CHANNELS 3

IMAGE SIZE:(IN PIXELS) 151200

IMAGE DATA TYPE: uint8

image Format JPEG

RESULT:

|  |  |
| --- | --- |
| **EXP NO:5** |  |
| **DATE:** |

AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
image = cv2.imread('resized\_image.jpg')  
res\_img=cv2.resize(image,(400,350))  
text='MAHESH:'  
position=(30,50)  
font=(cv2.FONT\_HERSHEY\_SIMPLEX)  
font\_scale=1  
color=(255,0,0)  
thickness=2  
cv2.putText(res\_img,text,position,font,font\_scale,color,thickness)  
cv2.imshow('image with text',res\_img)  
cv2.waitKey()  
cv2.destroyAllWindows()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 06 |  |
| DATE: |  |

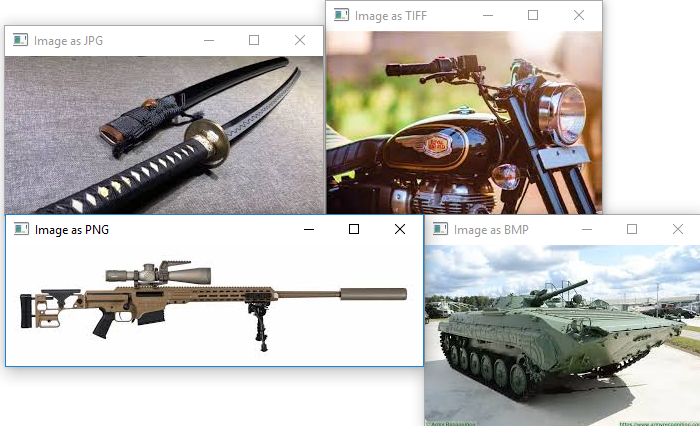
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
def read\_and\_display\_images():  
 image\_paths = [  
 'image.jpg',  
 'tank.bmp',  
 'sniper.png',  
 'bullet.tiff'  
 ]  
 for image\_path in image\_paths:  
 image = cv2.imread(image\_path)  
 if image is None:  
 print(f"Error: Image '{image\_path}' not found!")  
 continue  
 format\_name = image\_path.split('.')[-1].upper()  
 cv2.imshow(f"Image as {format\_name}", image)  
 cv2.waitKey(0)  
 cv2.destroyAllWindows()  
read\_and\_display\_images()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 07 |  |
| DATE: |  |

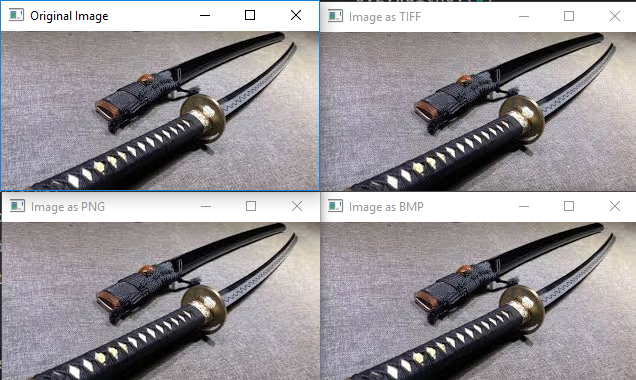
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
def read\_and\_display\_image(image\_path):  
 image = cv2.imread(image\_path)  
 if image is None:  
 print("Error: Image not found!")  
 return  
 cv2.imshow("Original Image", image)  
 cv2.imwrite('output\_image.png', image)  
 cv2.imwrite('output\_image.bmp', image)  
 cv2.imwrite('output\_image.tiff', image)  
 img\_png = cv2.imread('output\_image.png')  
 cv2.imshow("Image as PNG", img\_png)  
 img\_bmp = cv2.imread('output\_image.bmp')  
 cv2.imshow("Image as BMP", img\_bmp)  
 img\_tiff = cv2.imread('output\_image.tiff')  
 cv2.imshow("Image as TIFF", img\_tiff)  
 cv2.waitKey(0)  
 cv2.destroyAllWindows()  
image\_path = 'image.jpg'  
read\_and\_display\_image(image\_path)

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 08 |  |
| DATE: |  |

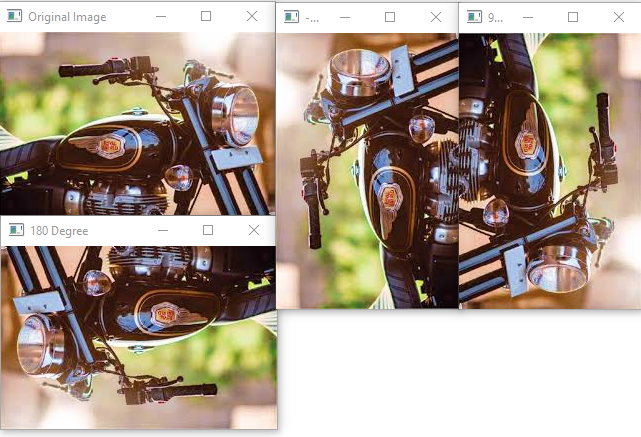
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
image = cv2.imread('bullet.tiff')  
rotated\_image = cv2.rotate(image, cv2.ROTATE\_90\_CLOCKWISE)  
rotated\_image\_1 = cv2.rotate(image, cv2.ROTATE\_90\_COUNTERCLOCKWISE)  
rotated\_image\_2 = cv2.rotate(image, cv2.ROTATE\_180)  
cv2.imshow("Original Image", image)  
cv2.imshow("90 Degree", rotated\_image)  
cv2.imshow("-90 Degree",rotated\_image\_1)  
cv2.imshow("180 Degree",rotated\_image\_2)  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
from PIL import Image  
image\_path = "csk.jpg"  
img = Image.open(image\_path)  
angle = 45  
rotated\_img = img.rotate(angle, expand=True)  
rotated\_img.show()  
rotated\_img.save(f'rotated\_{angle}\_degrees.jpg')

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 09 |  |
| DATE: |  |

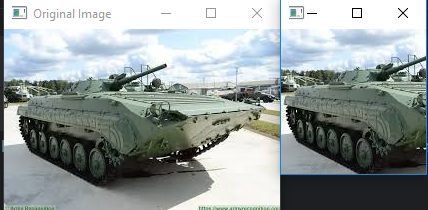
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
image = cv2.imread('tank.bmp')  
y\_start, y\_end = 10, 155  
x\_start, x\_end = 10, 155  
cropped\_image = image[y\_start:y\_end, x\_start:x\_end]  
cv2.imshow('Original Image', image)  
cv2.imshow('Cropped Image', cropped\_image)  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
cv2.imwrite('cropped\_image.jpg', cropped\_image)

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 10 |  |
| DATE: |  |

AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
image = cv2.imread('image.jpg')  
flipped\_horizontally = cv2.flip(image, 1)

flipped\_vertically = cv2.flip(image, 0)

flipped\_both = cv2.flip(image, -1)

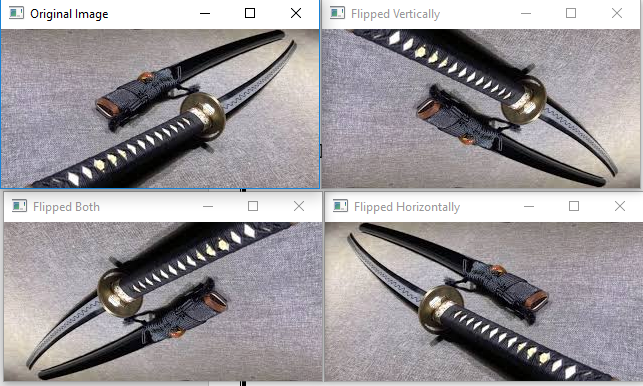
cv2.imshow('Original Image', image)

cv2.imshow('Flipped Horizontally', flipped\_horizontally)

cv2.imshow('Flipped Vertically', flipped\_vertically)

cv2.imshow('Flipped Both', flipped\_both)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 11 |  |
| DATE: |  |

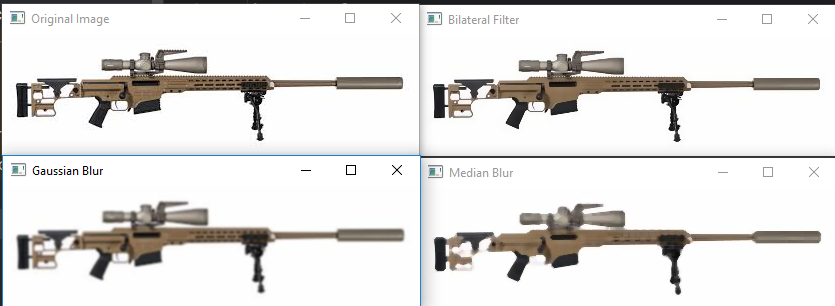
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
image = cv2.imread('sniper.png')  
cv2.imshow('Original Image', image)  
gaussian\_blur = cv2.GaussianBlur(image, (5, 5), 0)  
cv2.imshow('Gaussian Blur', gaussian\_blur)  
median\_blur = cv2.medianBlur(image, 5) # 5x5 kernel  
cv2.imshow('Median Blur', median\_blur)  
bilateral\_blur = cv2.bilateralFilter(image, 9, 75, 75)   
cv2.imshow('Bilateral Filter', bilateral\_blur)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 12 |  |
| DATE: |  |

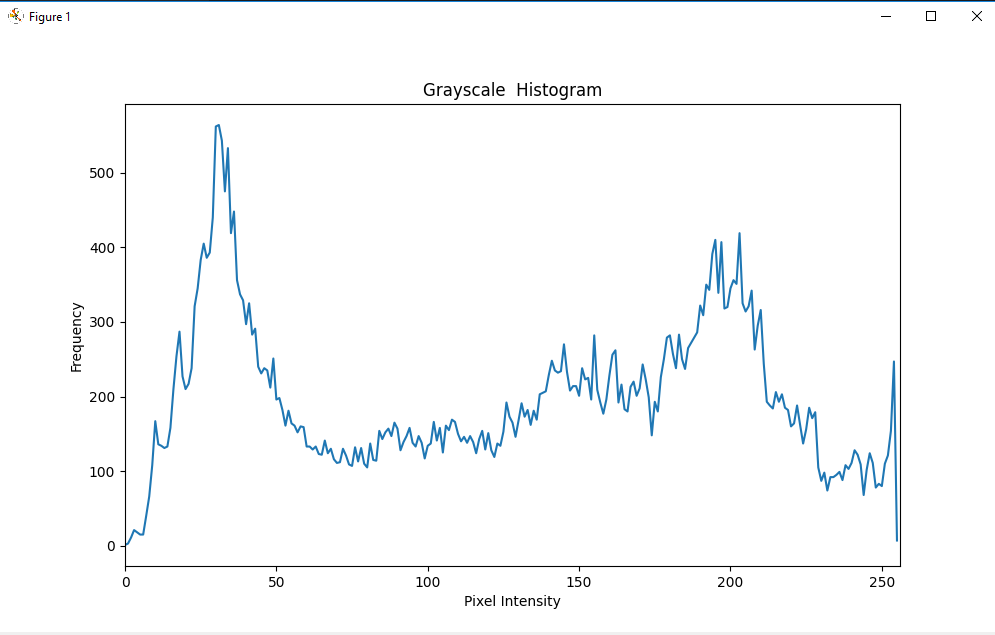
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
import matplotlib.pyplot as plt  
image = cv2.imread('bullet.tiff')  
image\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
hist = cv2.calcHist([image\_gray], [0], None, [256], [0, 256])  
plt.figure(figsize=(10, 6))  
plt.title('Grayscale Histogram')  
plt.xlabel('Pixel Intensity')  
plt.ylabel('Frequency')  
plt.plot(hist)  
plt.xlim([0, 256])  
plt.show()

OUTPUT:



RESULT: