|  |  |
| --- | --- |
| EXP.NO: 01 |  |
| DATE: |  |

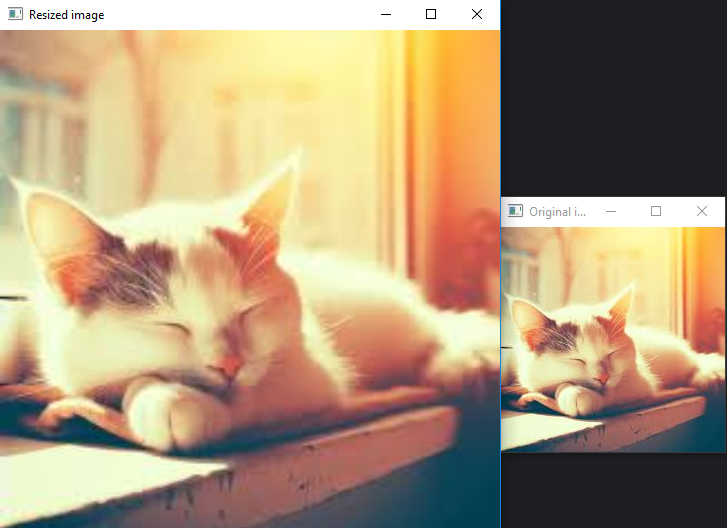
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
  
  
def main():  
 image = cv2.imread('C:\\Users\\CSE\\Desktop\\cat.jpg')  
 if image is None:  
 print("ERROR: could not find image")  
 return  
 res\_image = cv2.resize(image, (500, 500))  
 cv2.imwrite("resized\_image.jpg", res\_image)  
 print("Resized image is 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: |  |

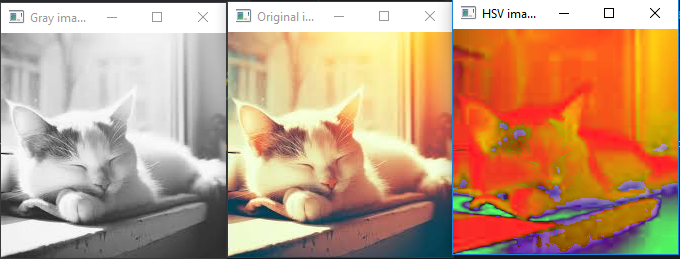
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
  
  
def main():  
 image = cv2.imread('C:\\Users\\CSE\\Desktop\\cat.jpg')  
 if image is None:  
 print("ERROR: could not find image")  
 return  
 gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
 hsv\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)  
 cv2.imshow("Original 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: 03 |  |
| DATE: |  |

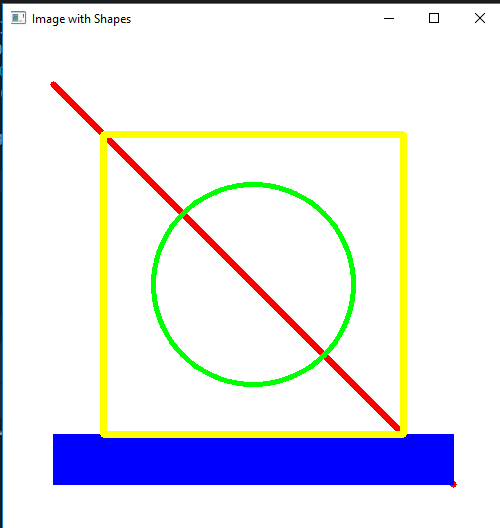
AIM:

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, 400), (0, 255, 255), 5)  
cv2.imshow("Image with Shapes", image)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

OUTPUT:



RESULT:

|  |  |
| --- | --- |
| EXP.NO: 04 |  |
| DATE: |  |

AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
from PIL import Image  
  
# Load an image  
image = cv2.imread('C:\\Users\\CSE\\Desktop\\cat.jpg')  
  
# Print image properties  
print("Image Height:", image.shape[0]) # Height  
print("Image Width:", image.shape[1]) # Width  
print("Number of Channels:", image.shape[2]) # Channels (e.g., 3 for BGR, 1 for grayscale)  
print("Image Size (in Pixels):", image.size) # Total number of pixels  
print("Image Data Type:", image.dtype) # Data type of image (e.g., uint8)  
# Open an image using Pillow  
image = Image.open('C:\\Users\\CSE\\Desktop\\cat.jpg')  
  
# Get the format of the image (e.g., 'JPEG', 'PNG', etc.)  
print("Image Format:", image.format)

OUTPUT:

Image Height: 159

Image Width: 318

Number of Channels: 3

Image Size (in Pixels): 151686

Image Data Type: uint8

Image Format: JPEG

RESULT:

|  |  |
| --- | --- |
| EXP.NO: 05 |  |
| DATE: |  |

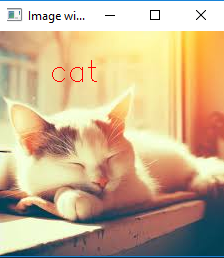
AIM:

ALGORITHM:

SOURCE CODE:

import cv2  
image = cv2.imread('cat.jpg')  
res\_img = cv2.resize(image,(400,350))  
text = 'cat'  
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(0)  
cv2.destroyAllWindows()

OUTPUT:



RESULT:

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

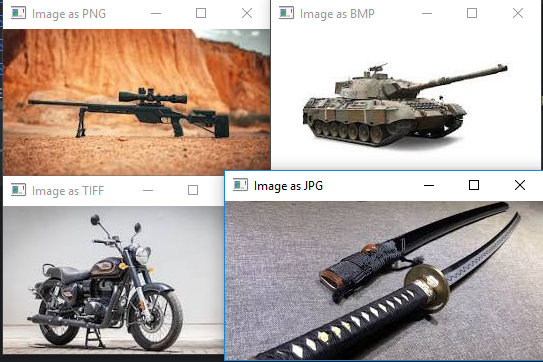
AIM:

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: |  |

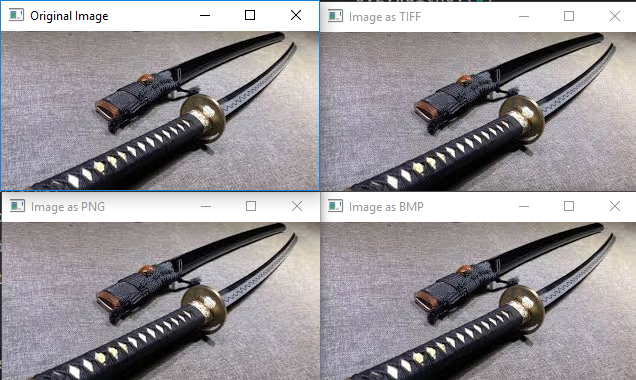
AIM:

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: |  |

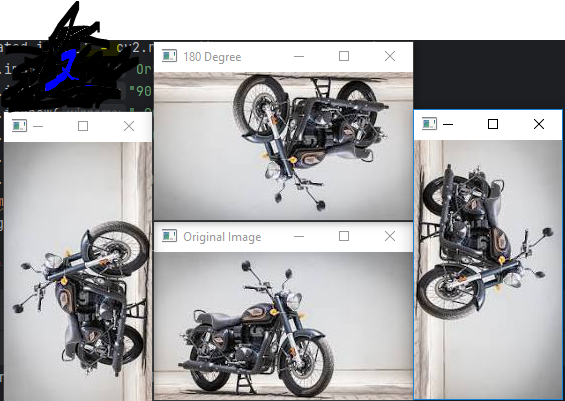
AIM:

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: |  |

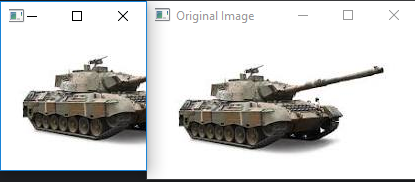
AIM:

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)

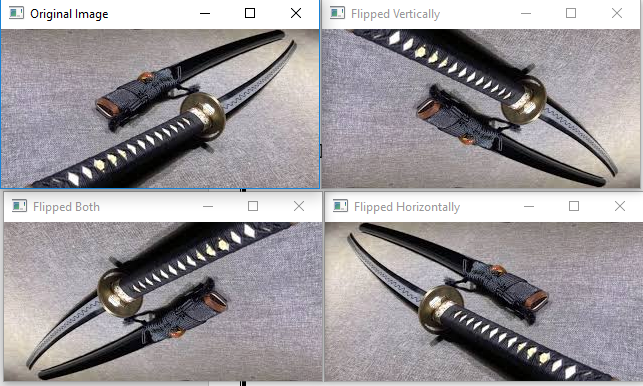
cv2.imshow('Original Image', image)

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: |  |

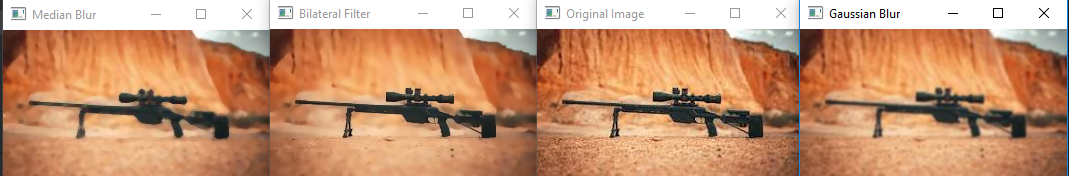
AIM:

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: