EXPT NO: 1	**BASIC IMAGE PROCESSING OPERATIONS**
DATE: 08/07/2025	

AIM:

To Implement various basic image processing operations like Reading image, writing image and conversion of images.

ALGORITHM:

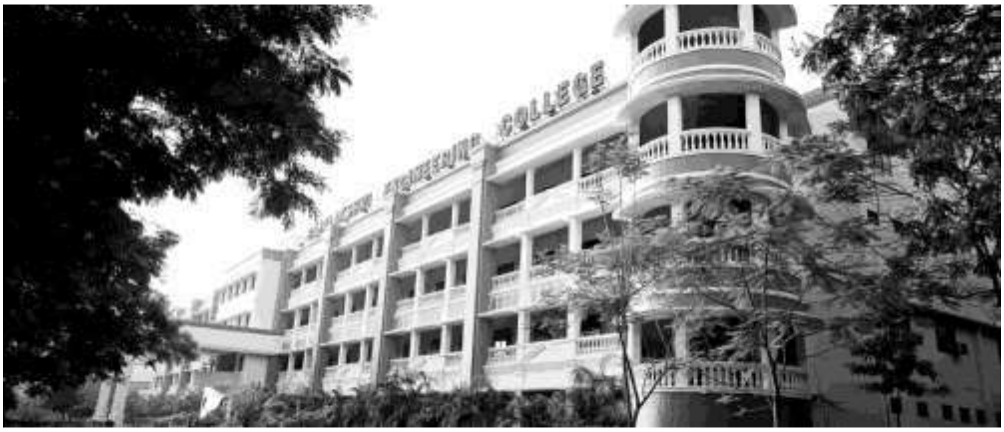
1. Import required libraries **(OpenCV, NumPy).**
2. Read the input image using **cv2.imread().**
3. Display the image using **cv2.imshow().**
4. Convert colour spaces (RGB ↔ Grayscale ↔ HSV) using **cv2.cvtColor().**
5. Write the output image using **cv2.imwrite().**
6. Close windows using **cv2.destroyAllWindows().**

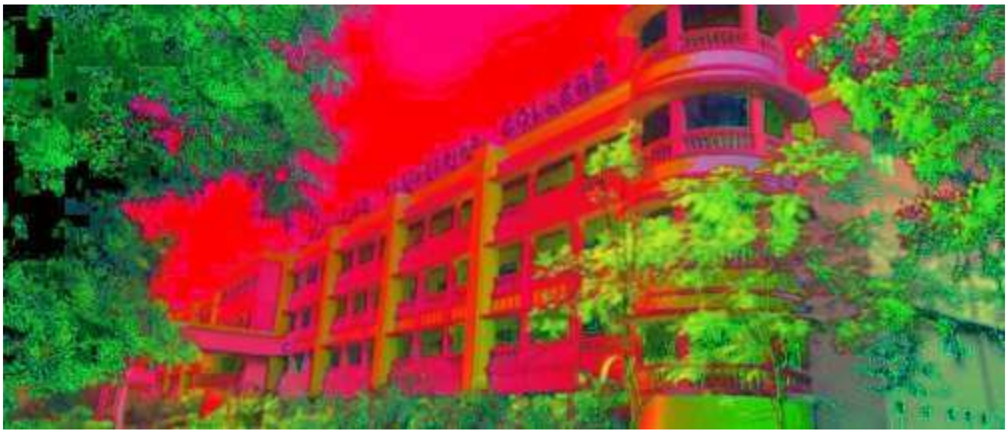
CODE:

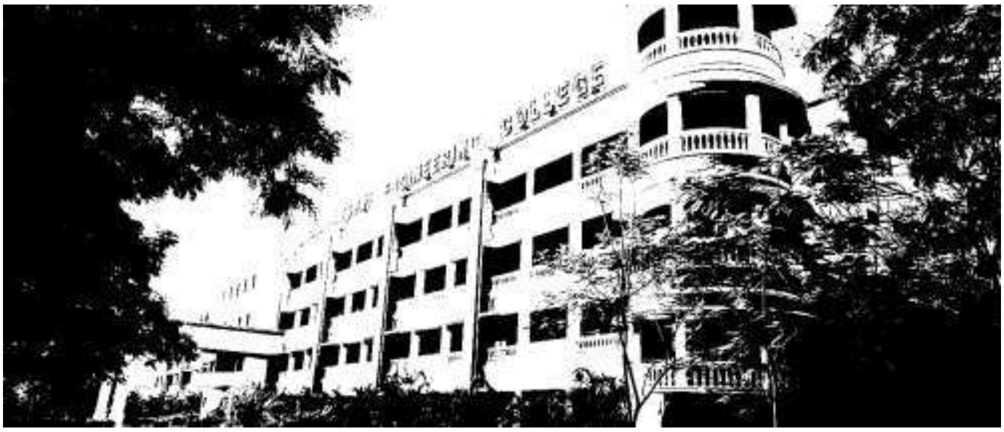
```
#PROGRAM 1


import cv2
from google.colab.patches import cv2_imshow

# ---- 1. Reading an Image ----
# Replace 'example.jpg' with your actual image path
image = cv2.imread('/content/rec.jpg')

# Check if the image was successfully loaded
if image is None:
    print("Error: Image not found.")
    exit()

# Display the original image
cv2_imshow(image)
```

```
# cv2.waitKey(0) # waitKey and destroyAllWindows are not needed with
cv2_imshow
# cv2.destroyAllWindows()


# ---- 2. Writing (Saving) the Image ----
cv2.imwrite('/content/rec.jpg', image)
print("Image saved as output.jpg")

# ---- 3a. Convert to Grayscale ----
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2_imshow(gray_image)
# cv2.waitKey(0)
# cv2.destroyAllWindows()

# ---- 3b. Convert to HSV ----
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
cv2_imshow(hsv_image)
# cv2.waitKey(0)
# cv2.destroyAllWindows()


# ---- 3c. Convert to Binary (Thresholding) ----
# Threshold value = 127, Max value = 255
_, binary_image = cv2.threshold(gray_image, 127, 255, cv2.THRESH_BINARY)
cv2_imshow(binary_image)
# cv2.waitKey(0)
# cv2.destroyAllWindows()
```

OUTPUT:







RESULT:

Thus, the various basic image processing operations like Reading image, writing image and conversion of images were implemented successfully.