Date : 13/08/2025

FCV WEEK4 SUBMISSION

Name: Davasam Karthikeya   Section: AIMLB   Reg No: 230962326

1. Write a program to create binary images using thresholding methods.

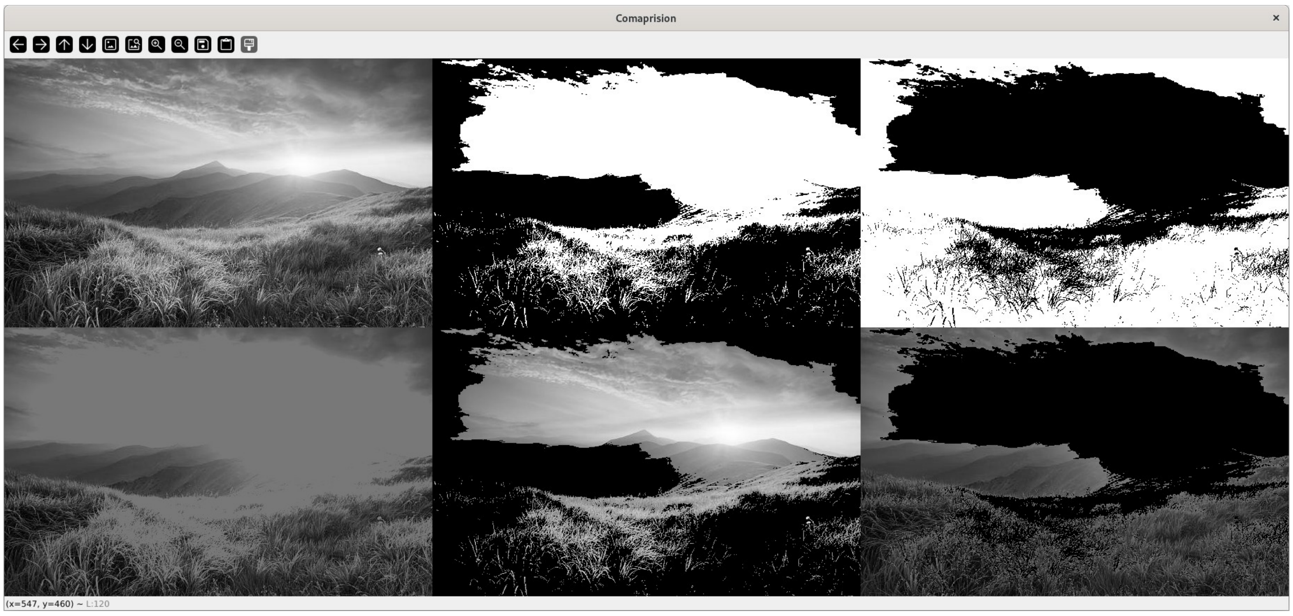
Code:-

```
import cv2 as cv
import numpy as np

class Thresholding:
    def BINARY_THRESHOLD(img: np.ndarray, thresh: int, maxval: int):
        height, width = img.shape
        out = np.zeros_like(img)
        for i in range(height):
            for j in range(width):
                if img[i, j] > thresh:
                    out[i, j] = maxval
                else:
                    out[i, j] = 0
        return out

    def BINARY_THRESHOLD_INV(img: np.ndarray, thresh: int, maxval: int):
        height, width = img.shape
        out = np.zeros_like(img)
        for i in range(height):
            for j in range(width):
                if img[i, j] < thresh:
                    out[i, j] = maxval
                else:
                    out[i, j] = 0
        return out

    def TRUNCATE(img: np.ndarray, thresh: int, maxval: int):
        return np.minimum(img, thresh)

    def TOZERO(img: np.ndarray, thresh: int, maxval: int):
        height, width = img.shape
        out = img.copy()
        for i in range(height):
            for j in range(width):
                if img[i, j] < thresh:
                    out[i, j] = 0
        return out

    def TOZERO_INV(img: np.ndarray, thresh: int, maxval: int):
        height, width = img.shape
        out = img.copy()
        for i in range(height):
            for j in range(width):
                if img[i, j] > thresh:
                    out[i, j] = 0
        return out


img = cv.imread('Week1/img.jpg')
img_gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

Date : 13/08/2025

```
thresh1 = Thresholding.BINARY_THRESHOLD(img_gray, 120, 255)
thresh2 = Thresholding.BINARY_THRESHOLD_INV(img_gray, 120, 255)
thresh3 = Thresholding.TRUNCATE(img_gray, 120, 255)
thresh4 = Thresholding.TOZERO(img_gray, 120, 255)
thresh5 = Thresholding.TOZERO_INV(img_gray, 120, 255)

cv.imshow('Comaprision', np.vstack([np.hstack([img_gray, thresh1, thresh2]),
np.hstack([thresh3, thresh4, thresh5])]))
cv.waitKey(0)
cv.destroyAllWindows()
```

Output:-



2. Write a program to detect lines using Hough transform.

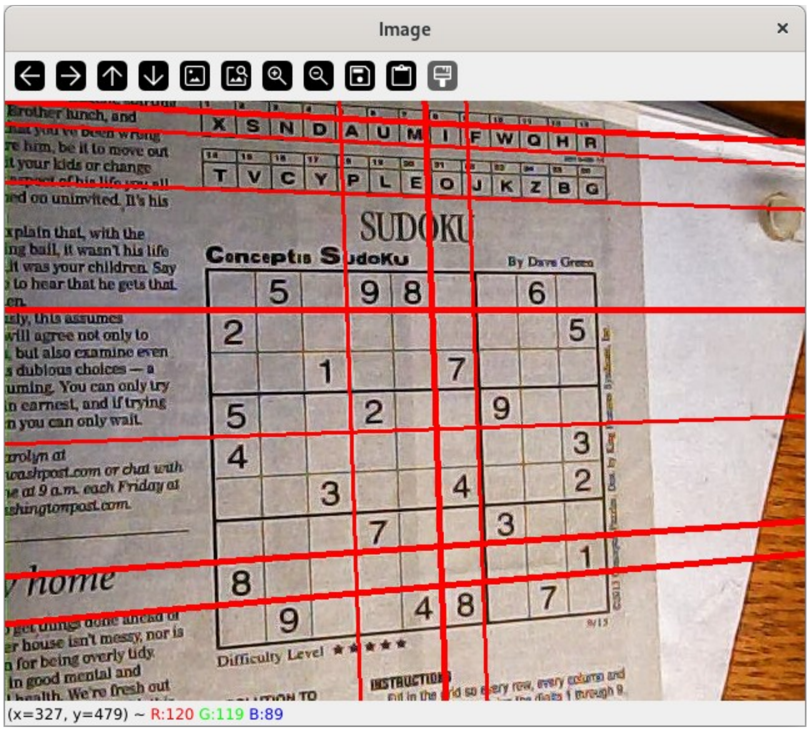
Code:-

```
import cv2 as cv
import numpy as np


def hough_lines(edges: cv.Mat, rho, min_theta, max_theta, theta, threshold):
    diag_len = int(np.ceil(np.sqrt(edges.shape[0]**2 + edges.shape[1]**2)))

    theta_angles = np.arange(min_theta, max_theta, theta)
    rho_values = np.arange(-diag_len, diag_len +1, rho)

    theta_count = len(theta_angles)
    rho_count = len(rho_values)

    accumulator = np.zeros((rho_count, theta_count))

    sins = np.sin(theta_angles)
    coss = np.cos(theta_angles)

    xs, ys = np.where(edges > 0)

    for x, y in zip(xs, ys):
```

Date : 13/08/2025

```
        for angle_idx in range(theta_count):
            cur_rho = x*coss[angle_idx] + y*sins[angle_idx]
            rho_pos = np.where(rho_values < cur_rho)[0][-1]

            accumulator[rho_pos][angle_idx] += 1
    accumulator /= np.max(accumulator)
    rho_index, theta_index = np.where(accumulator > threshold)

    return np.vstack([rho_values[rho_index], theta_angles[theta_index]]).T


img = cv.imread('Week4/Sudoku.jpg')
diag_len = int(np.ceil(np.sqrt(img.shape[0]**2 + img.shape[1]**2)))
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

edges = cv.Canny(gray, 50, 150, apertureSize=3)
lines = hough_lines(edges, 1, 0, np.pi, np.pi/180, 0.75)

for r_theta in lines:
    arr = np.array(r_theta, dtype=np.float64)
    r, theta = arr
    a = np.sin(theta)
    b = np.cos(theta)
    x0 = a*r
    y0 = b*r

    x1 = int(x0 + diag_len*(-b))
    y1 = int(y0 + diag_len*(a))
    x2 = int(x0 - diag_len*(-b))
    y2 = int(y0 - diag_len*(a))

    cv.line(img, (x1, y1), (x2, y2), (0, 0, 255), 2)

cv.imshow('Image', img)
cv.waitKey(0)
cv.destroyAllWindows()
```

Output:-



Date : 13/08/2025

3. Write a program to segment an image based on colour.

Code:-

```
import cv2 as cv
import numpy as np

img = cv.imread('Week4/Lanes.jpg')
height, width, _ = img.shape

lw_h, lw_s, lw_v = np.array([0, 0, 200])
uw_h, uw_s, uw_v = np.array([179, 50, 255])

hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
mask_img = np.zeros_like(img)

for i in range(height):
    for j in range(width):
        h,s,v = hsv[i][j]
        if lw_h < h < uw_h and lw_s < s < uw_s and lw_v < v < uw_v:
            mask_img[i][j] = (255, 255, 255)


cv.imshow('Image', np.hstack([img, mask_img]))
cv.waitKey(0)
cv.destroyAllWindows()
```

Output:-

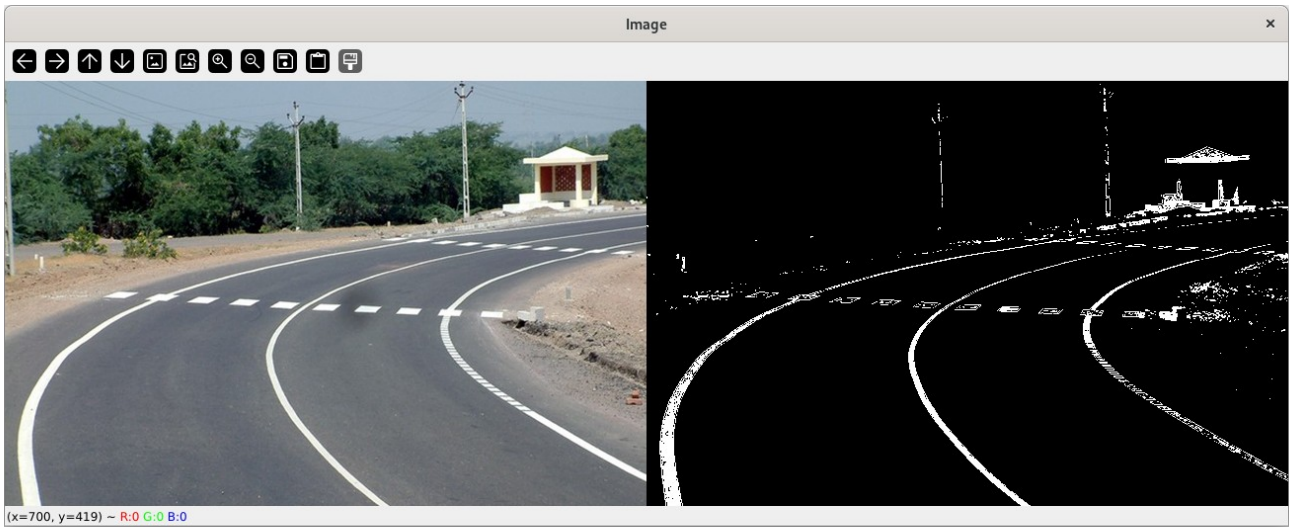


Image
(x=700, y=419) ~ R:0 G:0 B:0