15.

Perform basic Image Handling and processing operations on the image is to read an image in python and detect the corners in the image using Harris Corner Detection function

**AIM:**

To perform **Harris Corner Detection** on a given image using Python and OpenCV.

**PROCEDURE:**

1. Install OpenCV if not already installed using:

* pip install opencv-python

1. Import the required libraries (cv2 and numpy).
2. Read the input image using cv2.imread().
3. Convert the image to grayscale using cv2.cvtColor().
4. Apply the **Harris Corner Detection** function using cv2.cornerHarris().
5. Dilate the detected corners to mark them clearly.
6. Overlay the detected corners on the original image.
7. Display both the original and corner-detected images using cv2.imshow().
8. Save the corner-detected image using cv2.imwrite(), if needed.
9. Wait for a key press and close all image windows using cv2.waitKey(0) and cv2.destroyAllWindows().

**PROGRAM:**

import cv2

import numpy as np

# Read the input image

image = cv2.imread("image.jpg") # Replace with your image file path

# Convert the image to grayscale

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Convert to float32 for Harris Corner Detection

gray = np.float32(gray)

# Apply Harris Corner Detection

harris\_corners = cv2.cornerHarris(gray, blockSize=2, ksize=3, k=0.04)

# Dilate corner points for better visibility

harris\_corners = cv2.dilate(harris\_corners, None)

# Mark detected corners in red

image[harris\_corners > 0.01 \* harris\_corners.max()] = [0, 0, 255]

# Display images

cv2.imshow("Original Image", image)

cv2.imshow("Harris Corner Detection", image)

# Save the result

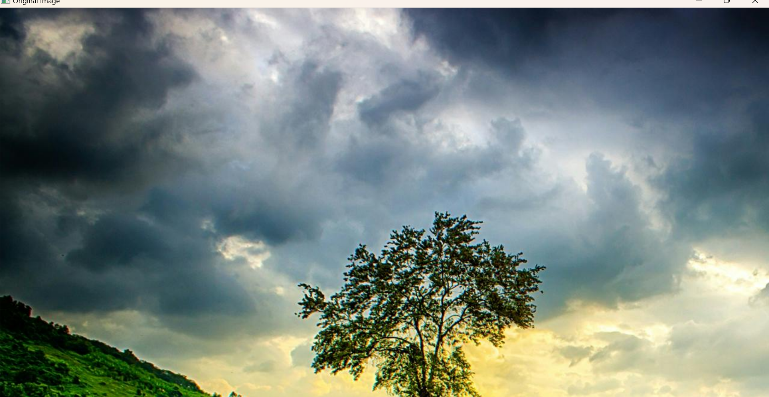
cv2.imwrite("harris\_corners.jpg", image)

# Wait for a key press and close all windows

cv2.waitKey(0)

cv2.destroyAllWindows()

**INPUT:**



**OUTPUT:**



**RESULT :**

The program successfully applies **Harris Corner Detection** to the given image, detects the corners, displays the result, and saves it as "harris\_corners.jpg".