**--Lab-5- To perform Detect corner of an image using OpenCV--**

To **detect corners in an image using OpenCV**, one of the most common methods is **Harris Corner Detection** or **Shi-Tomasi Corner Detection**.

Below is a **Python example using OpenCV** for both methods.

**✅ 1. Harris Corner Detection (OpenCV)**

import cv2

import numpy as np

# Load image and convert to grayscale

img = cv2.imread('your\_image.jpg')

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

# Convert to float32

gray = np.float32(gray)

# Harris corner detection

dst = cv2.cornerHarris(gray, blockSize=2, ksize=3, k=0.04)

# Dilate to mark the corners

dst = cv2.dilate(dst, None)

# Threshold for an optimal value

img[dst > 0.01 \* dst.max()] = [0, 0, 255] # Mark corners in red

cv2.imshow('Harris Corners', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

**✅ 2. Shi-Tomasi Corner Detection (cv2.goodFeaturesToTrack)**

import cv2

import numpy as np

# Load image and convert to grayscale

img = cv2.imread('your\_image.jpg')

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

# Detect corners using Shi-Tomasi method

corners = cv2.goodFeaturesToTrack(gray, maxCorners=100, qualityLevel=0.01, minDistance=10)

corners = np.int0(corners)

# Draw corners

for i in corners:

x, y = i.ravel()

cv2.circle(img, (x, y), 5, (0, 255, 0), -1)

cv2.imshow('Shi-Tomasi Corners', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

**📝 Notes:**

* blockSize: Neighborhood size considered for corner detection.
* ksize: Aperture parameter for the Sobel operator.
* k: Harris detector free parameter in the equation.
* maxCorners: Maximum number of corners to return.
* qualityLevel: Minimum accepted quality of image corners.