**Lab-4 To perform Circle Detection using OpenCV | Python**

**Circle Detection using OpenCV in Python** is commonly done using the **Hough Circle Transform**, a feature extraction technique for detecting circles in images.

**🔹 What is Circle Detection?**

**Circle Detection** is a process in computer vision to find circles in digital images. It is especially useful in robotics, image analysis, and medical imaging.

**🔹 Purpose**

* To locate circular shapes in an image.
* Helps in object recognition, shape analysis, and tracking.

**🔹 Applications**

* Coin recognition systems
* Eye (iris/pupil) detection
* Ball tracking in sports
* Traffic sign recognition
* Industrial automation

**🔹 OpenCV Function Used**

cv2.HoughCircles()

**✅ Example Code: Circle Detection using OpenCV**

import cv2

import numpy as np

# Load image

image = cv2.imread('circles.png') # Replace with your image path

output = image.copy()

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

# Apply Gaussian Blur to reduce noise

gray\_blurred = cv2.GaussianBlur(gray, (9, 9), 2)

# Detect circles using HoughCircles

circles = cv2.HoughCircles(gray\_blurred,

cv2.HOUGH\_GRADIENT,

dp=1.2,

minDist=20,

param1=50,

param2=30,

minRadius=0,

maxRadius=0)

# If some circles are detected, draw them

if circles is not None:

circles = np.uint16(np.around(circles))

for i in circles[0, :]:

# Draw the outer circle

cv2.circle(output, (i[0], i[1]), i[2], (0, 255, 0), 2)

# Draw the center of the circle

cv2.circle(output, (i[0], i[1]), 2, (0, 0, 255), 3)

# Show results

cv2.imshow('Detected Circles', output)

cv2.waitKey(0)

cv2.destroyAllWindows()

**🔹 Parameters Explained**

| **Parameter** | **Meaning** |
| --- | --- |
| dp | Inverse ratio of accumulator resolution to image resolution. |
| minDist | Minimum distance between detected centers. |
| param1 | Higher threshold for Canny edge detector. |
| param2 | Accumulator threshold for circle detection. |
| minRadius | Minimum radius of circles to detect. |
| maxRadius | Maximum radius of circles to detect. |