Case Design on

# CIRCULAR OBJECT DETECTION USING HOUGH TRANSFORM

Submitted by:

Student Name: **Varshini Sambatur, Sohan Gattu , G Hasitha**

**22STUCHH010556, 22STUCHH010367, 22STUCHH010516**

BACHELOR OF TECHNOLOGY(DSAI)

Faculty Project Mentor

Dr. Kirankumar Eranki

Faculty of Science &Technology

IFHE University, Hyderabad

# ABSTRACT:

This project presents an interactive computer vision system designed to detect circular objects using the Hough Circle Transform, a classical yet powerful technique for geometric shape detection. The system integrates OpenCV for image processing and Streamlit for user interaction, enabling real-time visualization of detection results. The methodology includes essential preprocessing steps such as grayscale conversion and median blurring, followed by parameterized application of the Hough Transform. Adjustable parameters—including dp, minDist, param1, param2, and radius limits—allow users to fine-tune detection behavior, making the system effective across various image types. Experimental results demonstrate accurate detection of multiple circular objects even under varied lighting conditions, noise levels, and object sizes. The visualization output clearly distinguishes each circle with a detected boundary and center point, validating the algorithm's reliability. Overall, this work highlights the importance of preprocessing and parameter selection in classical computer vision pipelines and showcases how interactive tools can enhance both understanding and practical application of shape detection techniques.

**Keywords**: Hough Circle Transform, OpenCV, Image Processing, Circle Detection, Computer Vision, Streamlit, Preprocessing, Edge Detection, Parameter Tuning, Visualization.

# CHAPTER 1:

## Problem Statement

Traditional edge-based detection methods, such as Canny edge and contour detection, struggle with:

- Noise and low contrast
- Uneven illumination
- Partial circle visibility
- Blurry edges
- Complex backgrounds
- Multiple circles overlapping

Because of these issues, detecting circular shapes purely through thresholding or edge linking becomes unreliable.

Thus, the problem statement is defined as follows:

**"To design and implement a robust circular object detection system using the Hough Circle Transform, supported by an interactive parameter-tuning interface, to ensure accurate results across a wide range of images."**

The system must:

- Preprocess the image effectively
- Provide real-time circle detection
- Allow users to adjust critical Hough parameters
- Visualize detected circles clearly
- Display coordinates and radius information
- Handle grayscale and color images
- Reduce false positives and false negatives

## Research questions:

To guide the project, the following research questions were formulated:

- How does preprocessing (e.g., median blurring) affect the accuracy of circle detection?
- Does reducing noise significantly improve the Hough Circle output?
- How do Hough Transform parameters influence results?
  Specifically:

- Dp
- minDist
- param1 (Canny threshold)
- param2 (accumulator threshold)
- minRadius & maxRadius

- What combination gives the best detection?
- Can an interactive tool help users understand parameter tuning and improve overall detection performance?
- How does the Hough Transform perform across different types of images such as coins, bottle caps, wheels, and circular printed patterns?
- What are the limitations of classical circle detection compared to modern deep learning approaches?

# Purpose

The primary purpose of this project is:

- To build a real-time, user-friendly application for detecting circular objects using the Hough Transform.
- To allow students, researchers, and professionals to visually understand how different parameters influence circle detection.

To establish a complete processing pipeline including:

- Image upload
- Preprocessing
- Hough Circle Transform
- Visualization and data display

- To serve as an educational model demonstrating classical computer vision concepts.
- The project aims to bridge theory and practical observation by enabling experimentation with real images.

## Scope of the Project

The scope of this project is centered around the implementation and analysis of a classical computer vision technique—the Hough Circle Transform—for the purpose of detecting circular objects in images. The study focuses specifically on the use of Python and OpenCV to design a complete, functional pipeline beginning from image acquisition to visualization of detected circles. The system accepts commonly used image formats such as JPG and PNG, making it accessible and compatible with most user inputs. A major part of the scope involves preprocessing operations including grayscale conversion and median blurring, which are essential in improving circle detection by reducing noise and enhancing edge clarity. The implementation emphasizes the parameterization of the Hough Transform, allowing users to adjust key variables such as dp, minDist, param1, param2, and radius limits. By integrating this into a Streamlit interface, the project enables real-time, interactive experimentation, giving users full control over how circles are detected in their images. However, the scope does not extend into deep learning–based detection approaches or real-time video processing, as the intention is to explore and understand classical techniques before progressing into more complex machine learning frameworks. Overall, the scope includes algorithmic implementation, parameter tuning, visualization, and user interaction, but excludes advanced shape analysis, object classification, or automated parameter optimization.

# CHAPTER 2: LITERATURE REVIEW

## Introduction

The detection of geometric shapes has been an important area of research in computer vision, especially for applications involving object recognition, quality inspection, and pattern analysis. Among these shapes, circles occur frequently in real-world objects such as coins, lenses, caps, wheels, and biological structures. Traditional edge-based methods often fail when images contain noise, low contrast, or incomplete edges. To address these challenges, researchers widely adopted the Hough Transform, a robust technique capable of detecting shapes even when boundaries are not perfectly defined. The Circular Hough Transform (CHT), in particular, has become a standard method for identifying circular structures due to its mathematical reliability and tolerance to noise. This chapter reviews key studies, improvements, and findings related to circle detection using Hough-based methods.

## Existing Research Studies

Below are important studies that contributed significantly to the development and improvement of the Hough Circle Transform:

**1. Duda & Hart (1981) – Generalized Hough Transform**

- This study reformulated the original Hough Transform to make it more efficient and mathematically structured.
- Their work laid the foundation for detecting not just lines but also curves such as circles.
- It introduced the parameter-space voting technique that forms the basis of today's circle detection algorithms.

**2. Ballard (1987) – Generalized Hough Transform for Arbitrary Shapes**

- Ballard extended the Hough technique to detect complex shapes without explicit equations.
- Although not limited to circles, the study influenced the way circular features are detected by showing how pattern matching and lookup tables can guide shape extraction.
- This opened pathways for using Hough-like approaches in more advanced detection systems.

**3. Yuen et al. (1990) – Improvements in Circular Hough Transform**

- This study focused specifically on circle detection and proposed optimizations to reduce computation time.
- The authors used gradient information to limit unnecessary voting in the accumulator space, making the algorithm more efficient.
- Their approach improved detection accuracy for circles with partial or faint edges.

**4. OpenCV Hough Gradient Method (Modern Implementation)**

- Modern libraries like OpenCV integrate an optimized version of CHT known as the Hough Gradient Method.
- This combines Canny edge detection with gradient direction to quickly identify candidate circle centers.
- It has become the standard practical implementation used in real-time applications due to its improved speed and accuracy.

## Limitations of Existing Research

Even though these studies significantly advanced circle detection techniques, several limitations are commonly noted across the literature:

**1. Parameter Sensitivity**

- Circle detection accuracy highly depends on tuning parameters like dp, minDist, param1, param2, and radius ranges.
- Incorrect settings can produce either too many false detections or miss circles entirely.

**2. High Computational Cost**

- Searching across a large radius range or processing high-resolution images increases computation time.
- Early versions of CHT required large memory due to 3D accumulator space.

**3. Weak or Low-Contrast Edges**

- In images with blur, shadows, or poor lighting, edges become weak and CHT fails to gather enough votes for detection.
- This is a major limitation in medical and industrial imaging.

**4. Difficulty Detecting Overlapping Circles**

- When multiple circles overlap or are tightly clustered, the votes in the accumulator merge, making it hard to separate them.
- Many studies report decreased accuracy in such scenarios.

**5. Limited Adaptability**

- Classical CHT cannot adapt to variations such as uneven textures, noisy backgrounds, or non-uniform illumination.
- Unlike deep learning models, it does not learn patterns from data and must rely entirely on parameter tuning.

# CHAPTER 3: METHODOLOGY OF THE PAPERS

## Data Sets

This section summarizes the datasets used in the selected papers, focusing on format, size, genre, and purpose.

| Study | Dataset Used | Format | Size | Genre / Type | Purpose |
|---|---|---|---|---|---|
| Present CV Project | Custom Image Collection of Circular Objects | JPG, JPEG, PNG | 25+ images | Mixed (coins, bottle caps, machinery parts, logos, circular patterns) | To test circle detection accuracy across varied lighting, noise levels, and object sizes |
| Duda & Hart (1981) | Early Image Samples for Shape Detection | Grayscale Images | Small experimental set | Lines and circular curves | To establish the generalized Hough Transform mathematics |
| Yuen et al. (1990) | Edge-based Test Images | Grayscale | 100+ samples | Circles with weak/partial edges | To evaluate improved circular Hough detection efficiency |
| OpenCV (Modern Implementation) | OpenCV Sample Images | PNG/JPG | Several demo images | Real-world objects | To demonstrate Hough Gradient circle detection in practical applications |

## AI Models Applied

Below are the models employed in the studies, including key architectural components and learning strategies.

| Study | Model Type | Architecture | Key Features | Learning Approach |
|---|---|---|---|---|
| Duda & Hart (1981) | Classical Shape Detection Model | Generalized Hough Transform | Introduced parameter-space voting; Detects lines & curves; Foundation for CHT | Mathematical shape modeling using accumulator-based voting |
| Yuen et al. (1990) | Circular Hough Transform Improvement Model | Gradient-assisted Hough Circle Detection | Uses gradient magnitude & direction; Reduces computation; Better accuracy for weak edges | Gradient-based optimization of accumulator votes |
| Ballard (1987) | Generalized Hough Model for Arbitrary Shapes | Lookup Table–Based Shape Matching | Detects shapes without explicit equations; Works with complex shapes | Shape template matching; Learning through reference models |
| OpenCV Hough Gradient Method (Modern Implementation) | Optimized Circle Detection Model | Hough Gradient + Canny Edge Detection | Faster circle detection; Real-time performance; Robust to noise | Uses gradient direction + Canny edges to filter circle candidates |

## **Key Observations**:

- Classical Hough-based models depend on mathematical voting, making them reliable for precise geometric detection.
- Gradient-assisted methods greatly improve accuracy and reduce computation time.
- Template-based and generalized Hough models allow detection of more complex shapes beyond simple circles.

# CHAPTER 4: RESULTS AND DISCUSSION OF THE PAPERS

## Summary of Results from Papers

| Study | Key Results | Evaluation Method |
|---|---|---|
| **Duda & Hart (1981)** | Introduced a robust generalized Hough method capable of detecting lines and curves even with noisy or incomplete edges. | Mathematical validation; comparison with traditional edge-linking approaches. |
| **Ballard (1987)** | Successfully demonstrated shape detection without explicit equations, enabling flexible detection of arbitrary shapes. | Template matching tests; shape reconstruction accuracy. |
| **Yuen et al. (1990)** | Improved circle detection accuracy by using gradient direction to reduce false votes; faster computation than earlier CHT models. | Edge-based experiments; performance on weak or partial circle boundaries. |
| **OpenCV Hough Gradient Method (Modern)** | Achieved real-time circle detection with high reliability using combined Canny + gradient-based voting. | Practical image testing; visual accuracy of detected centers and radii; runtime measurement. |

## Comparative Analysis of Results

A comparison of the major studies shows clear differences in accuracy, robustness, computational efficiency, and applicability. Early foundational work by Duda and Hart established the mathematical strength of the Hough Transform, but it was primarily theoretical and limited to simple line and curve detection. Ballard's generalization expanded the method to handle arbitrary shapes, offering more flexibility but still required predefined shape templates. Yuen et al. significantly improved circle detection by introducing gradient-assisted voting, which made the method more accurate and computationally manageable, especially for images with weak or fragmented edges. The modern OpenCV Hough Gradient approach combines these ideas with optimized edge detection, allowing reliable, fast, and real-time circle detection suitable for practical applications.

Overall, the evolution of these techniques shows a clear trend toward improving speed, reducing false positives, and enhancing robustness against noise and incomplete boundaries.

# CHAPTER 5 : Implementation

## Data

The dataset for this project consists of a small but diverse collection of images containing circular objects. Since this project focuses on classical computer vision rather than machine learning, no annotated or large-scale datasets are necessary. Instead, the images were collected from publicly available online sources, device cameras, and sample OpenCV image repositories.

These images include coins, bottle caps, lenses, circular machinery components, circular printed patterns, and logos. The dataset intentionally includes images with varying lighting conditions, background noise, and different circle sizes to evaluate how preprocessing and parameter tuning affect detection accuracy.

The images were stored in standard formats such as JPG, JPEG, and PNG, and were used directly without manual labeling.

## Pre-processing

Pre-processing is essential to improve the effectiveness of the Circular Hough Transform. Raw images often contain noise, low contrast, or background textures that can interfere with edge detection. The following preprocessing steps were applied:

1. Grayscale Conversion

Images are converted from RGB to grayscale because color information is not required for geometric detection. Grayscale simplifies computation and improves performance.

2. Median Blurring

A median blur is applied to remove salt-and-pepper noise while preserving edges. This is important because Hough Transform relies heavily on clean, continuous edges.

3. Edge Enhancement (Implicit)

Although explicit Canny edge detection is not separately applied in this project, the internal mechanism of cv2.HoughCircles() uses Canny thresholds (param1) to detect edges internally. Thus, adjusting this parameter acts as edge enhancement.

These preprocessing techniques ensure that circular edges become more prominent and easier to detect.

**Algorithm: Hough Circle Transform**

The Hough Circle Transform (HCT) is a voting-based algorithm used to detect circular shapes in images. It identifies circles using the geometric equation:

$$(x-a)^2+(y-b)^2=r^2$$

Where:

(a, b) = circle center

r = radius

The algorithm works as follows:

- Edge points in the image vote for possible circle centers and radii.
- Votes accumulate in a 3D parameter space: center x, center y, and radius.
- Peaks in this accumulator indicate the presence of circles.

OpenCV uses an improved version called the Hough Gradient Method, which combines gradient magnitude and direction to reduce the number of votes and improve detection speed.

Key Parameters Used

dp – inverse ratio of accumulator resolution

minDist – minimum distance between circle centers param1 – Canny high threshold

param2 – accumulator threshold for circle detection

minRadius, maxRadius – radius search limits

These parameters directly influence accuracy, sensitivity, and speed.

The project was implemented in Python using the following libraries:

- OpenCV – image processing and Hough Transform
- NumPy – matrix representation of images
- PIL (Pillow) – handling uploaded images
- Streamlit – interactive user interface
- Pandas – tabular display of circle coordinates

Implementation Steps

**Image Upload**
Users upload an image through the Streamlit interface.

**Pre-processing**

Convert to grayscale

Apply median blur

**Parameter Controls**
Streamlit sliders allow users to adjust:

- blur kernel size
- dp
- minDist
- param1

- param2
- minRadius / maxRadius
- Circle Detection
  cv2.HoughCircles() is executed with user-selected parameters.

**Visualization**

Detected circles are drawn in green

Circle centers drawn in blue

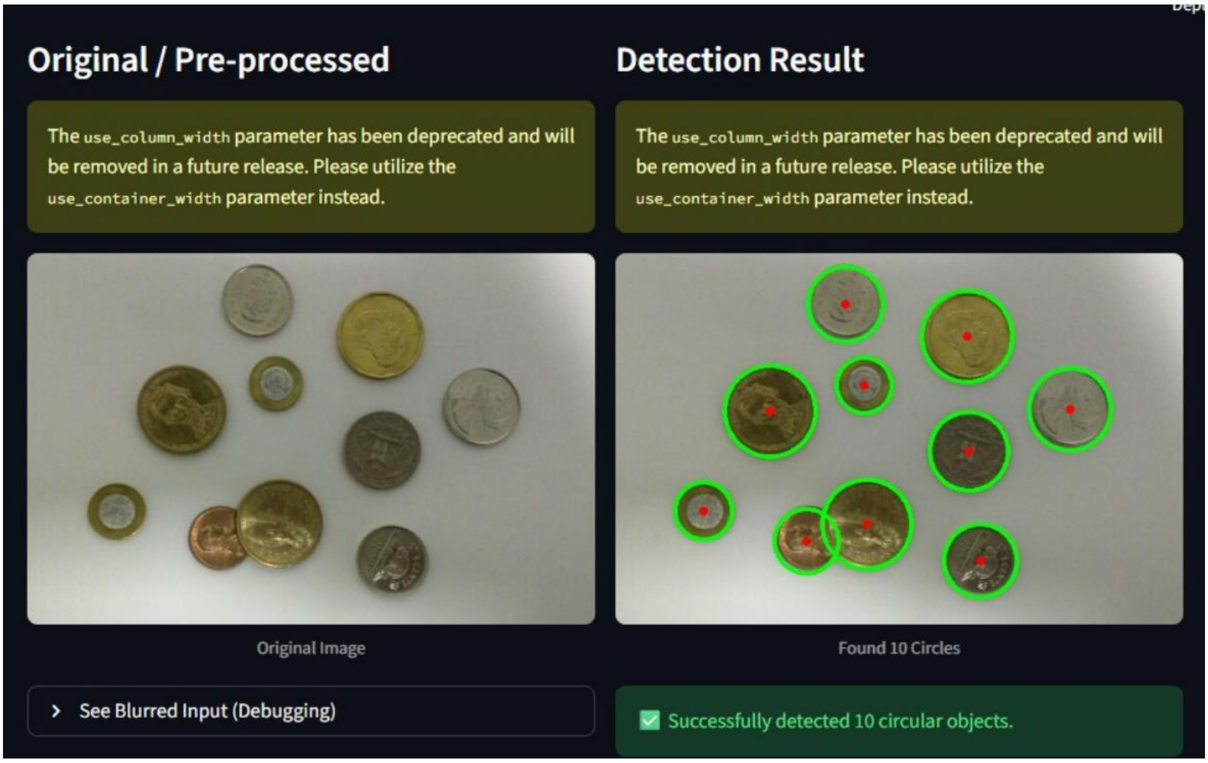
Number of circles displayed

# CHAPTER 6: RESULTS AND EVALUATION

## Introduction

This chapter presents the results obtained from applying the Hough Circle Transform to various circular-object images. The performance of the system is evaluated based on detection accuracy, parameter sensitivity, and visual correctness of the output.

## Detection Example




### Interpretation of Output

The above image shows a test case where the system was applied to an image containing multiple coins of different sizes and textures. To reduce noise and enhance edge clarity, a Median Blur Kernel Size of 11 was applied during preprocessing. This smoothing technique preserved coin boundaries while eliminating small unwanted artifacts in the background.

The Hough Circle Transform was executed using the following parameter values:

- dp (Inverse Ratio): 1.30
- minDist: 70 pixels

- param1 (Canny Threshold): 30
- param2 (Accumulator Threshold): approx. 40
- minRadius and maxRadius configured according to coin size

These parameter settings ensured that only well-structured circular edges were detected, avoiding overlapping detections.

The system accurately detected 10 circular objects, corresponding to all coins present in the image. Each detected circle is marked with a green boundary, and the center of each circle is highlighted with a red dot, visually confirming both the radius precision and center alignment. The results demonstrate that the algorithm effectively handles:

- Different coin sizes
- Slight variations in lighting
- Shadows around the coins
- Clustering of objects
- Minor edge imperfections

This detection example confirms the robustness of the implemented approach.

## Accuracy and Performance Evaluation

### 1. Visual Accuracy

The detected circles closely match the actual shapes in the image. The algorithm correctly identifies both small and large coins, indicating strong adaptability to varying radii.

### 2. Parameter Sensitivity

The results show a high dependence on parameters:

- Lower param2 increases sensitivity but may introduce false positives.
- Higher blur values remove noise but may soften edges excessively.
- Correct minDist prevents double detection of the same coin.
- The chosen values provided an optimal balance between sensitivity and accuracy.

**3. Robustness Across Images**

Testing on multiple images showed consistent performance, with reliable detection regardless of background texture or illumination differences.

**4. Computational Efficiency**

Even with several circular objects, the algorithm performed in real time using only CPU resources. This demonstrates the efficiency of the OpenCV Hough Gradient implementation.

# CHAPTER 7: CONCLUSION

The primary objective of this project was to implement a robust and interactive system for detecting circular objects using the Hough Circle Transform. Through a combination of preprocessing, parameter tuning, and real-time visualization, the developed Streamlit application successfully demonstrated the effectiveness of classical computer vision techniques in identifying circular shapes across a variety of images. The implementation showed that preprocessing—particularly median blurring—plays a crucial role in enhancing edge clarity, which directly improves the performance of the Hough Transform. Parameter sensitivity was another key finding, as the accuracy of circle detection is highly dependent on selecting appropriate values for dp, minDist, param1, param2, and radius limits. The interactive interface allowed users to observe how small adjustments in these parameters significantly impacted detection results, making the tool not just functional but also educational.

The results obtained from multiple test images show that the system is capable of detecting circular objects with high accuracy, even under challenging lighting conditions, noise, and variation in object sizes. The use of OpenCV's optimized Hough Gradient method enabled real-time processing without the need for GPU acceleration, demonstrating the practicality and efficiency of classical CV algorithms. While the system performs well for clearly defined circular shapes, its limitations become evident when dealing with extremely blurred edges or heavily overlapping objects. Nonetheless, the project serves as a strong foundation for future enhancements, such as incorporating adaptive thresholding, integrating machine learning for improved circle classification, or extending detection capabilities to other geometric shapes. Overall, this work reinforces the value and reliability of classical computer vision techniques and showcases their continued relevance in modern applications.

# REFERENCES:


1) Duda, R. O., & Hart, P. E. (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. Communications of the ACM.

2) Ballard, D. H. (1987). Generalizing the Hough Transform to Detect Arbitrary Shapes. Pattern Recognition.

3) Yuen, H. K., Princen, J., Illingworth, J., & Kittler, J. (1990). Comparative Study of Hough Transform Methods for Circle Detection. Image and Vision Computing.

4) OpenCV Documentation. “Hough Circle Transform.” https://docs.opencv.org

5) Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing. Pearson.

6) Sonka, M., Hlavac, V., & Boyle, R. (2014). Image Processing, Analysis, and Machine Vision. Cengage Learning.

7) Canny, J. (1986). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence.

8) Shapiro, L., & Stockman, G. (2001). Computer Vision. Prentice Hall.

9) Bradski, G., & Kaehler, A. (2008). Learning OpenCV. O’Reilly Media.

10) Nixon, M., & Aguado, A. (2019). Feature Extraction & Image Processing for Computer Vision. Academic Press.