

# Shape Detection Lab - Project Documentation

## 1. Overview

Shape Detection Lab is a browser-based web application that identifies and classifies geometric shapes such as circles, triangles, rectangles, pentagons, and stars using pure JavaScript and the HTML5 Canvas API. The app runs entirely client-side and uses image segmentation, boundary analysis, and geometric feature extraction. It is deployed as a Trusted Web Activity (TWA) to work like a native Android app with a secure Digital Access Link.

## 2. Workflow Summary

Step 1: Image Acquisition - Upload or select a built-in test image rendered on the HTML canvas.

Step 2: Grayscale Conversion - Convert RGB pixels to grayscale to simplify analysis.

Step 3: Thresholding - Convert grayscale to binary image for segmentation.

Step 4: Connected Component Analysis - Identify separate shape regions using DFS labeling.

Step 5: Boundary Extraction - Extract and simplify shape edges using Ramer-Douglas-Peucker (RDP).

Step 6: Feature Extraction - Compute perimeter, area, circularity, solidity, vertices, aspect ratio, and concavity.

## 3. Shape Classification

Shapes are identified using rule-based thresholds on extracted features:

- Circle: High circularity ( $>0.9$ ), equal width and height, high solidity.
- Triangle: 3 vertices.
- Square: 4 vertices, equal sides, 90 degree angles.
- Rectangle: 4 vertices, unequal sides but right angles.
- Pentagon: 5 to 7 vertices, moderate circularity (0.68 to 0.9).
- Star: 8 or more vertices, concave angles ratio  $\geq 0.15$ .
- Polygon: Default fallback.

## 4. Visualization

Each detected shape is displayed with a bounding box, colored outline, and confidence score label

on the canvas. Results are also displayed in JSON format showing shape type, coordinates, and confidence.

## 5. Architecture Overview

Frontend: HTML5 Canvas + JavaScript (ES6) + CSS (dark theme, responsive design).

Core Algorithm: Image binarization -> Connected components -> Boundary detection -> RDP simplification -> Feature computation -> Rule-based classification.

## 6. TWA Integration (Trusted Web Activity)

The app is deployed as a Progressive Web App (PWA) and installable via TWA on Android.

- TWA allows web apps to run in fullscreen mode inside Chrome without browser UI.
- Digital Asset Link (assetlinks.json) verifies the app and website ownership.
- Users can install via Add to Home Screen or Play Store.

Example DAL: <https://<your-domain>/.well-known/assetlinks.json>

## 7. Advantages

- 100 percent JavaScript (no libraries)
- Runs offline
- Cross-platform (desktop and Android)
- Interactive visualization
- Extendable detection logic

## 8. Limitations and Future Work

Limitations: Sensitive to lighting, overlapping shapes, low contrast.

Future Work: Adaptive thresholding, ML-based classification, improved star detection, contour-based refinement.

## 9. Conclusion

Shape Detection Lab proves that computer vision tasks like shape detection can run efficiently in browsers using only native APIs. Combined with TWA deployment, it achieves offline functionality and native experience through secure Digital Access Links.