

GROUP PROJECT

OUR TEAM



Arushi Mishra
Batch-2026



Deepti Kushawaha
Batch-2026

PROJECT GOALS

Given Different Shapes in form of hand drawn curves(doodles),first task is to detect different Geometric shapes present in it

Regularising the Curves

In that Hand drawn curves we have to detect different line of symmetries

Symmetry Detection

Completing the Incomplete Curves based on regularity,symmetry etc.

Completing the shapes

TASK-SHAPE DETECTION

- **STEP 1**

Converts an SVG file to a PNG image with transparent background, replaces transparency with white, and performs image preprocessing (Gaussian blur, adaptive thresholding, and morphological operations) to clean up and enhance the contours.

- **STEP 2**

Utilizes edge detection and contour finding to detect shapes in the processed image. Classifies shapes as lines, triangles, rectangles, squares, pentagons, hexagons, circles, ellipses, and stars based on contour characteristics and Hough Circle detection and Visualization of output that we got.

TASK-SYMMETRY DETECTION

- **STEP 1**

Converts an SVG file to a PNG image like we did in Task-1. Checks for vertical, horizontal, and diagonal lines of symmetry in the detected contours. It uses image transformations and structural similarity to determine if the contour exhibits symmetry along these lines. Draws the detected symmetry lines on the image.

- **STEP 2**

Saves and displays the original and annotated images side by side with symmetry lines drawn. Prints the detected lines of symmetry for each contour

In this Task we use library of python called skimage.metrics structural_similarity model of it.

TASK-COMPLETION OF CURVES

• STEP 1

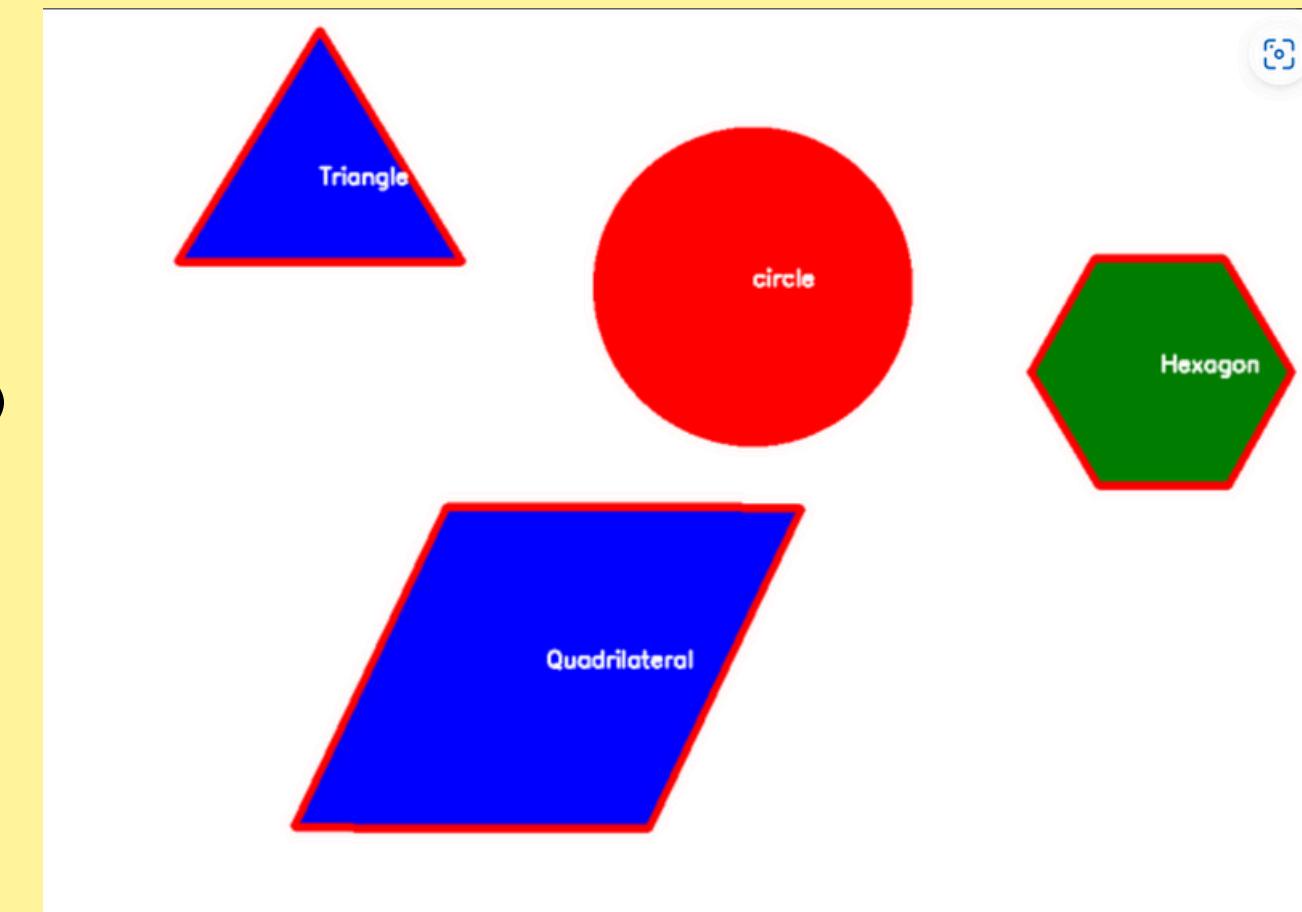
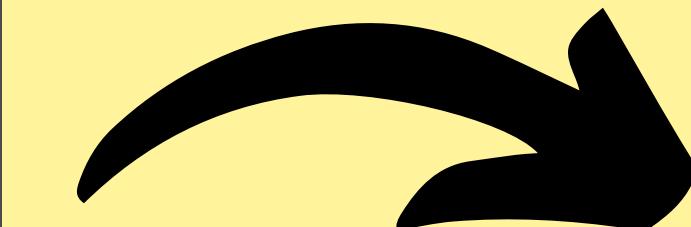
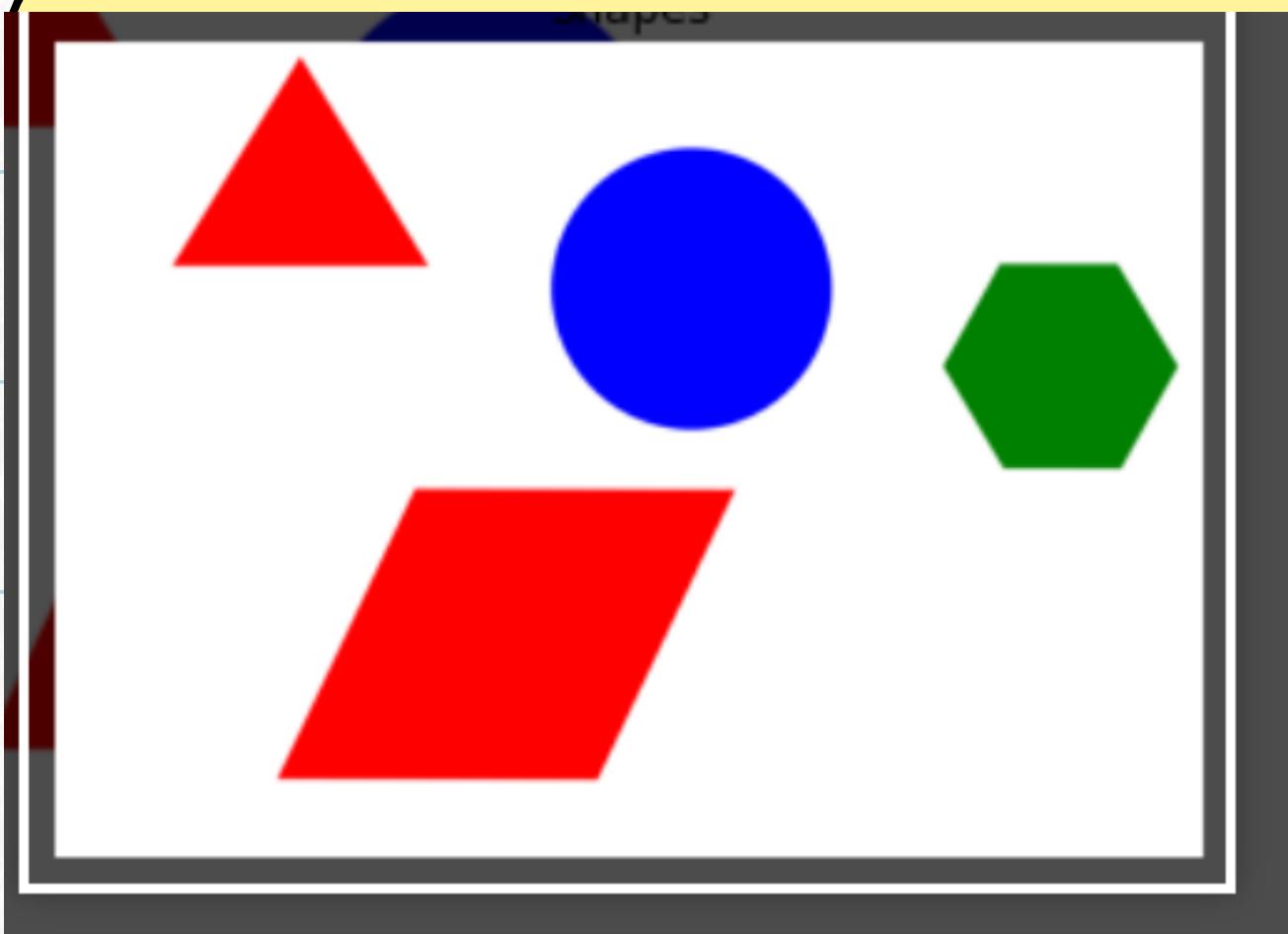
Curve Completion: Extracts coordinates from SVG paths, identifies gaps, and uses spline interpolation to smooth and complete curves.

Visualization: Plots original curves in blue and completed curves in red to visualize the enhancements.

• STEP 2

Libraries: Utilizes `svgpathtools` for SVG parsing, `numpy` and `scipy` for numerical and interpolation tasks, and `matplotlib` for plotting.

VISUALISATION-TASK1

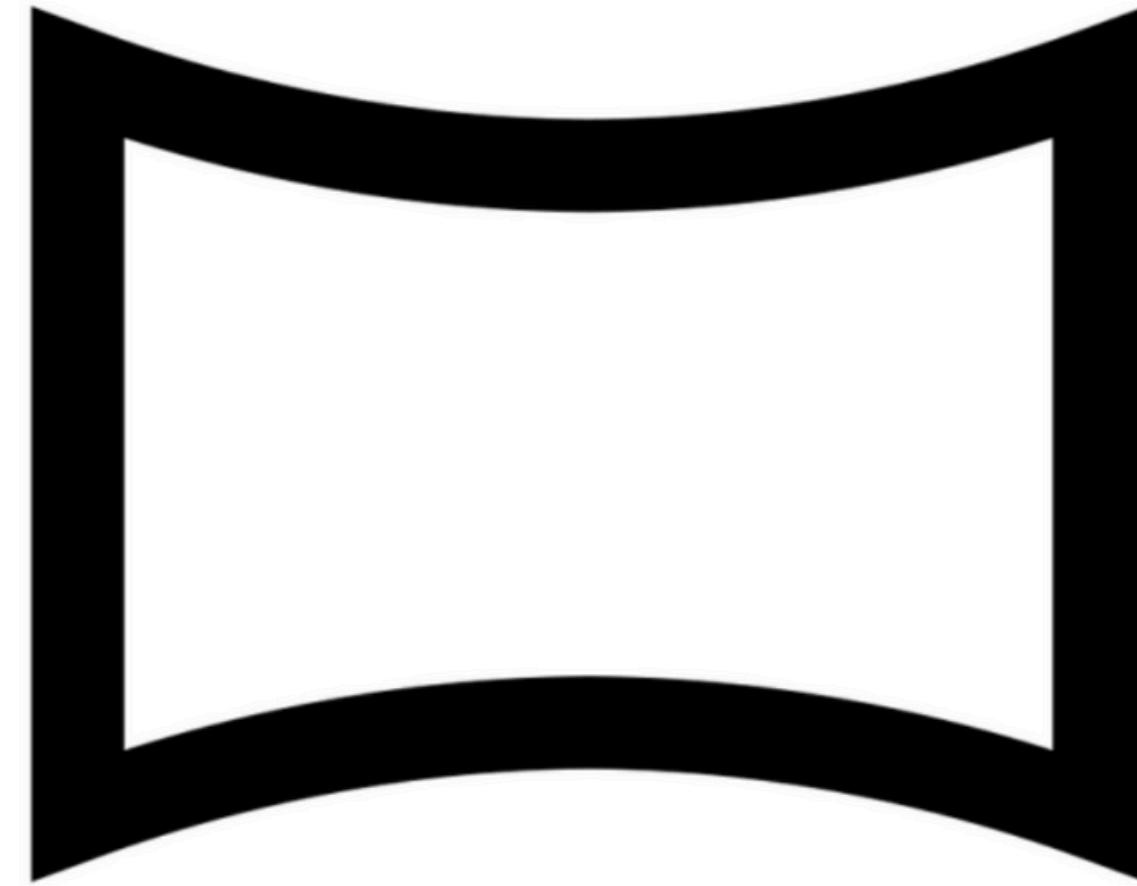


INPUT IMAGES

**NOTICE THE RED BOUNDARIES IN OUTPUT IMAGES THESE ARE CALLED COUNTERS, WHICH HELPS
IN DETECTING THE SHAPES IN THE CURVE**

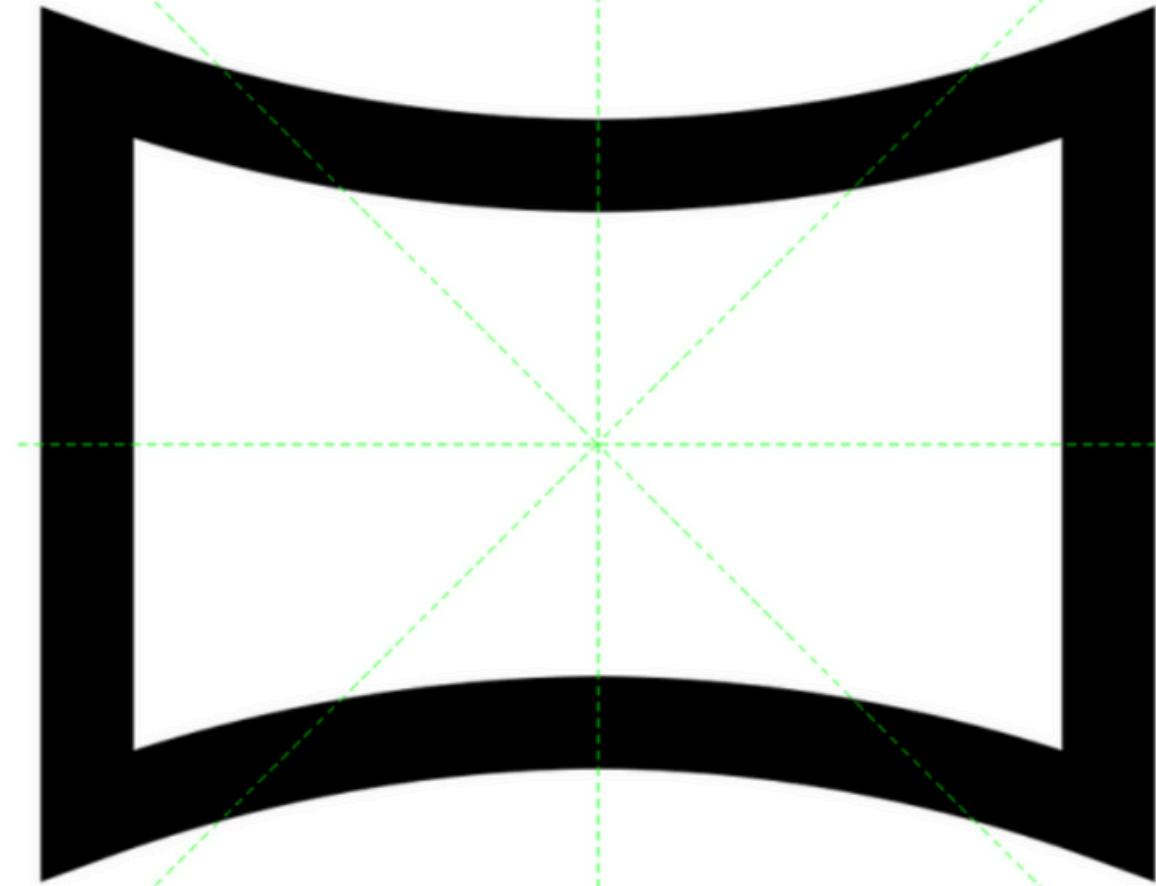
OUTPUT IMAGES

VISUALISATION-TASK2



INPUT IMAGES

Contour at (359, 360): vertical, horizontal, diagonal1, diagonal2



OUTPUT IMAGES

THESE ARE THE RESULTS THAT I HAVE TESTED WITH MY CODE (SCREENSHOT OF JUPYTER NOTEBOOK)

VISUALISATION-TASK3

INPUT IMAGES

THESE ARE THE RESULTS THAT I HAVE TESTED WITH MY CODE (SCREENSHOT OF JUPYTER NOTEBOOK)

OUTPUT IMAGES

ALGORITHMS USED

TASK 1

Library :-CairoSVG,Open Cv,Numpy,Matplotlib,PIL
Algorithm Used:-Gaussian blur,Adaptive Thresholding,Morphological Operations,Contour Approximation (approxPolyDP),Some logic for shapes detection

TASK 2

Library:-All Task1 Library,scikit-image (SSIM)
Algorithm Used: Structural Similarity Index,Affine Transformation

TASK 3

Library:-All Task1 Library,scipy.interp olate (splprep, splev),svgwrite
Algorithm Used: B-spline interpolation ,Gap Identification,Spline and Beizer Curve Smoothing

THANK YOU