The problem statement "Curvetopia: A Journey into the World of Curves" involves processing 2D curves from polylines to cubic Bézier curves, with tasks that include regularizing curves, exploring symmetry, and completing incomplete curves.

Objectives

Input: A set of polylines representing line art.
Output: A set of curves with regularization, symmetry, and completeness.

Tasks Breakdown:
1. Regularize Curves:
   Identify regular shapes like straight lines, circles, ellipses, rectangles, rounded rectangles, regular polygons, and star shapes from a given set of curves.
   The algorithm should handle hand-drawn shapes and distinguish between regular and irregular shapes.
2. Exploring Symmetry in Curves:
   Identify reflection symmetries in closed shapes.
   Fit identical Bézier curves on points that are symmetric.
3. Completing Incomplete Curves:
   Complete curves that have been partially occluded or disconnected.
   Handle different levels of occlusion, including fully contained, partially contained, and disconnected curves.

Edge Cases:
Irregular hand-drawn shapes that don't fit into standard geometric categories.
Curves with minor irregularities that still exhibit overall symmetry.
Complex occlusions where parts of the curve are missing or disconnected.

Approach

This approach works good for shapes which contain straight lines

1. *Bezier Curve Calculation*: The code defines a function to calculate points on a quadratic Bézier curve using three control points (p0, p1, p2). It also has a function to draw this curve on an image by connecting these points with lines.

2. *Symmetry Lines Drawing*: Another function is used to draw symmetry lines for different shapes detected in the image. It calculates the centroid of the shape and draws vertical, horizontal, and diagonal symmetry lines based on the type of shape (like circles, ellipses, squares, or regular polygons).

3. *Image Processing*:
   - *Read and Convert*: The code reads an image and converts it to grayscale.
   - *Thresholding*: It applies binary thresholding to create a black-and-white image where shapes become distinct from the background.

- *Contour Detection*: It detects the contours (outlines) of shapes in the thresholded image.

4. *Contour Analysis*:
   - For each detected contour, the code filters out those that are too small or too large.
   - It approximates the contour to a simpler polygonal shape and then uses Bézier curves to draw these approximated shapes on the image.

5. *Shape Detection*:
   - The code identifies shapes based on the number of vertices in the approximated contour:
     - *2 vertices*: A straight line.
     - *3 vertices*: A triangle.
     - *4 vertices*: Either a square or rectangle, based on the aspect ratio of the bounding box.
     - *5 to 9 vertices*: A regular polygon.
     - *10 vertices*: A star (treated as a regular polygon for symmetry).
   - For shapes that don't fit these categories, it fits an ellipse to the contour to determine if the shape is a circle or an ellipse based on the aspect ratio.

6. *Drawing on Image*:
   - The detected shapes are drawn on the image using Bézier curves or ellipses.
   - Symmetry lines are also drawn based on the shape detected, highlighting potential symmetry.

Finally, the processed image with drawn shapes and symmetry lines is displayed.
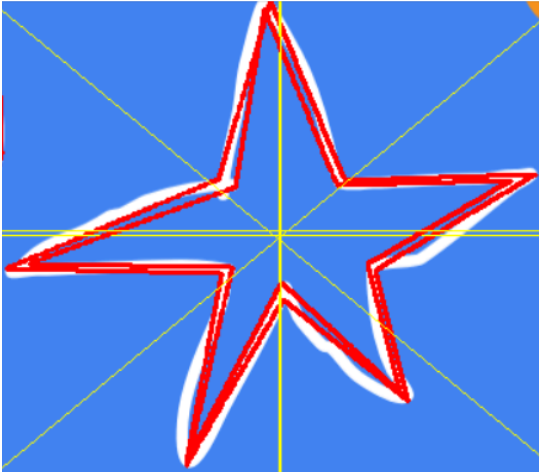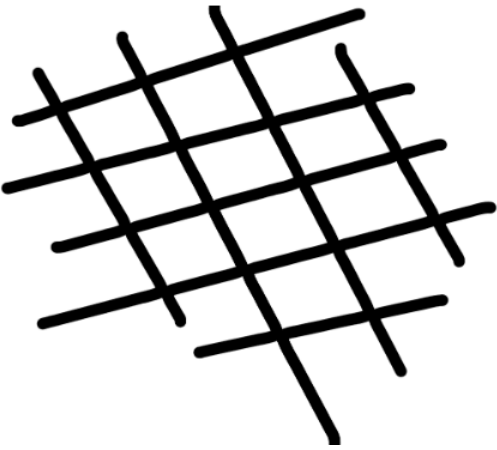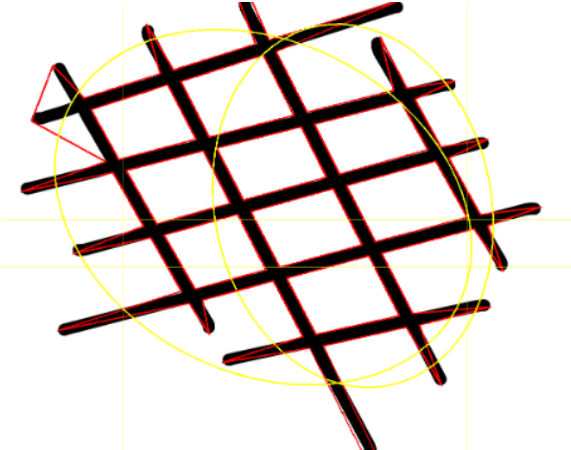

Spline Interpolation

This works better for more rounded and curved shapes
This approach is same as the counter detection approach but instead of using counter detection to regularize the shapes which was not able to work good on more rounded shapes so Here in this approach we are using spline implementation to get smoother rounded curves

Shape Completion Approach:

- Converts SVG image to PNG format for processing.
- Converts the image to grayscale and applies binary thresholding.
- Detects contours in the image.
- Fits ellipses to the contours and identifies outer and inner ellipses.
- Creates masks to identify missing parts of the ellipses.
- Draws missing parts in green (outer) and red (inner) on the original image.
- Completes and draws the full ellipses on the original image.
- Displays the final processed image.

Input	Output
Border Detection	
Spline Interpolation	

Shape Completion