TopoGO: Knot Detection of Images

Yicheng Zhang[1], Zhengyang Cao[2]

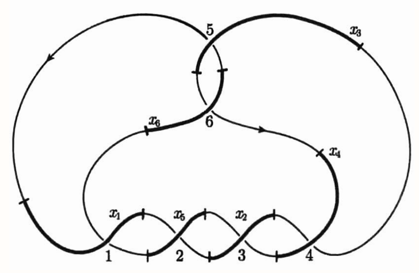
[1, 2]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China. Github: [1]@RUSRUSHB, [2]@drinktoomuchsax

Introduction

Knot theory is an important branch of mathematics that plays a pivotal role in various fields. The Alexander polynomial is used to uniquely describe knots. We automate its extraction using image processing and mathematical modeling, enhancing accuracy and efficiency.

Procedure for Alexander Polynomial

- Construct a planar diagram of the knot.
- Identify and number line segments and crossings.
- Construct the Alexander matrix.
- Calculate the determinant and normalize the polynomial.



(a) Numbered Diagram

$$\begin{pmatrix} 1-t & -1 & 0 & 0 & -1 & 0 \\ 0 & t & 1-t & -1 & 0 & 0 \\ 0 & 0 & 0 & t & 1-t & t \\ 0 & 0 & -1 & 1-t & 0 & -1 \\ -1 & 1-t & t & 0 & 0 & 0 \\ t & 0 & 0 & 0 & t & 1-t \end{pmatrix}$$

(b) Alexander Matrix

Figure 1: How To Get Alexander Matrix

Methodology

The overall process starts with pre-processing the image into a binary form, then segmenting it. We use two approaches: sliding window and thinning. The sliding window leads to pattern recognition, while thinning involves endpoints matching to derive crossing data.

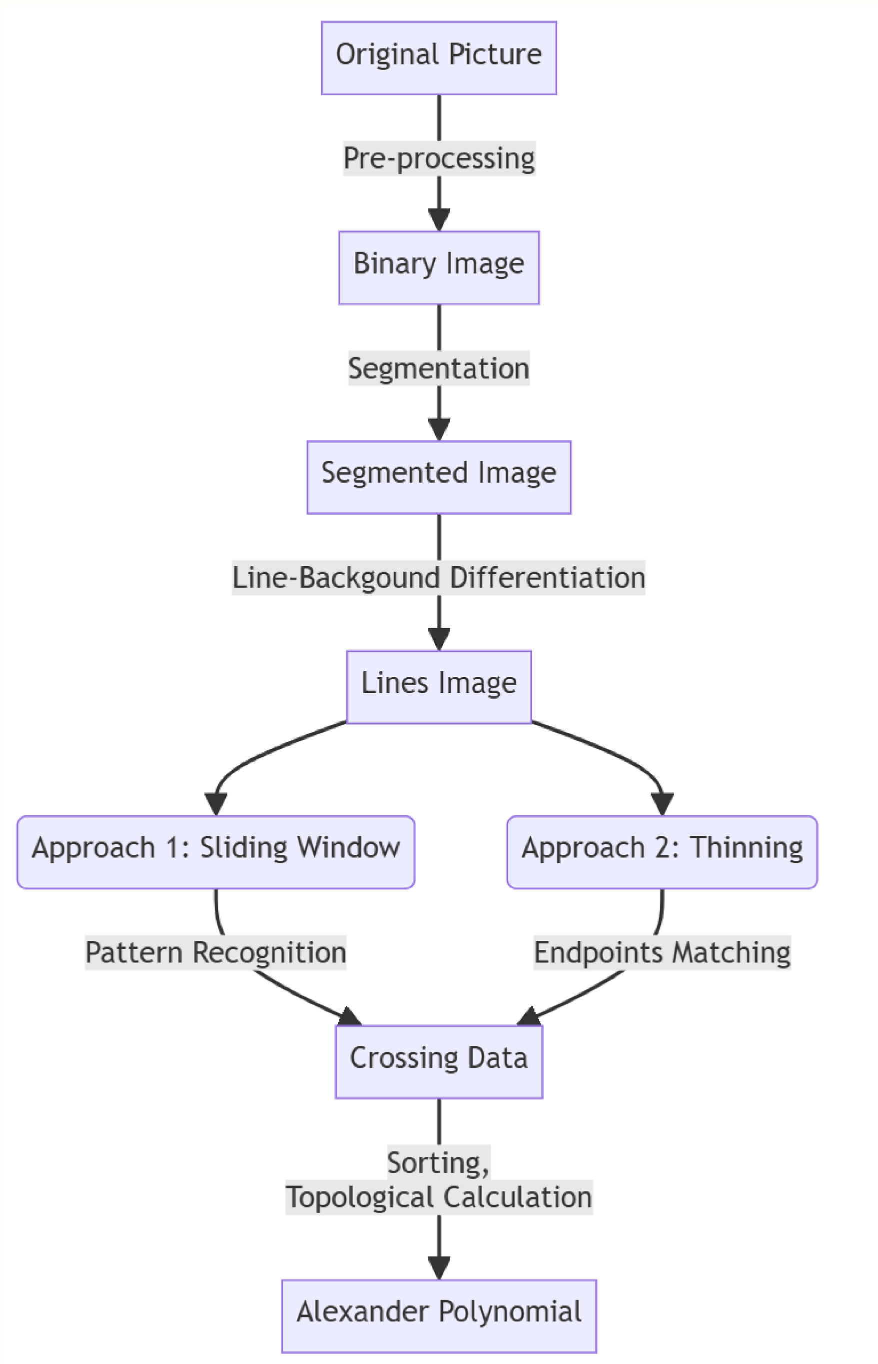
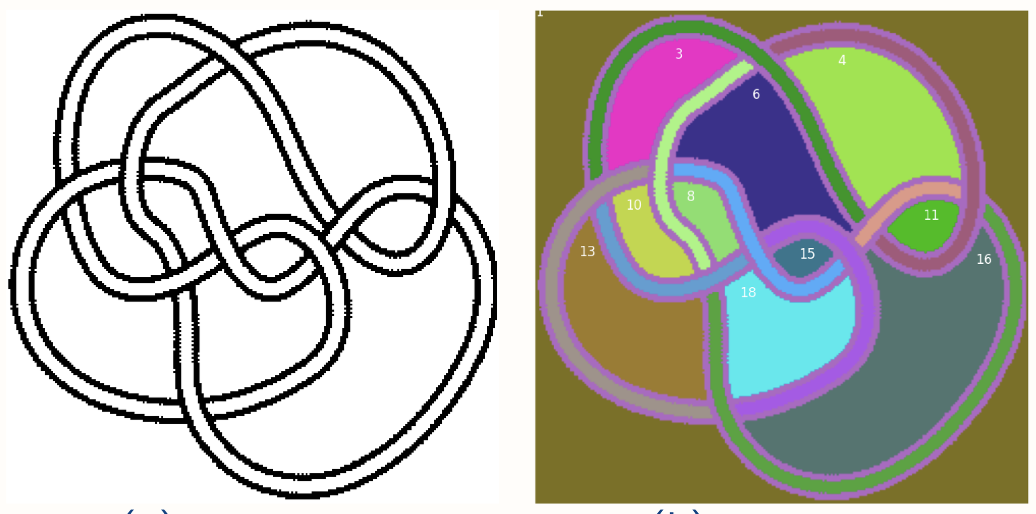


Figure 2: Diagram of Overall Process

Union-Find Labeling

Union-Find Labeling is used to identify and label closed connected components within an image. This technique assigns a unique label to each connected region in the image, allowing for the differentiation and analysis of distinct objects or regions. By grouping connected pixels together, Union-Find Labeling facilitates the segmentation of an image into meaningful components, which can then be further analyzed or processed for various applications in computer vision and image processing.



(a) Original Image (b) Labeled Image

Figure 3: Segmentation of the Binarized Image

Maximum Inscribed Circle

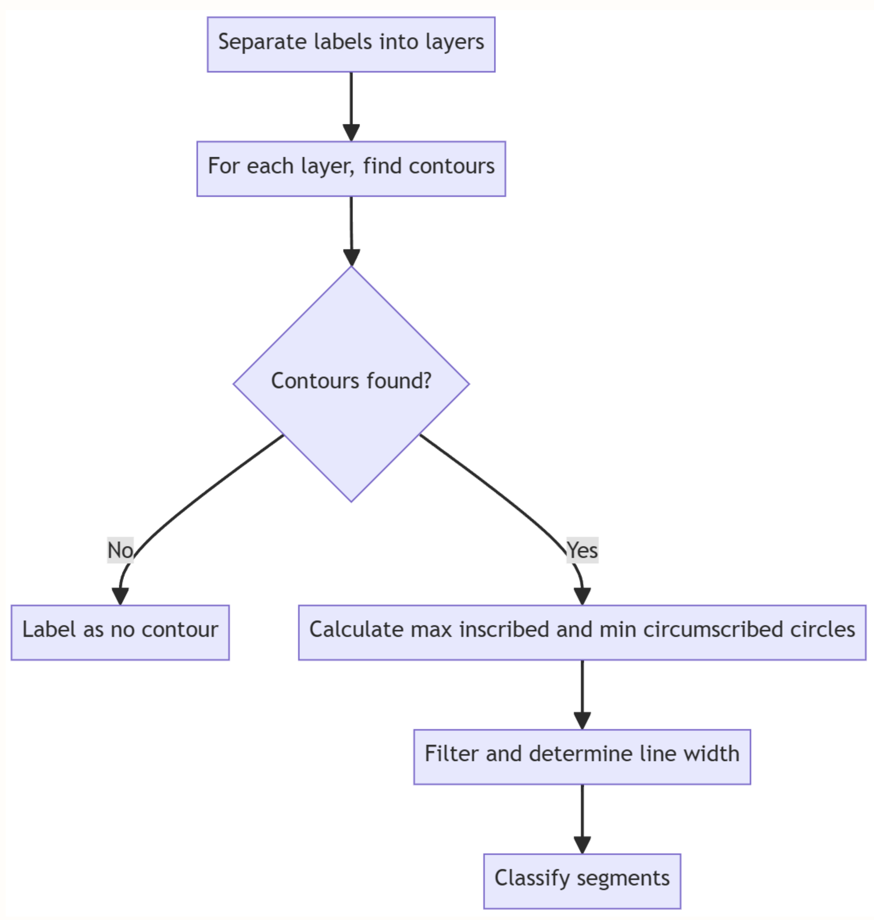


Figure 4: Process of distinguishing line segments from the background in the Python file is_line.py.

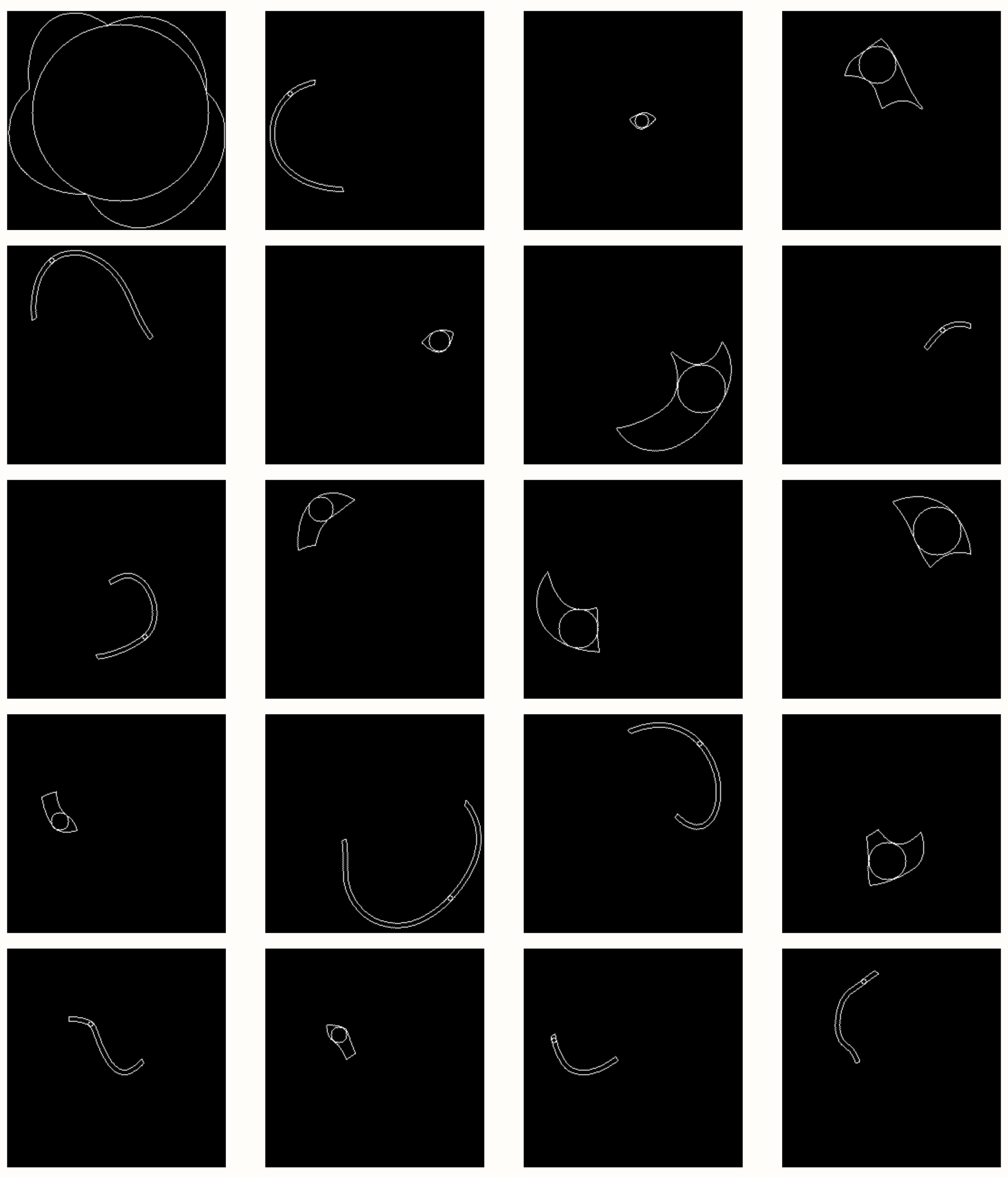
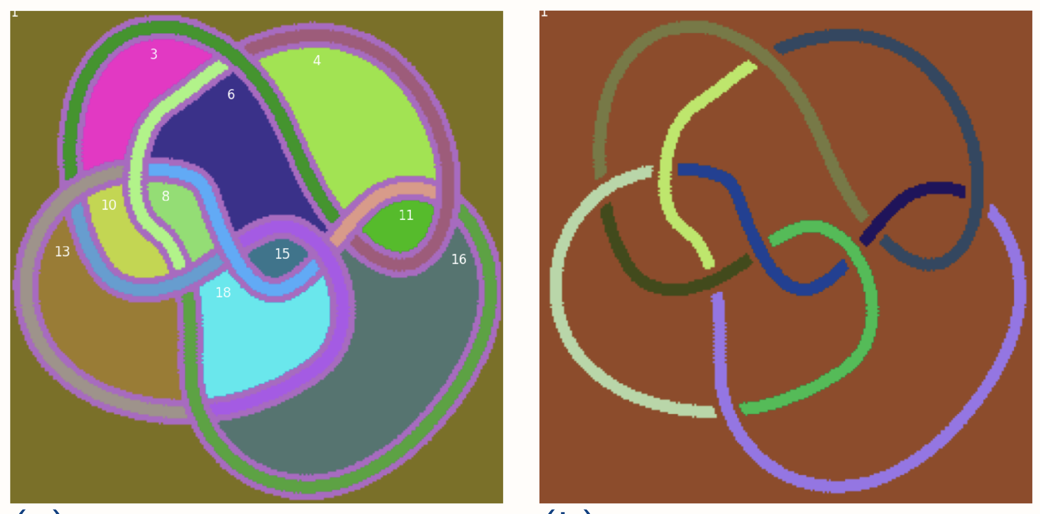


Figure 5: For each layer in the labeled image we find the Maximum Inscribed Circle



(a) Labeled Image with Background Labeled (b) Label Image without Background Labeled

Figure 6: Segmentation of the Binarized Image

Results and Discussion

We tested 151 images from the Rolfsen Knot Table. Lower methods are always superior in accuracy.

Table 1: Comparison of Knot Detection Methods

Method	Accuracy	Runtime	Fail Cases
Three-Color	37.1%	110s	(1)(2)(3)(4)
Alternating Pattern	42.4%	217s	(2)(3)(4)
Adaptive Window	66.2%	356s	(3)(4)
Adap. Win. & Alt. Pat.	95.4%	552s	(3) (4)
Thinning	96.7%	207s	(4)

(1) Close crossings (2) Very close crossings
(3) Large crossings (4) Background errors

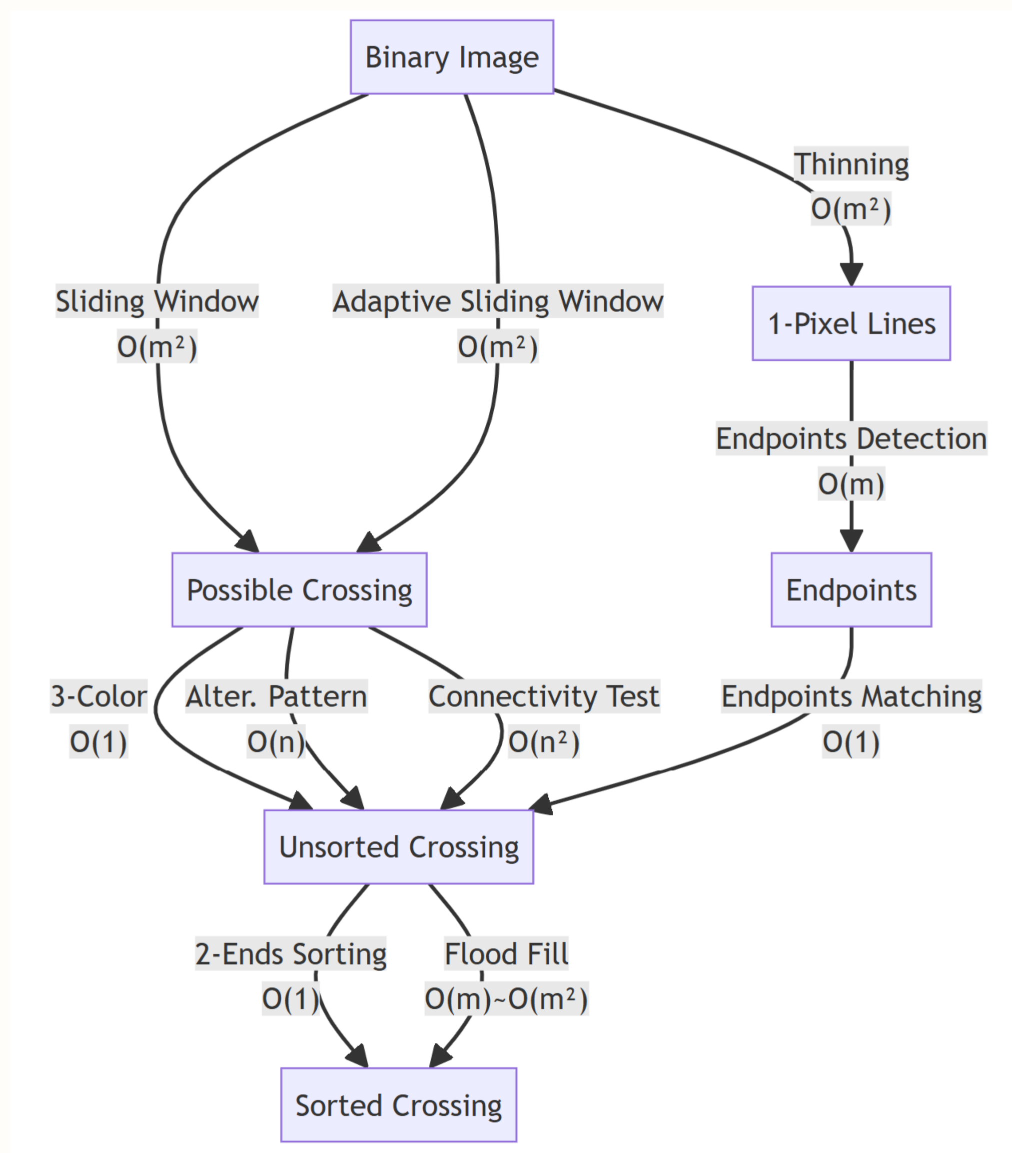


Figure 7: Complexity of Approaches

(m: size of the picture. n: size of the sliding window.) Three-Color method requires $O(m^2)$ time and space complexity, and Thinning method requires $O(m^3)$.

Conclusion

We presented two approaches for knot detection: sliding window and thinning. The Thinning method achieved the best accuracy at 96.7%, avoiding background errors and too-large-crossing errors. The Three-Color method is the fastest though not robust against unregulated inputs. It could be improved with regulated images or restrictions like Alternating Pattern test.