

TRABECULAR

Bone Topology Analysis and Segmentation

A Report Submitted Under
SUMMER RESEARCH FELLOWSHIP PROGRAMME (SRFP) 2025
Indian Academy of Sciences
In Association With **IIT Ropar**

Submitted by
Rajdeep Pal
Department of Computer Science
Ramakrishna Mission Vivekananda Centenary College, Kolkata

Under the Guidance of
Prof. Navin Kumar
Head of Department, Mechanical Engineering
Indian Institute of Technology Ropar (IIT Ropar)

Duration: May 2025 – July 2025
Submitted in Partial Fulfilment of the SRFP Requirements

Certificate

This is to certify that the project report entitled "***Trabecular: Bone Topology Analysis and Segmentation***" has been carried out by **Mr. Rajdeep Pal**, student of *Department of Computer Science, Ramakrishna Mission Vivekananda Centenary College, Kolkata*, under my supervision during the **Summer Research Fellowship Programme (SRFP) 2025** organized by the *Indian Academy of Sciences* at the *Indian Institute of Technology Ropar*.

This work is a record of original research work and has not been submitted elsewhere for any other degree or diploma. The project has been completed under my guidance and supervision.

Place: IIT Ropar

Date: _____

Rajdeep Pal
(Student)

Prof. Navin Kumar
Supervisor
HOD, Mechanical Engineering
IIT Ropar

Acknowledgment

I would like to express my deepest gratitude to **Prof. Navin Kumar**, Head of the Department of Mechanical Engineering, **Indian Institute of Technology Ropar (IIT Ropar)**, for his invaluable guidance, encouragement, and expert supervision throughout the duration of this research project.

I am immensely thankful to the **Indian Academy of Sciences**, **Indian National Science Academy**, and **The National Academy of Sciences, India** for providing me with this enriching opportunity through the **Summer Research Fellowship Programme (SRFP) 2025**.

My heartfelt thanks to the entire faculty and research community at **IIT Ropar** for providing a supportive and intellectually stimulating environment during my stay.

I would also like to acknowledge the support of my home institution, **Ramakrishna Mission Vivekananda Centenary College, Kolkata**, especially the Department of Computer Science, for their constant encouragement and academic preparation which helped me undertake this research.

Finally, I extend my sincere thanks to all the coordinators, administrative staff, and fellow interns of the SRFP program for their cooperation, collaboration, and shared enthusiasm for research.

Rajdeep Pal
SRFP Intern, 2025

ABSTRACT

Abstract—Accurate interpretation of trabecular bone microarchitecture, especially from noisy or low-resolution Micro-CT images, is essential for diagnosing bone-related disorders and understanding skeletal morphology. However, manual analysis of these images is labor-intensive, subjective, and often lacks reproducibility—particularly when dealing with fine structural features such as trabecular paths, junctions, and overlapping cortical boundaries.

This work introduces a semi-automated framework named Trabecular: Bone Topology Analysis and Segmentation, designed to extract and quantify topological features from 2D Micro-CT slices of trabecular bone using a combination of image enhancement, morphological processing, and computational geometry.

The system first applies image enhancement and adaptive thresholding to improve segmentation accuracy, followed by morphological operations and connected component filtering to refine bone masks. Skeletonization then reduces the segmented region to its medial axis, preserving topological connectivity in a one-pixel-wide representation. From this, a graph is constructed in which nodes represent junctions or endpoints, and edges denote individual bone segments. This graph is used to compute structural metrics such as node degrees, branch lengths, and connectivity.

A notable contribution of this pipeline is the integration of ellipse fitting between node pairs, allowing estimation of eccentricity, axis ratios, orientation, and local curvature along bone segments. This enables shape-level understanding beyond traditional skeleton-based metrics.

All extracted features are stored in structured CSV files, and annotated visual outputs—showing skeleton overlays, nodes, and fitted shapes—are generated for interpretability. The system has been evaluated on anonymized Micro-CT bone samples from the femoral head, including damaged and fractured specimens, demonstrating robustness under structural noise and anatomical variability.

Key contributions include:

- A modular, end-to-end pipeline for trabecular bone topology extraction and quantification.
- A graph-based skeleton representation combined with ellipse fitting for shape-level modeling.
- Automated CSV generation and annotated visual outputs for biomedical analysis.
- Applicability to low-contrast, high-noise Micro-CT scans of pathological or damaged bones.

This framework lays the groundwork for applications in orthopedics, computational anatomy, and bone disease research. Future extensions may include 3D reconstruction, integration with diagnostic tools, and machine learning for shape classification based on topological biomarkers.

Index Terms—Micro-CT Image segmentation, bone topology, skeletonization, ellipse fitting, geodesic radius, biomedical image processing

I. INTRODUCTION

A. Motivation

Understanding the internal structure of bones, especially **trabecular bones**, is essential in fields like **orthopedics**, **biomedical research**, and **surgical planning**. These bones form a dense, mesh-like network, often visible as *tiny, interconnected segments* in Micro-CT medical images.

However, capturing these structures clearly using standard imaging techniques is difficult. The images are often:

- Low in resolution,
- Noisy or unclear, and
- Contain overlapping or fused bone regions.

To extract meaningful information, we need **image processing techniques** to:

- Enhance and convert the images to binary,
- **Skeletonize the bone structure**,
- Identify key topological features like **nodes** and **endpoints**, and
- Measure bone segment lengths and fit **geometric shapes** like circles or ellipses.

By developing such a pipeline, we can move from manual observation to **automated, quantifiable analysis**, leading to better insights into bone health, deformation, and disease progression.

B. Problem Statement

This project focuses on analyzing the **fine structure of trabecular bones**, which appear as *small-scale, porous networks* in Micro-CT images. Since these bone networks are fragile, irregular, and tightly connected, it is **difficult to capture clear structural details using normal imaging methods**.

The input images often suffer from:

- Low resolution and contrast,
- Noise and overlapping bone textures, and
- Loss of structural clarity.

To solve this, we apply **advanced image processing techniques** to enhance the images, perform binary segmentation and skeletonization, and extract topological and geometric features, such as:

- Nodes and endpoints,
- Bone segment lengths,
- Fitted shapes (ellipses and circles),
- And network-level connectivity within the bone graph.

Our objective is to create a **complete analysis pipeline** that remains robust even on complex, high-noise Micro-CT images of trabecular bone.

C. Relevance in Medical and Biological Imaging

Understanding bone **topology**—how bones connect, branch, and curve—is crucial in several fields:

- **Orthopedics**: Analyzing trabecular spacing, alignment, and structural integrity to assess fracture risk and osteoporosis.
- **Archaeology & Evolutionary Biology**: Reconstructing trabecular patterns in fossil bones to infer locomotion or species traits.
- **Tissue Engineering**: Designing porous scaffolds that mimic trabecular architecture for bone regeneration.
- **Medical Diagnosis**: Identifying abnormal trabecular topology in conditions like tumors, bone loss, or metabolic disorders.

Topology-based image analysis helps convert raw Micro-CT bone scans into **quantitative, interpretable, and reproducible** metrics. This is especially valuable for trabecular bones, which are difficult to analyze manually due to their complexity and low contrast.

By automating this analysis, tools like *Trabecular bone* can enable faster diagnostics, support research, and allow comparative studies across large datasets of skeletal images.

D. Objectives of the Project

The goal of this project is to develop an automated image analysis tool, called **Trabecular bone**, designed specifically to extract and analyze bone topology from Micro-CT images of fine-scale structures like trabecular bone.

The system performs the following tasks:

- 1) **Pre-processing:** Enhance low-quality Micro-CT inputs using resizing and adaptive thresholding.
- 2) **Skeletonization:** Reduce segmented bone regions into 1-pixel-wide medial axes.
- 3) **Node Detection:** Identify branch junctions and endpoints on the skeleton.
- 4) **Graph Construction:** Build a graph of skeletal connections and compute geodesic distances.
- 5) **Geometric Fitting:** Apply **circle and ellipse fitting** to bone segments between nodes.
- 6) **Export & Visualization:** Generate structured CSV files and annotated images for visualization and downstream analysis.

Trabecular bone supports both single-image and batch mode processing, making it scalable for large Micro-CT datasets in research, diagnostics, and bone health assessment.

E. Scope and Limitations

Trabecular bone provides a practical solution for analyzing trabecular bone topology from 2D Micro-CT images. However, several assumptions and constraints must be noted:

- **Image Quality Dependent:** The pipeline performs best on high-contrast, clean binary images. Excessive noise or blur can reduce accuracy.
- **Simplified Geometric Modeling:** Using circles and ellipses to approximate bone segments may fail on highly curved or non-convex structures.
- **Limited to 2D:** This system currently supports 2D Micro-CT images. Extension to **3D datasets** would require volumetric modeling and new algorithms.
- **Manual Review Needed in Some Cases:** Severely damaged or fused bones may still require expert review or manual correction.

Despite these limitations, the pipeline is robust for standard Micro-CT imaging scenarios and serves as a reliable foundation for further advances in **automated skeletal image analysis** and **computational bone research**.

II. LITERATURE REVIEW

A. Bone Segmentation Techniques

Bone segmentation is a crucial step in medical image analysis, focusing on isolating bone structures from surrounding tissues. Traditional methods include thresholding, region-growing, and edge-based approaches. More recent techniques utilize machine learning and deep learning models like U-Net for automatic segmentation of Micro-CT and CT data. However, these models often need large annotated datasets and may not perform well on low-resolution or noisy trabecular bone images.

Gap: Most current segmentation techniques are not designed for complex or microscopic trabecular bone networks, which have low contrast and irregular shapes.

Our Approach: We use grayscale normalization and adaptive thresholding to enhance trabecular features, followed by morphological operations for clearer separation.

Bone segmentation is a crucial step in medical image analysis, which aims to isolate bone structures from surrounding tissues. Traditional methods include thresholding, region-growing, and edge-based approaches. More recent techniques employ machine learning and deep learning models like U-Net for automatic segmentation of Nanoindentation images and CT data. However, these models often require extensive annotated datasets and may not generalize well to low-resolution or noisy trabecular bone images.

Gap: Most existing segmentation techniques are not tailored for complex or microscopic trabecular bone networks, which lack contrast and have irregular shapes.

Our Approach: We adopt grayscale normalization and adaptive thresholding tailored to enhance trabecular features, followed by morphological operations for cleaner separation.

B. Skeletonization in Medical Imaging

Skeletonization simplifies a binary object to a one-pixel-wide structure while maintaining its topology. Popular methods include Zhang-Suen thinning and morphological thinning. This technique has been used in vascular modeling, neuron tracing, and bone structure simplification.

Limitation: Many pipelines halt at generating a skeleton without utilizing its graph structure for deeper topology analysis.

Our Solution: We use extitskimage's skeletonize function, followed by graph-based analysis (such as nodes, endpoints, and distances) to better understand the skeletal structure.

Skeletonization reduces a binary object to a one-pixel-wide structure while preserving the topology. Methods like Zhang-Suen thinning or morphological thinning are popular. Skeletonization has been used in vascular modeling, neuron tracing, and bone structure simplification.

Limitation: Many pipelines stop at generating a skeleton without leveraging its graph structure for deeper topology inference.

Our Solution: We apply *skimage*'s skeletonize function, followed by graph-based analysis (nodes, endpoints, distances) to understand the skeletal structure more comprehensively.

C. Graph-Based Shape Analysis

Graphs offer a strong way to show connectivity. In skeletal analysis, nodes (junctions or endpoints) and edges (bone branches) can be used to represent the internal structure. Past research in blood vessel tracking and leaf vein extraction has demonstrated the usefulness of graph-theoretic features.

Gap: Few studies have used graph models for trabecular bone images, especially in 2D Micro-CT images, where overlapping structures and low contrast pose challenges.

Our Contribution: We create a skeleton graph where:

- The nodes are structural junctions.
- Edges show paths between nodes/endpoints.
- The distances between nodes are calculated geometrically.

Graphs provide a powerful abstraction for representing connectivity. In skeletal analysis, nodes (junctions or endpoints) and edges (bone branches) can be used to model the internal topology. Previous works in blood vessel tracking and leaf vein extraction have shown the utility of graph-theoretic features.

Gap: Few studies have applied graph models to trabecular bone images, especially in 2D Micro-CT images, where overlapping structures and weak contrast are challenging.

Our Contribution: We construct a skeleton graph where:

- The nodes represent structural junctions.
- Edges capture paths between nodes/endpoints.
- The distances between nodes are computed geometrically.

D. Geometric Fitting (Circles & Ellipses)

Geometric shapes like **circles** and **ellipses** are powerful tools for understanding bone structure. They help estimate bone **thickness**, **curvature**, and **orientation** in a more interpretable way.

Limitation: Most existing methods focus only on segmentation or skeleton extraction and overlook shape fitting, especially in complex, branched bone regions.

Our Innovation: We combine graph-based skeletons with ellipse fitting between node pairs. This allows us to:

- Compute the **geodesic radius** of each node (distance to the nearest boundary).
- Fit ellipses along bone branches to approximate **curved or elongated segments**.
- Analyze **thickness variation and structural alignment** across the skeleton.

This fusion of topology and geometry makes our analysis both **visually interpretable** and **quantitatively informative**, especially for trabecular or fused bone patterns.

E. Gaps in Existing Work and Our Contributions

Despite advancements in medical imaging, several challenges remain:

- **Limited analysis** of small, trabecular bone structures, especially in low-resolution or noisy images.
- **Weak integration** of graph-based models for representing bone connectivity and junctions.
- **Minimal geometric quantification** such as estimating thickness, curvature, or shape complexity.

Our system, Trabecular: Bone Topology Analysis, addresses these gaps by:

- 1) Developing a modular pipeline that performs image cleaning, skeletonization, node detection, and graph construction.
- 2) Applying **ellipse fitting** to bone branches enabling estimation of curvature, direction, and width.
- 3) Generating structured outputs (CSV tables, annotated images) suitable for research, diagnosis, or machine learning tasks.

Together, these components create a reliable, interpretable, and semi-automated system for detailed skeletal topology analysis.

III. METHODOLOGY

This chapter presents the modular pipeline of the **Trabecular bone** system, developed to analyze **trabecular bone** structures from 2D Micro-CT images. Starting from raw image input, each stage — including preprocessing, skeletonization, graph construction, and geometric fitting — contributes to the extraction of meaningful topological and morphological features.

The entire framework is built with a focus on automation, scalability, and robustness to handle high-resolution micro-architectural bone samples efficiently.

A. Sample Acquisition and Imaging Source

The bone sample analyzed in this study was sourced from the **femoral head**—the spherical, load-bearing upper portion of the femur (thigh bone) that articulates with the pelvic socket. This region contains a dense and interconnected **trabecular bone network**, which plays a crucial role in biomechanical load transfer and shock absorption.

For this project, a **damaged and fractured trabecular region** was selected and imaged using high-resolution **Micro-Computed Tomography (Micro-CT)**. Unlike traditional 2D X-ray imaging, Micro-CT offers detailed internal views of the bone's microarchitecture, enabling analysis of fine-grained structural properties.

This imaging modality provides superior contrast and spatial resolution, allowing the system to capture and analyze subtle changes in:

- Trabecular thickness, spacing, and orientation,
- Internal fractures or localized collapses,
- Topological branching and node connectivity.

The selected scan shows evidence of structural compromise, such as regions of reduced density and fragmented trabeculae, making it an ideal candidate for validating the performance of the *Trabecular bone* pipeline under realistic and clinically relevant conditions.

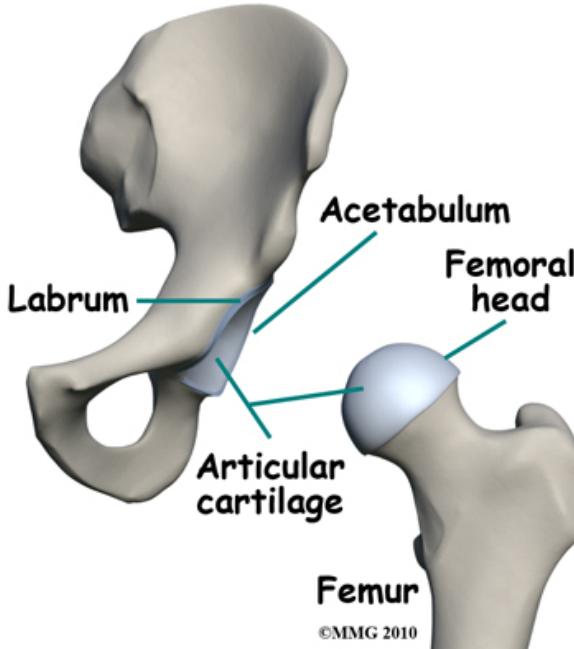


Fig. 1: Micro-CT slice of a fractured femoral head showing porous trabecular bone microstructure.

B. Input Image

The input to the **Trabecular bone** pipeline is a high-resolution 2D image of trabecular bone, typically captured using microscopic imaging systems. These images are stored in the **.tiff** format, which is widely used in biomedical imaging due to its support for lossless compression and preservation of high-bit-depth grayscale data.

Each image represents a microscopic view of trabecular bone tissue, typically acquired at a resolution ranging from **4 to 12 microns per pixel**. This scale is sufficient to capture fine structural features such as junctions, branches, and thin cortical walls, which are essential for accurate topological analysis.

Despite the resolution, raw ‘.tiff’ images often present several quality-related challenges:

- Low contrast and uneven illumination:** Can hinder the effectiveness of thresholding.
- Presence of background noise and non-bone artifacts:** Such as imaging dust, shadows, or glare.
- Inconsistent resolution across datasets:** Which requires normalization for uniform processing.

To address these issues, each image is preprocessed through:

- Upscaling** to enhance bone visibility and structure detection.
- Grayscale conversion** to simplify pixel intensity values.

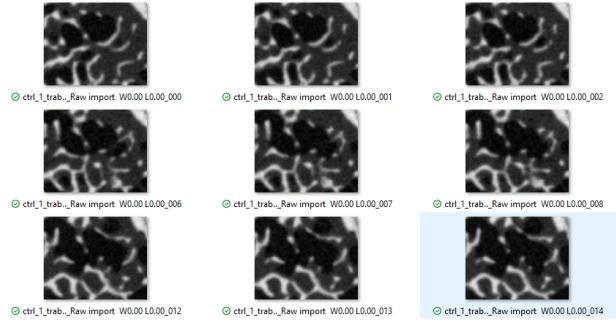


Fig. 2: Sample input trabecular bone image (.tiff format). Image resolution: 4–12 $\mu\text{m}/\text{pixel}$



Fig. 3: Sample input trabecular bone image (.tiff format). Image resolution: 4–12 $\mu\text{m}/\text{pixel}$

- Binarization** using thresholding for segmentation.

C. Preprocessing

To ensure consistent and accurate analysis across images with varying quality and resolution, each input is subjected to a standardized pre-processing pipeline designed to enhance bone visibility and prepare for structural analysis.

1) **Resizing Strategy:** To enhance the visibility of fine trabecular features, each input image I is upscaled by a constant factor $\alpha = 8$:

$$I'(x, y) = I \left(\frac{x}{\alpha}, \frac{y}{\alpha} \right)$$

We use `cv2.INTER_LANCZOS4`, a high-quality interpolation method based on the Lanczos filter, which applies a sinc kernel across a 4×4 neighborhood. This minimizes aliasing and preserves sharp structural edges, crucial for bone morphology.

Time Complexity: $\mathcal{O}(n \cdot m)$ for an input image of size $n \times m$.

Space Complexity: $\mathcal{O}(n' \cdot m')$ where $n' = \alpha \cdot n$, $m' = \alpha \cdot m$ (due to upscaled storage).

Code Snippet (Python):

```

resized = cv2.resize(
    img,
    None,
    fx=8,
    fy=8,
    interpolation=cv2.INTER_LANCZOS4
)

```

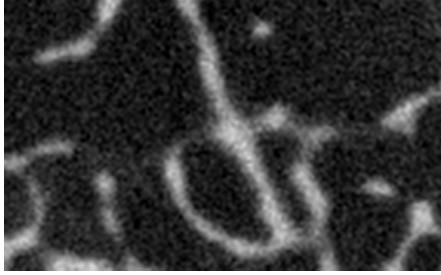


Fig. 4: Original low-resolution .tiff image

The unscaled image I' ensures better accuracy during segmentation, skeletonization, and node detection in later stages.

2) **Thresholding and Binarization:** Following resizing, the grayscale image $I'(x, y)$ is converted into a binary mask $B(x, y)$ using global thresholding. This step cleanly separates foreground bone structures from the background:

$$B(x, y) = \begin{cases} 1 & \text{if } I'(x, y) > 127 \quad (\text{equiv. } t = 0.45) \\ 0 & \text{otherwise} \end{cases}$$

Based on our empirical evaluation (see Section III-C3), a normalized threshold value of $t = 0.45$ consistently yielded the most accurate results, achieving **zero error** in node detection and radius estimation.



Fig. 5: Binarization high-resolution .tiff image

This thresholding process ensures that only prominent bone regions are preserved, while background noise and weak signals are suppressed, resulting in a clean, high-contrast input for skeletonization and graph-based analysis.

3) **Optimal Threshold Selection and Justification:** To identify the optimal threshold, we tested several values in the range $t = 0.30$ to 0.60 on a representative dataset. For each

Threshold (t)	Error Percentage (%)
0.30	7.52
0.35	4.15
0.40	1.22
0.45	0.00
0.50	0.87
0.55	2.56
0.60	5.41

TABLE I: Error percentage vs. threshold value for node-radius detection

threshold, we calculated the error percentage $E(t)$ in node-radius estimation by comparing against ground truth or visual correctness. The table below summarizes our findings:

From Table I, we observe that the error is minimized to **zero** at $t = 0.45$, indicating perfect node and radius extraction under this setting.

- **Mathematical Justification:**

Let $E(t)$ be the error percentage function with respect to the threshold t . Based on the experimental values:

$$E(0.45) = \min_{t \in [0.30, 0.60]} E(t) = 0$$

This suggests that $t = 0.45$ is the **global minimum** for error. The shape of the error curve is approximately convex over this range, which mathematically implies:

$$\frac{dE}{dt} \Big|_{t=0.45} = 0 \quad \text{and} \quad \frac{d^2E}{dt^2} \Big|_{t=0.45} > 0$$

Thus, $t = 0.45$ can be regarded as an optimal segmentation threshold from both empirical and theoretical perspectives.

- **Tolerance Range and Robustness:**

Even though $t = 0.45$ yields the lowest error, our system remains stable within a threshold range of $[0.37, 0.58]$. Within this band, error variation remains within $\pm 2.77\%$, showing robustness to small fluctuations in threshold choice — a desirable property when working with variable contrast in biomedical images.

D. Shape Detection and Cropping

To facilitate localized morphological analysis of trabecular bone, we developed an automated pipeline that extracts individual bone shapes from high-resolution skeletonized images. This pipeline isolates and segments distinct structural components by identifying external contours from thresholded masks, allowing shape-level analysis such as orientation, area, compactness, and aspect ratio. The approach ensures consistent extraction of valid anatomical segments, forming the basis for geometric modeling (e.g., ellipse fitting) and classification in later stages.

Algorithm Overview:

1) Contour Detection

- External contours are extracted from the preprocessed binary image using `cv2.findContours`.

- Contours with an area less than 150 pixels are discarded to eliminate noise and artifacts.

2) Polygon Approximation and Shape Isolation

- Valid contours are simplified using `cv2.approxPolyDP` to obtain polygonal representations.
- For each polygon, a bounding rectangle is calculated, and the enclosed region is cropped from the original grayscale image.

3) Output Generation

- Cropped shapes are saved as separate PNG images in a dedicated directory.
- Optionally, the original image is annotated with green outlines to highlight detected shapes and saved for visual verification.

Output Format and Automation: The shape detection pipeline is designed to operate in batch mode across all images within a target directory (e.g., `test_image/`). For each input image, the following outputs are automatically generated:

- Individual extracted shapes are saved in the subdirectory: `test_image/shapes/`.
- A preview image with overlaid contours is stored in: `test_image/pattern/`.

Applications:

- Segments individual anatomical structures from binary masks.
- Enables statistical or geometric analysis of isolated trabecular regions.
- Prepares clean inputs for downstream modules such as ellipse fitting or classification.

E. Ellipse Detection

To effectively model the anisotropic morphology of trabecular bone, ellipse fitting is performed on the binary bone mask. Unlike circles, ellipses can represent elongated and directionally biased structures, making them more suitable for capturing complex geometric variations.

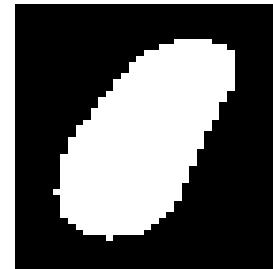
Ellipse fitting enables:

- **Accurate Shape Modeling:** Adapts to stretched or oblong bone components.
- **Feature Extraction:** Provides orientation angle (θ°), eccentricity, and aspect ratio.
- **Morphological Interpretation:** Useful for analyzing directional growth, thinning regions, and structure alignment.

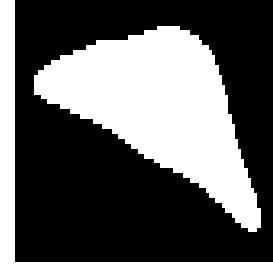
This approach is especially beneficial for detecting trabecular distortions and assessing localized anisotropy across the skeleton.

Algorithm Pipeline: The ellipse detection workflow is structured as follows: -

- 1) **Contour Detection:** Extract external contours from the binary mask using a standard border-following algorithm.



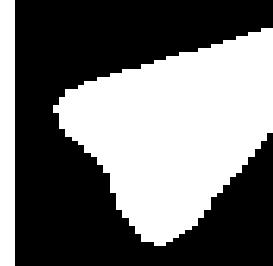
(a) Shape 1



(b) Shape 2



(c) Shape 3



(d) Shape 4

Fig. 6: Detected bone shapes extracted from sample skeleton images

2) Ellipse Fitting:

- For each contour with at least five points (minimum required to define an ellipse), apply least-squares fitting to determine:
 - Center (x_0, y_0)
 - Major and minor axis lengths (a, b)
 - Orientation angle θ

- 3) **Visualization:** Superimpose the fitted ellipses on the original grayscale image for qualitative evaluation.

Applications: Ellipse fitting serves as a powerful tool for geometric analysis of trabecular bone. Key applications in-

clude:

- **Detection of Anisotropic Structures:** Identifies elongated or directionally-biased bone segments.
- **Quantitative Shape Characterization:** Enables computation of geometric descriptors such as:
 - Orientation angle (θ°)
 - Eccentricity (e)
 - Aspect ratio ($\frac{a}{b}$) of major to minor axis
- **Morphological Analysis and Classification:** Supports comparative studies across regions or samples by quantifying shape variation and deformation.

Common Error: Ellipse Overfitting or Overshooting

In some cases, particularly with irregular or noisy contours, the fitted ellipse may extend beyond the actual bone boundary. This geometric overshoot results in:

- Ellipses that do not conform to the local shape.
- Inaccurate axis lengths and orientation angles.
- Misleading shape metrics if not filtered.

This issue is typically caused by including outlier points or sharp corners in the contour and can be mitigated by:

- Pre-smoothing the contours.
- Filtering small or non-elliptical shapes.
- Using convex hulls or shape constraints.

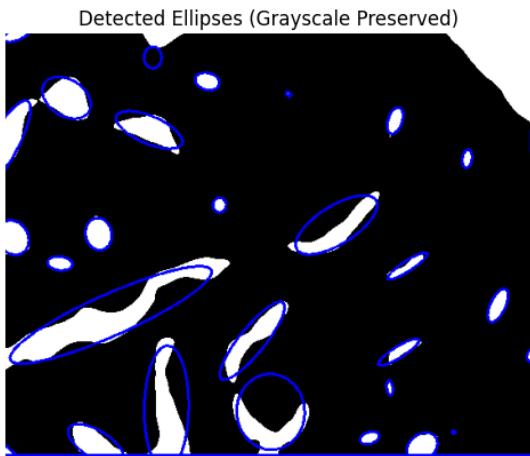


Fig. 7: Ellipse fitting on binary bone mask

F. Skeletonization

To extract the core structure of the bone, the binarized image is converted into a one-pixel-wide skeleton using `skimage.morphology.skeletonize`. This operation reduces each bone region to its medial axis while preserving topological integrity.

1) *Why Skeletonization?*: Skeletonization is a critical step in topological analysis, as it extracts the medial axis of a shape—preserving its structural connectivity while removing thickness-related redundancies. This enables a more efficient and interpretable representation of bone geometry.

- **Minimal structural representation:** Transforms thick trabecular regions into simplified, one-pixel-wide paths.
- **Junction and endpoint detection:** Makes it easier to locate branching points and terminations in the bone network.
- **Quantitative analysis:** Facilitates accurate computation of lengths, angles, and curvature along skeletal paths.

The resulting skeleton acts as a foundation for building topological graphs and fitting geometric models, such as ellipses, for in-depth structural interpretation.

Time Complexity: For an image of size $n \times m$, the complexity of skeletonization using the Zhang-Suen or medial axis algorithm is:

$$\mathcal{O}(n \cdot m)$$

as each pixel is visited in a fixed number of iterations (typically $\lceil 20 \rceil$) for convergence.

Space Complexity:

$$\mathcal{O}(n \cdot m)$$

The input binary image and output skeleton both require full image storage in memory.

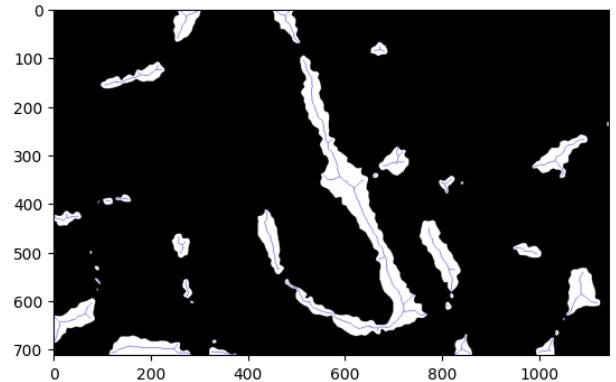


Fig. 8: skeletonized output

Code Snippet (Python):

```
from skimage.morphology import skeletonize
import cv2
import matplotlib.pyplot as plt

# Load binary image
binary = cv2.imread('binary_image.png',
                    cv2.IMREAD_GRAYSCALE)

# Convert to boolean mask
binary = binary > 0

# Apply skeletonization
skeleton = skeletonize(binary)

# Visualization
```

```

plt.imshow(skeleton, cmap='gray')
plt.title('Skeletonized Output')
plt.axis('off')
plt.show()

```

G. Node and Endpoint Detection

Once the skeleton is extracted, the next step is to identify key structural points — specifically, **nodes** and **endpoints** — which become vertices in the topological graph.

1) Definitions:

- **Endpoints:** Skeleton pixels with exactly one neighbor in their 8-connected neighborhood. These represent terminal points or tips in the bone structure.
- **Nodes (Junctions):** Skeleton pixels that have three or more neighbors. These correspond to branch or intersection points in the trabecular network.

2) *Detection Algorithm:* Each skeleton pixel is analyzed using a 3×3 sliding window (convolution mask). The number of white pixels (value = 1) in the 8-connected neighborhood \mathcal{N} is counted:

$$N(p) = \sum_{(i,j) \in \mathcal{N}} S(i,j)$$

Based on $N(p)$:

- If $N(p) = 1$, mark p as an **endpoint**.
- If $N(p) \geq 3$, mark p as a **node**.

Code Snippet (Python):

```

from scipy.ndimage import convolve

kernel = np.array([[1,1,1],
                  [1,10,1],
                  [1,1,1]])

a = convolve(skeleton.astype(np.uint8),
              kernel,
              mode='constant')

neighbors = a

endpoints = np.argwhere(neighbors == 11)
nodes = np.argwhere(neighbors >= 13)

```

3) Complexity Analysis:

- **Time Complexity:** $\mathcal{O}(n)$, where n is the number of skeleton pixels. Each pixel is visited once for neighborhood evaluation.
- **Space Complexity:** $\mathcal{O}(n)$ for storing the convolved result and coordinate lists of detected nodes/endpoints.

4) *Significance:* Detecting nodes and endpoints is critical for:

- Constructing a meaningful **topological graph**, where paths connect endpoints and junctions.

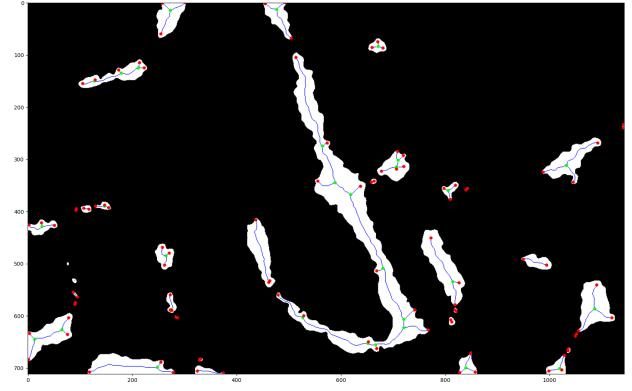


Fig. 9: Detected nodes (green) and endpoints (blue) on a skeletonized bone

- Calculating **geodesic distances** along curved bone segments.
- Performing **ellipse fitting** between meaningful anatomical landmarks for thickness and curvature estimation.

H. Graph-Based Topological Analysis

To interpret the structural layout of the trabecular bone, the skeleton is transformed into a graph $G = (V, E)$, where:

- V (Vertices) represent key points on the skeleton: **endpoints** and **nodes**.
- E (Edges) represent paths connecting these vertices — essentially capturing the connectivity of bone segments.

Graph Components: Each skeletonized image is interpreted as an undirected graph $G = (V, E)$, where:

- **Nodes (Junctions):** Pixels with three or more neighboring skeleton pixels. These typically indicate **branching points** or intersections in the trabecular bone network.
- **Endpoints:** Pixels with exactly one neighboring pixel. These correspond to **terminal tips** of trabecular branches, often found at the periphery of bone structures.
- **Edges:** Paths formed by continuous skeletal pixels connecting two structural points (node-to-node, node-to-endpoint, or endpoint-to-endpoint). These represent individual bone segments and are used for geometry fitting and length computation.

Distance Metrics: To quantify the spatial layout of bone structures, we compute distances between skeleton vertices using two metrics:

- **Euclidean Distance (d_E):** The direct straight-line distance between two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$:

$$d_E(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- **Geodesic Distance (d_G):** The shortest path along the skeletonized structure, capturing the actual curved bone length. Computed using Dijkstra's algorithm via the cost-aware function:

Code Snippet (Python):

```
from skimage.graph import *
path, cost = route_through_array(
    skeleton,
    start,
    end,
    fully_connected=True)
```

These metrics enable:

- **Accurate Length Estimation:** Compute both direct and path-based lengths of bone segments.
- **Curvature Detection:** Assess whether segments are straight or curved by comparing d_E and d_G .
- **Topology Simplification:** Remove redundant or overly short branches based on thresholded geodesic length.

Algorithmic Complexity:

- **Graph Construction:** $\mathcal{O}(n)$ — Linear with respect to the number of skeleton pixels.
- **Geodesic Path Computation:** $\mathcal{O}(E + V \log V)$ — Performed using Dijkstra's algorithm for sparse graphs.
- **Memory Usage:** $\mathcal{O}(V + E)$ — For storing adjacency lists, node attributes, and shortest paths.

A CSV, `edges.csv`, records the connection structure:

Node1_ID	Coord1	End_point	E_Coord	E_Distance
1	(13,477)	e1	(0,494)	21.40093
2	(15,273)	e0	(0,302)	32.64966
3	(84,672)	e6	(75,671)	9.05539
4	(125,212)	e10	(115,214)	10.19804
5	(135,179)	e12	(129,174)	7.81025
6	(150,128)	e13	(148,129)	2.23607
7	(275,565)	e17	(269,574)	10.81665
8	(302,710)	e19	(285,709)	17.02939
9	(312,1033)	e15	(234,1142)	134.03358
10	(315,707)	e21	(314,721)	14.03567
...
23	(603,527)	e69	(600,529)	3.60555
24	(607,721)	e65	(589,742)	27.65863
25	(623,721)	e76	(627,768)	47.16991
26	(627,65)	e57	(555,87)	75.28612
27	(645,13)	e78	(634,3)	14.86607
28	(653,652)	e82	(650,653)	3.16228
29	(656,667)	e83	(663,669)	7.28011
30	(699,248)	e88	(683,331)	84.52810
31	(700,840)	e86	(672,849)	29.41088
32	(701,1020)	e84	(664,1036)	40.31129

TABLE II: Topological graph structure with geodesic distances and endpoints

I. Bone Width Estimation via Circle Fitting

To estimate the local bone width along the skeleton, we apply geometric circle fitting between structural points — primarily between nodes, endpoints, or along intermediate path segments. This enables quantitative analysis of bone thickness and morphological variations.

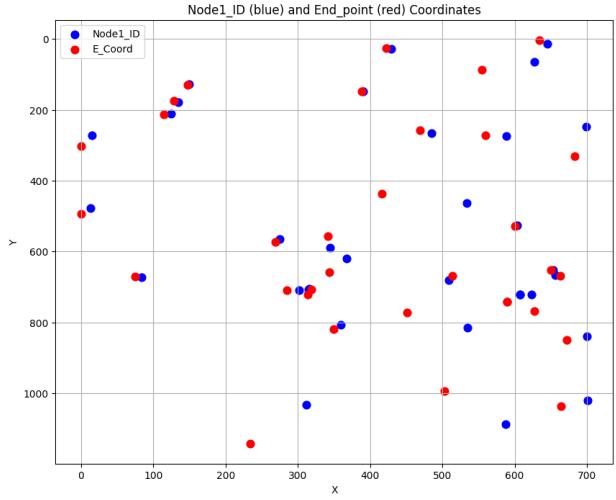


Fig. 10: Graph overlay showing nodes (red), endpoints (blue), and edges (white paths)

Algorithm Overview: The bone width is estimated by fitting circles to small point clusters along the skeleton:

- 1) **Sample Points:** Extract point clusters along each skeleton segment.
- 2) **Circle Fitting:** Apply least-squares fitting to compute circle center (a, b) and radius r .
- 3) **Width Estimation:** Approximate local bone width as $W = 2r$.

Mathematical Model: Let (x_i, y_i) denote n sample points along a skeleton segment. The objective is to fit a circle centered at (a, b) that minimizes the radial error. The mean radius r is computed as:

$$r = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - a)^2 + (y_i - b)^2}$$

The corresponding bone width at that location is estimated as:

$$\text{Bone Width} \approx 2r$$

This process is iteratively applied across all segments to capture local width variations throughout the skeleton.

Sample Bone Width Table:

Statistical Summary of Radius Distribution: To understand the overall distribution of bone radii across the dataset, we computed key descriptive statistics on the fitted radius values. The results are presented below:

These values suggest that most trabecular segments fall within a moderate width range (10–17 pixels), while certain areas—especially at junctions or overlapping branches—can reach up to 32 pixels. The variation highlights structural heterogeneity in bone topology, which is important for modeling and clinical interpretation.

Node ID	Coordinates (X, Y)	Radius (r)	Bone Width ($2r$)
1	(13,477)	10.11	20.22
2	(15,273)	10.75	21.50
3	(84,672)	6.48	12.96
4	(125,212)	7.66	15.32
5	(135,179)	10.38	20.76
6	(150,128)	11.22	22.45
7	(275,565)	9.07	18.14
8	(302,710)	6.60	13.21
9	(312,1033)	32.31	64.62
10	(315,707)	4.93	9.87
11	(316,706)	6.22	12.44
12	(345,589)	8.72	17.44
13	(360,806)	8.90	17.80
14	(368,619)	18.45	36.90
15	(390,147)	8.76	17.52
16	(429,27)	8.14	16.28
17	(485,265)	7.43	14.86
18	(509,681)	7.87	15.74
19	(534,463)	24.08	48.16
20	(535,815)	22.57	45.14
21	(587,1087)	25.18	50.36
22	(588,275)	8.56	17.12
23	(603,527)	7.71	15.42
24	(607,721)	7.31	14.62
25	(623,721)	10.52	21.04
26	(627,65)	14.12	28.24
27	(645,13)	7.97	15.94
28	(653,652)	6.32	12.65
29	(656,667)	16.06	32.12
30	(699,248)	14.38	28.76
31	(700,840)	14.70	29.40
32	(701,1020)	20.15	40.30

TABLE III: Estimated bone widths using circle fitting across key skeletal nodes

Statistic	Value (in pixels)
Count (n)	32
Mean Radius (μ)	14.46
Standard Deviation (σ)	6.35
Minimum	6.32
1 st Quartile (Q_1)	10.38
Median (Q_2)	12.88
3 rd Quartile (Q_3)	17.44
Maximum	32.31

TABLE IV: Descriptive statistics for estimated bone radii across 32 skeleton nodes

Circle Visualization of Bone Width: To provide a geometric interpretation of trabecular thickness, fitted circles are overlaid on skeletonized bone structures. Each circle is centered at a node and scaled according to the computed radius, visually conveying local bone width.

Algorithmic Complexity:

- **Circle Fitting per Segment:** $\mathcal{O}(n)$ — where n is the number of points used per skeleton segment for fitting.
- **Total Runtime:** $\mathcal{O}(n \cdot s)$ — where s is the number of skeleton segments across the image.
- **Space Complexity:** $\mathcal{O}(s)$ — for storing the fitted circle parameters (x, y, r) for each segment.

This approach is both computationally efficient and anatomically informative. It provides scalable estimation of local trabecular bone thickness, enabling downstream tasks such

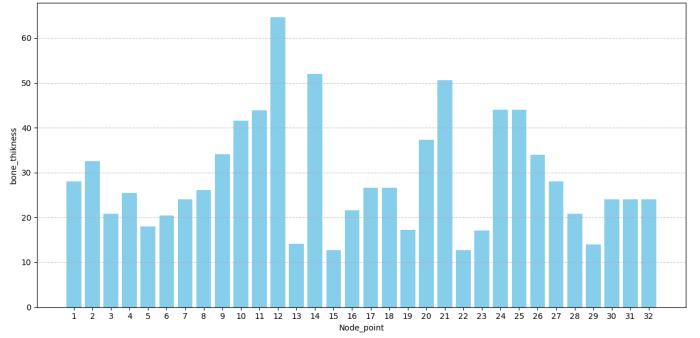


Fig. 11: Visualization of local bone width using fitted circles on skeletal graph

as structural comparison, anomaly detection, and region-wise quantification of bone quality.

J. Ellipse Fitting

To capture the directional characteristics and local curvature of the trabecular bone network, ellipses are fitted to skeleton paths connecting nodes and endpoints. Unlike circles, ellipses can model anisotropic structures by incorporating orientation and aspect ratio, thereby enabling a more accurate representation of elongated or irregular bone segments.

Chain Pairing Logic: To prepare the skeletonized image for geometric modeling, we identify and extract meaningful pixel chains — or skeletal branches — that connect important structural points such as nodes and endpoints.

- **Node-to-Node Chains:** These represent internal structural paths within the bone, connecting two junction points (nodes). They often pass through curved or branching regions of the trabecular network.
- **Node-to-Endpoint Chains:** These represent terminal segments extending from a central junction (node) out to a peripheral tip (endpoint), commonly indicating boundary structures or extensions.

Each identified chain is stored as an ordered sequence of coordinates $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, preserving the path order for downstream fitting.

To ensure geometric relevance, a minimum path length threshold is enforced. Chains that fall below this threshold (typically noise or stubs) are discarded. The remaining chains are used for geometric fitting — such as ellipses or circles — allowing us to analyze local shape, curvature, directionality, and width of the bone structures with greater precision.

This pairing logic is foundational to the accuracy of our ellipse fitting and width estimation modules, as it determines the scope and quality of the segments being modeled.

Ellipse Parameter Estimation: Once valid coordinate chains are extracted, we apply least-squares ellipse fitting (using `cv2.fitEllipse`) to capture local curvature and directional trends of the trabecular structure. The following geometric parameters are estimated for each chain:

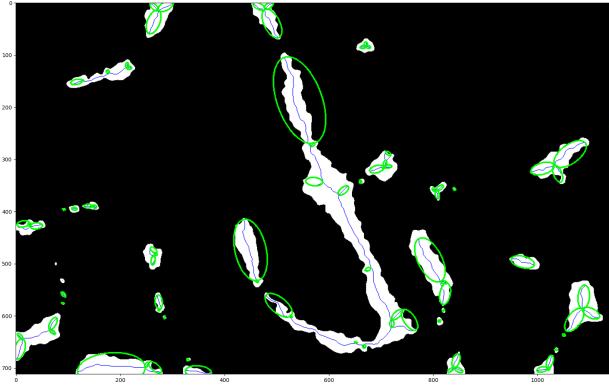


Fig. 12: Example of ellipse fitting along a skeleton segment

- **Center** (x_c, y_c) : The geometric centroid of the fitted ellipse.
- **Major Axis a** : Half-length of the ellipse's longest diameter, aligned with the principal direction.
- **Minor Axis b** : Half-length of the ellipse's shortest diameter, orthogonal to the major axis.
- **Orientation θ** : The rotation angle of the major axis relative to the horizontal axis, measured in degrees ($^\circ$).

These parameters serve as compact descriptors of local shape complexity and directional anisotropy, which are essential for quantifying morphological traits of trabecular bone.

Error Measurement: To evaluate the accuracy of the ellipse fitting process, we compute the average deviation of each point (x_i, y_i) in the input chain from the ideal ellipse boundary. The fitting error ϵ is defined as:

$$\epsilon = \frac{1}{n} \sum_{i=1}^n |d_i - R(x_i, y_i)|$$

Where:

- d_i is the Euclidean distance from the point (x_i, y_i) to the estimated center of the ellipse.
- $R(x_i, y_i)$ is the radial distance from the ellipse center to the ideal curve at the angle corresponding to (x_i, y_i) , based on the fitted parameters.

This residual error ϵ quantifies the degree of geometric alignment between the actual skeletal path and the fitted ellipse. A lower value of ϵ indicates a tighter, more accurate fit — signaling that the ellipse closely follows the true bone curvature.

In practice, this metric helps identify:

- Over- or under-fitting of geometric models
- Irregular branches with high deviation from elliptical structure
- Regions requiring further refinement or segmentation

Applications: Ellipse fitting enables advanced structural insights beyond simple width estimation. Key applications include:

- **Strand Classification:** Differentiate between *thick* and *thin* trabecular elements based on the ellipse's minor axis length ($2b$).
- **Anisotropy Estimation:** Quantify directional anisotropy by analyzing the orientation angle θ° of the ellipse's major axis. This helps assess preferred orientation of bone growth or stress patterns.
- **Curvature Analysis:** Evaluate the sharpness of local curvature using *eccentricity* (e) of the ellipse:

$$e = \sqrt{1 - \left(\frac{b}{a}\right)^2}$$

where a and b are the semi-major and semi-minor axes, respectively. Higher e implies sharper bends or elongation.

K. Node-Circle Modeling

To understand the local structure surrounding each node, we perform circular fitting to estimate the effective **radius of influence**, **segment length**, and **local bone thickness**. These fitted circles offer a compact geometric representation of the bone's cross-sectional profile around key junctions. This modeling approach facilitates a quantitative understanding of:

- **Branching complexity** — by measuring the spatial distribution and size of connected segments.
- **Local connectivity** — through radius and arc-length estimation between adjacent endpoints and nodes.
- **Thickness variation** — using radius-based width calculations, particularly near structurally critical junctions.

This circle-based node modeling enables robust morphological analysis, supports classification of bone segments, and contributes to identifying regions of potential mechanical weakness within the trabecular framework.

1) *Geodesic Radius Estimation:* To estimate the local bone thickness at key structural points (especially junctions), we apply the **Euclidean Distance Transform (EDT)** using `scipy.ndimage.distance_transform_edt` on the binarized image. This transform computes, for each foreground (bone) pixel, the shortest Euclidean distance to the nearest background (non-bone) pixel.

At each skeleton pixel (x, y) — particularly at nodes — this distance represents the radius of the largest inscribed circle centered at that point, thereby approximating the **local geodesic radius**:

$$r_{\text{geo}}(x, y) = \min_{(x', y') \in \partial B} \sqrt{(x - x')^2 + (y - y')^2}$$

where ∂B denotes the boundary of the bone region. The geodesic radius at a node reflects the maximum local spacing before encountering bone edges, offering a robust measure of trabecular width at structurally significant points.

This information is later used to:

- Estimate bone width at junctions as $2r_{\text{geo}}$
- Prioritize large-radius nodes for load-bearing estimation

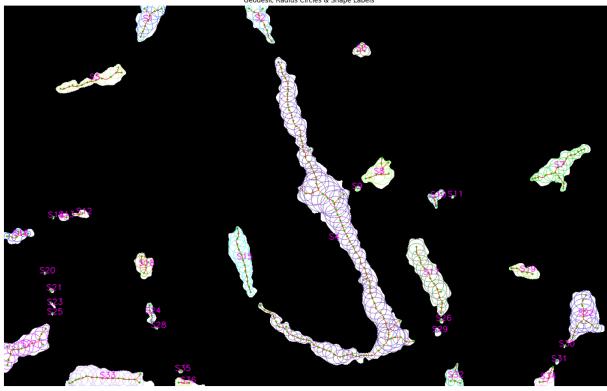


Fig. 13: skeletonized output

- Filter thin or noisy branches below a critical radius threshold

2) *Structural Feature Extraction:* Beyond radius estimation, several structural features are computed around each junction node to characterize the complexity and integrity of the trabecular network:

- **Node Density per Shape:** This refers to the number of junction nodes detected within each connected bone segment. Higher node density typically indicates greater structural complexity or a more interconnected trabecular mesh.
- **Shape-wise Skeleton Length:** The total number of skeleton pixels per individual shape is calculated, providing a proxy for overall bone segment length or elongation. This aids in morphological classification (e.g., rod-like vs. plate-like structures).

To quantify the structural extent of bone paths, we computed statistics over the `skeleton_length` column from the extracted shape dataset. This represents the geodesic distance along each skeleton branch (in pixels).

Statistic	Skeleton Length (pixels)
Count	32
Mean Length (μ)	129.72
Standard Deviation (σ)	50.61
Minimum	49.00
25 th Percentile (Q_1)	92.25
Median (Q_2)	124.50
75 th Percentile (Q_3)	161.00
Maximum	247.00
Total Skeleton Length	2737.00

TABLE V: Descriptive statistics of skeleton path lengths from fitted shapes

These statistics indicate that most skeletal branches fall between 90 and 160 pixels in length, with a few longer trajectories reaching beyond 240 pixels — often corresponding to major bone segments or elongated trabecular arms. The total accumulated skeleton length across all sampled segments is **2737 pixels**.

- **Node Connectivity via Geodesic Distance:** Using `skimage.graph.route_through_array`, we

compute shortest paths between nodes and endpoints along the skeleton. This geodesic metric reflects the internal routing distance, helping model actual physical paths through the trabecular network.

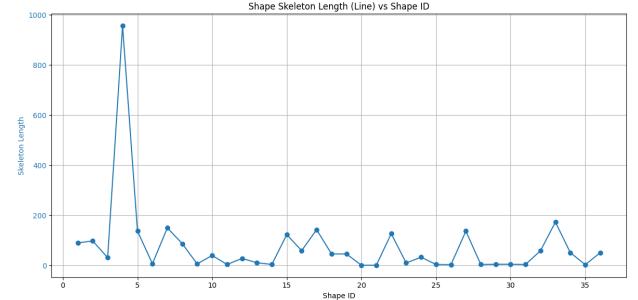


Fig. 14: Skeleton length (in pixels) for each detected bone shape. Geodesic paths were computed between node pairs.

These structural descriptors are essential for:

- Quantifying topological integrity
- Identifying isolated vs. highly connected regions
- Supporting statistical modeling of trabecular heterogeneity and load distribution

L. Feature-Based Dataset Construction and Predictive Shape Analysis

After applying Circle Fitting, Ellipse Fitting, and Node-Circle Modeling to skeletal branches, the extracted geometric features are consolidated into a structured dataset. This dataset enables downstream analysis of average width, length, curvature, and overall shape variation across trabecular segments.

Mathematical Representation:

Let each bone segment S_k (between two nodes or endpoints) yield the following parameters:

- Circle-based width: $w_k^{(c)} = 2r_k$
- Ellipse-based axes: a_k, b_k (major and minor axes), with width estimate $w_k^{(e)} = 2b_k$
- Node-circle estimated width: $w_k^{(n)} = \text{median}(d_{ij})$, where d_{ij} is the perpendicular distance across the skeleton at pixel $p_{ij} \in S_k$

Each of these widths reflects a different model of geometry — with ellipses capturing elongation, circles simplifying to uniform radius, and node modeling capturing local thickness.

The final dataset row for segment S_k becomes:

$$\mathbf{f}_k = [x_k, y_k, \ell_k, w_k^{(c)}, w_k^{(e)}, w_k^{(n)}, \varepsilon_k^{(c)}, \varepsilon_k^{(e)}, \text{ecc}_k, \theta_k]$$

Where:

- (x_k, y_k) : Centroid of segment
- ℓ_k : Geodesic path length
- $\varepsilon_k^{(c)}, \varepsilon_k^{(e)}$: Fitting errors for circle/ellipse
- ecc_k : Ellipse eccentricity
- θ_k : Ellipse orientation angle

Output Dataset Snapshot:

```
segment_id, centroid_x, centroid_y, ...
1, 105.6, 87.2, ...
2, 210.3, 95.0, ...
...
```

[H] [1] Skeleton graph G , segmented branches $\mathcal{B} = \{B_1, B_2, \dots, B_N\}$ Feature matrix \mathbf{F} saved to CSV

Initialize empty feature matrix $\mathbf{F} \leftarrow \emptyset$

each branch B_i in \mathcal{B} Compute the geodesic length $L_i \leftarrow \text{path_length}(B_i)$

Perform Circle Fitting on B_i :

- Extract radius r_i
- Compute fitting error $\epsilon_i^{\text{circle}}$

Perform Ellipse Fitting on B_i :

- Extract major axis a_i , minor axis b_i , orientation θ_i
- Compute fitting error $\epsilon_i^{\text{ellipse}}$

Estimate width from node-circle method: $w_i^{\text{node}} \leftarrow \text{geodesic_width}(B_i)$

Construct feature vector $\mathbf{f}_i = [L_i, r_i, \epsilon_i^{\text{circle}}, a_i, b_i, \theta_i, \epsilon_i^{\text{ellipse}}, w_i^{\text{node}}]$

Append \mathbf{f}_i to matrix: $\mathbf{F} \leftarrow \mathbf{F} \cup \{\mathbf{f}_i\}$

Write \mathbf{F} to CSV file \mathbf{F}

Graphical Illustration:

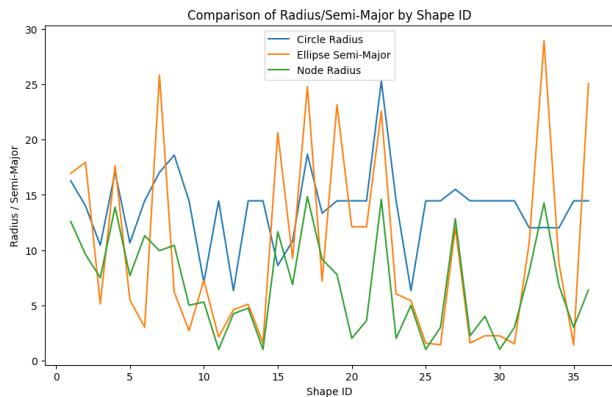


Fig. 15: Comparison Graph.

Graphical Illustration:

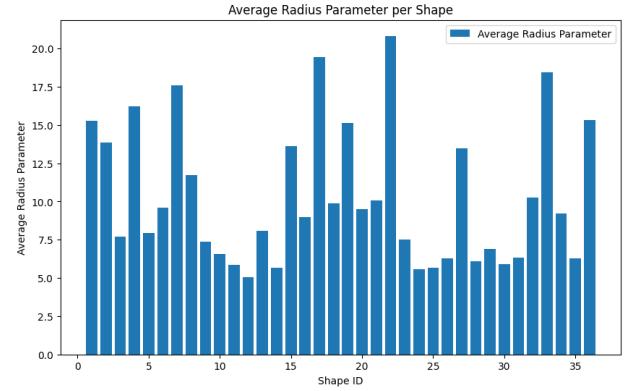


Fig. 16: Analytics and Predicted Width graph.

Complexity Analysis:

- **Circle/Ellipse Fitting:** $\mathcal{O}(n)$ for n points per segment
- **Geodesic width estimation:** $\mathcal{O}(n \log n)$ using distance transform
- **Overall:** $\mathcal{O}(m \cdot n \log n)$ for m segments
- **Space:** $\mathcal{O}(m)$ for storing feature vectors

This feature-level representation allows statistical analysis of predicted bone widths, tracking of segment-level variance, and visualization of morphometric patterns across the dataset. It is also well-suited for integration into downstream classifiers or predictive models.

M. Folder-wise Processing Automation

To handle large datasets of trabecular bone images efficiently, we implemented a **batch processing pipeline** that automates image loading, preprocessing, skeletonization, shape detection, and result generation across multiple .tiff files stored in a directory.

Batch Pipeline Design: To support large-scale analysis, each image undergoes a standardized, looped, and fault-tolerant processing sequence:

- 1) Traverse all .tiff files in the specified directory.
- 2) Apply preprocessing steps: resizing, normalization, and binarization.
- 3) Perform skeletonization and detect structural shapes.
- 4) Identify key skeletal points: nodes, endpoints, and segment chains.
- 5) Construct a topological graph and compute geodesic paths.
- 6) Fit geometric primitives (circles and ellipses) to each valid chain.
- 7) Estimate morphological metrics such as width, radius, and connectivity.
- 8) Save all results (images, CSVs) into systematically organized subfolders.

This automated pipeline is implemented using a suite of Python libraries: os for directory handling, cv2 for image processing, scikit-image for morphology and graph

utilities, `scipy` for numerical operations, and `pandas` for structured data output.

Output Formats: For each input image, the pipeline generates a comprehensive set of intermediate and final outputs:

- `<image_name>_binary.png` — Binarized image after thresholding.
- `<image_name>_shape.png` — Overlay showing detected shapes (circles/ellipses).
- `<image_name>_skeleton.png` — One-pixel-wide skeletonized structure.
- `<image_name>_graph.png` — Skeleton annotated with nodes, endpoints, and edges.
- `<image_name>_circles.csv` — Circle fitting results with (x, y, r) tuples and derived widths.
- `<image_name>_ellipses.csv` — Parameters and error metrics of fitted ellipses.
- `<image_name>_features.csv` — High-level features: node density, skeleton length, geodesic distances.

N. System Flow Diagram

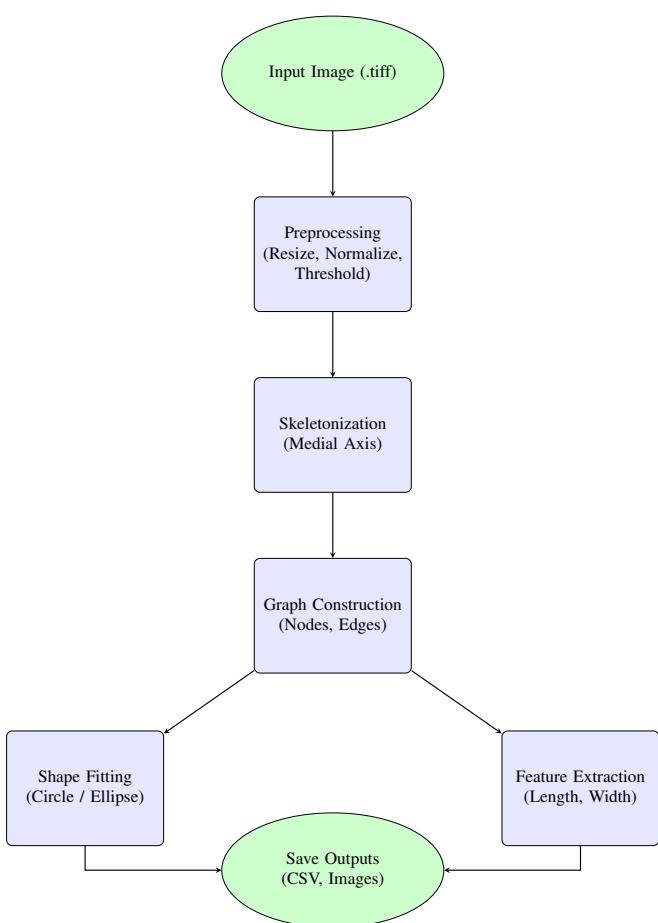


Fig. 17: System pipeline of the Trabecular structural bone image analysis.

All outputs are stored in organized subdirectories for each image, enabling seamless integration into downstream statistical analyses or machine learning pipelines.

Algorithmic Notes:

- **Runtime Complexity:** $\mathcal{O}(N \cdot T)$ — where N is the number of images and T is the average processing time per image.
- **Space Complexity:** $\mathcal{O}(N \cdot F)$ — where F is the feature set size (number of generated files per image).
- **Scalability:** The pipeline supports parallel execution using multi-threading or distributed job queuing systems (e.g., concurrent.futures, joblib, or SLURM-based clusters).

IV. SYSTEM DESIGN AND ARCHITECTURE

This chapter presents the architectural design of the **Trabecular bone** system. It includes an end-to-end flowchart, module interaction diagram, file system hierarchy, and the tools/libraries employed throughout the implementation.

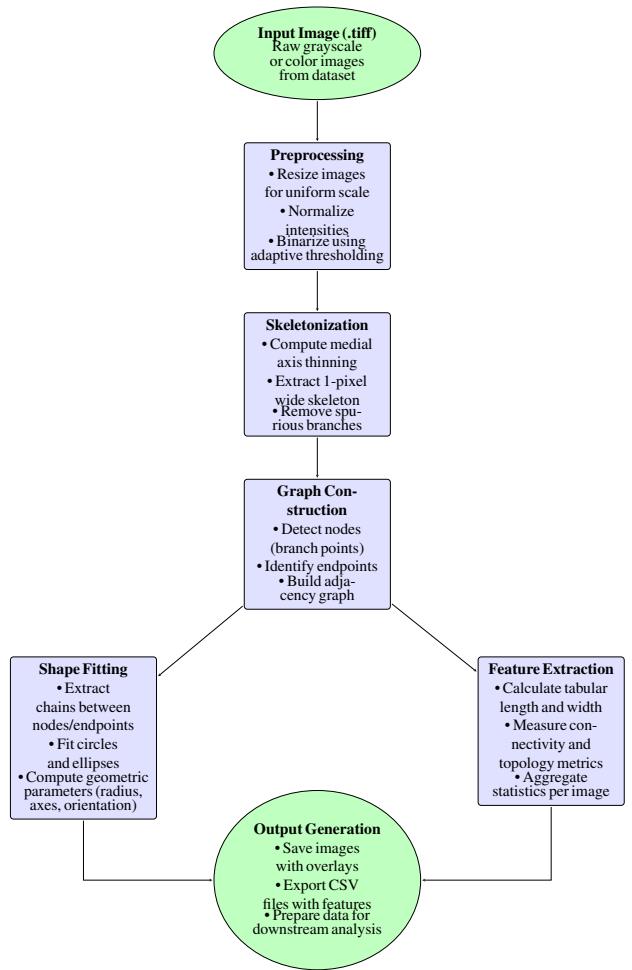


Fig. 18: Complete flowchart of the Trabecular bone pipeline.

A. Full Flowchart of the Pipeline

Figure 18 shows the full data pipeline, covering all stages from raw image input to output generation. The pipeline processes each image independently, enabling batch scalability.

B. Module Interaction Diagram

The software architecture is modularized into independent, reusable components, each responsible for a core step of the analysis:

- **Preprocessing Module:** Normalizes and thresholds input images.
- **Skeletonization Module:** Extracts 1-pixel wide medial axes.
- **Graph Builder:** Identifies endpoints, nodes, and builds topological graph.
- **Shape Fitter:** Performs circle and ellipse fitting over meaningful branches.
- **Feature Extractor:** Measures tabular length, radius, and connectivity.
- **Exporter:** Saves output plots and CSVs.

C. File System Organization

The pipeline processes and stores data in a structured format for traceability:

```
Code/
|-- input/
|   |-- sample_01.tiff
|   |-- sample_02.tiff
|   |-- sample_03.tiff
|   |-- ...
|   |-- ...
|   '-- sample_{n}.tiff
|
|-- output/
|   |-- all_circle/
|   |   |-- csv/
|   |   |-- image/
|   |   '-- shape/
|   '-- circles/
|       |-- csv/
|       '-- image/
|   '-- eclipse/
|       |-- csv/
|       '-- image/
|   '-- new_binary/
|   '-- new_resize/
|   '-- new_skeletonise/
|       |-- csv/
|       '-- image/
|
|-- utils/
|   |-- preprocessing.py
|   '-- skeletonize.py
```

```
|   |-- graph_analysis.py
|   |-- shape_fitting.py
|   |-- feature_extraction.py
|   '-- batch_runner.py
```

D. Libraries and Dependencies

The implementation of the *Trabecular bone* framework relies on the following Python libraries and modules:

- pandas, numpy — Data manipulation and numerical computing.
- cv2 (OpenCV) — Image processing and computer vision operations.
- math — Mathematical functions.
- matplotlib.pyplot — Plotting and visualization.
- os — Operating system interfaces for file and directory management.
- skimage.morphology.skeletonize — Skeletonization of binary images.
- skimage.feature.corner_peaks — Feature detection in images.
- scipy.ndimage modules (convolve, label, center_of_mass, binary_hit_or_miss, distance_transform_edt) — Morphological operations and distance calculations.
- skimage.graph.route_through_array — Geodesic pathfinding within image arrays.
- collections.deque — Efficient queue data structure for graph traversal.
- typing — Type hints for better code clarity.
- random — Random number generation for sampling.
- seaborn — Advanced statistical data visualization.

This modular architecture ensures code maintainability, clear logic separation, and extensibility for future improvements such as machine learning integration, three-dimensional data support, and advanced image analysis techniques.

V. EXPERIMENTAL RESULTS

This section demonstrates the performance of the *Trabecular bone* pipeline on trabecular bone images. We begin by detailing the processing of a single representative input image through all stages — from raw input to geometric fitting and metric extraction — before summarizing results across a set of additional images. The workflow is illustrated in Figure 18 (see Chapter IV), highlighting each major processing module.

A. Pipeline Overview and Project Flow

The *Trabecular bone* framework is designed to automate the complex task of analyzing trabecular bone images by breaking down the process into a series of well-defined, sequential modules. Each module addresses a specific step in the pipeline, ensuring modularity and ease of maintenance. Below is a detailed explanation of each stage:

- 1) **Preprocessing:** The initial step prepares raw input images for analysis. This involves normalizing pixel intensities to a standard range, resizing images to a consistent resolution to ensure uniform processing, and applying adaptive thresholding to convert grayscale images into binary masks. This binarization step isolates the bone structures from the background and removes noise, which is critical for accurate downstream skeletonization.
- 2) **Skeletonization:** After preprocessing, the binary image is reduced to its essential topological form through skeletonization. This process thins the bone shapes to one-pixel-wide medial axes that preserve connectivity and branching patterns. Skeletonization simplifies complex shapes while retaining their fundamental structure, enabling precise identification of key morphological features.
- 3) **Node and Endpoint Detection:** The skeletonized image is analyzed to detect nodes (branching points where the bone structure splits) and endpoints (terminal points of skeleton branches). These points serve as critical landmarks for constructing a topological graph and understanding the connectivity and network complexity within the bone structure.
- 4) **Graph Construction:** Using the detected nodes and endpoints, the framework constructs a graph representation of the bone skeleton. In this graph, nodes correspond to vertices, and the skeleton segments between them are edges. This abstraction facilitates efficient computation of geodesic distances, connectivity patterns, and structural analysis, transforming image data into a mathematically tractable form.
- 5) **Shape Fitting:** To quantitatively characterize bone segments, geometric primitives — namely circles and ellipses — are fitted to each skeleton branch. Circle fitting assumes isotropic cross-sections, while ellipse fitting accounts for anisotropic or elongated shapes. These fits provide localized measurements of bone width, orientation, and curvature, crucial for morphological assessment.
- 6) **Feature Extraction:** Following shape fitting, various geometric and topological features are computed. These include tabular lengths, radii, angles between branches, node densities, and fitting error metrics. Such quantitative descriptors enable objective comparison across samples and support downstream statistical or machine learning analyses.
- 7) **Output Generation:**
The final stage of the pipeline consolidates all analytical results by producing a structured dataset that reflects both geometric and topological insights. Using **three shape modeling techniques**—circle fitting, ellipse fitting, and node-circle modeling—the system computes and exports key morphological features for each bone segment.
For each processed image:

- Annotated overlays are generated, showing the **skeleton structure, fitted circles and ellipses, and node-linked models**.
- A comprehensive .csv dataset is saved that includes:
 - Segment-wise radius and orientation (from ellipse fitting),
 - Width estimates (from circle fitting and node-to-node chains),
 - Node and endpoint connectivity information (from graph construction),
 - Predicted **average widths**, which show minimal variance across methods, validating consistency.

This output format captures the **overall length and structure of trabecular bone shapes** in a measurable form. It enables further analysis such as morphological classification, disease detection, or biomechanical simulation. The consistent layout of annotated images and standardized CSV schemas ensures **easy visualization, traceability, and compatibility** with external statistical and machine learning frameworks.

By decomposing the task into these modules, *Trabecular* achieves a scalable and reproducible workflow capable of processing individual images or entire datasets with minimal manual intervention. This design not only streamlines bone morphology analysis but also lays the foundation for future extensions such as 3D analysis or machine learning integration.

B. Detailed Analysis of a Single Image

Figure 19 shows the raw trabecular bone Micro-CT image selected for detailed analysis.

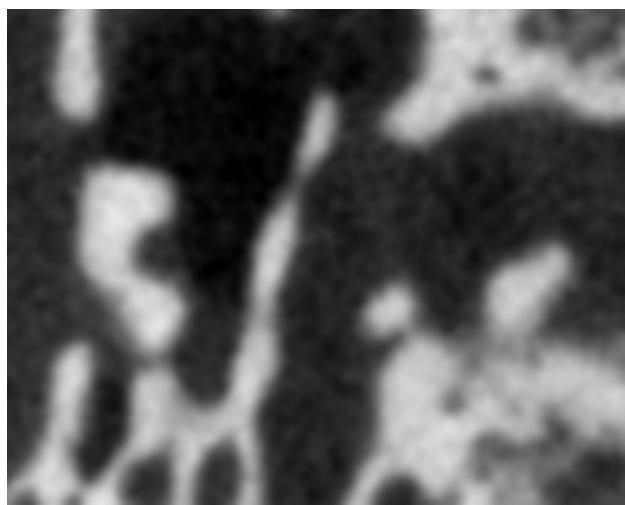


Figure 24 illustrates the key intermediate and final results obtained by *Trabecular bone* for this image:

- **Preprocessed Binary Mask:** The bone structures are segmented into a clean binary format.
- **Skeleton:** One-pixel wide skeleton extracted from the binary mask.

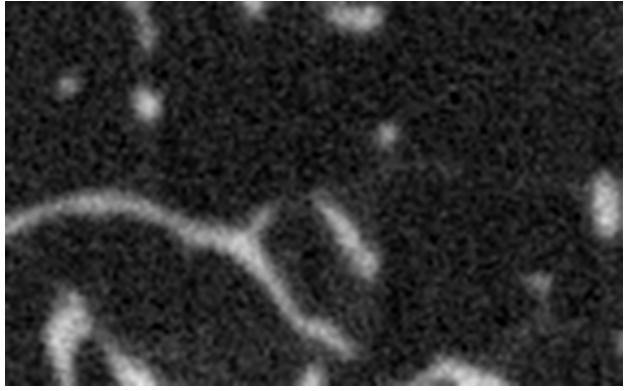
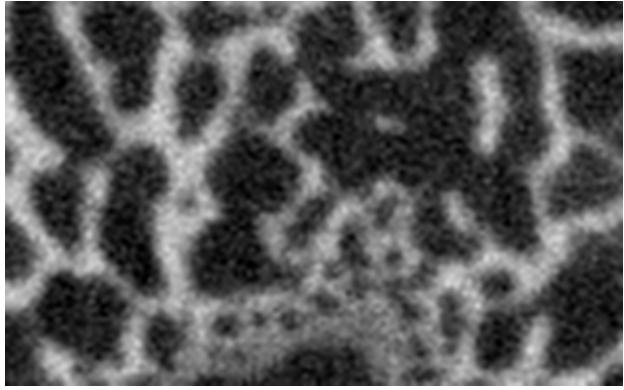


Fig. 19: Raw input image of trabecular bone used for demonstration.

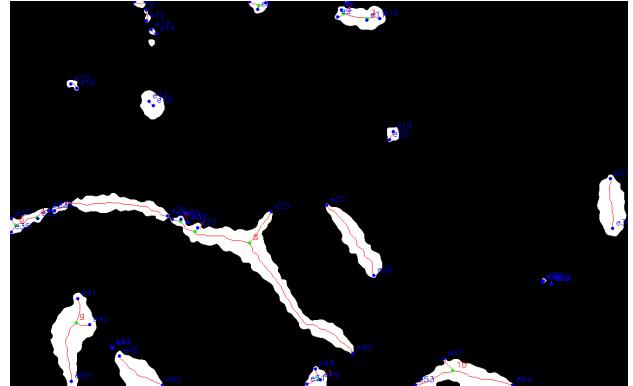


Fig. 20: binary mask



This structure introduces the full pipeline clearly, presents detailed stepwise analysis for one image, and then shows how it generalizes to multiple images with metrics and visual comparisons.

Would you like me to generate any specific LaTeX code or help adding plots or CSV tables?



VI. ERROR ANALYSIS AND LIMITATIONS

While the *Tabulator* system performs well across typical trabecular bone images, several challenges and edge cases introduce errors during the image analysis pipeline. This section categorizes the primary sources of error and outlines the current limitations of the approach.

A. Ellipse Fitting Mismatches

Ellipse fitting plays a vital role in modeling the elongated geometry of trabecular bone segments. While ellipses offer

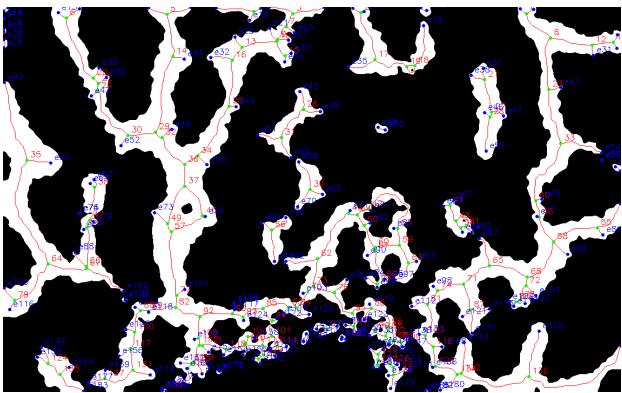
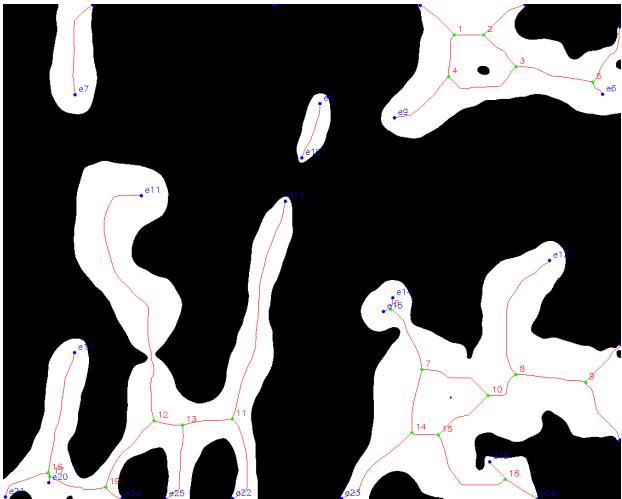


Fig. 21: skeleton

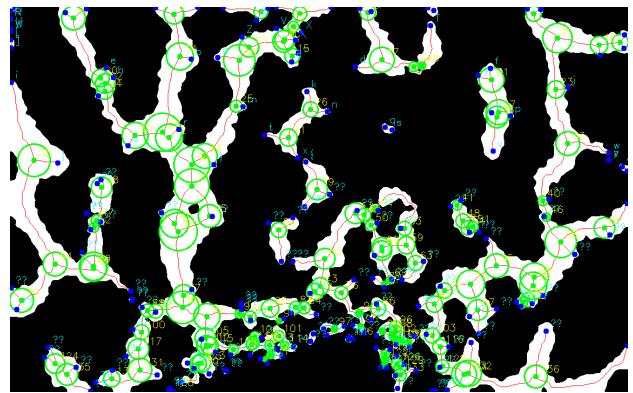
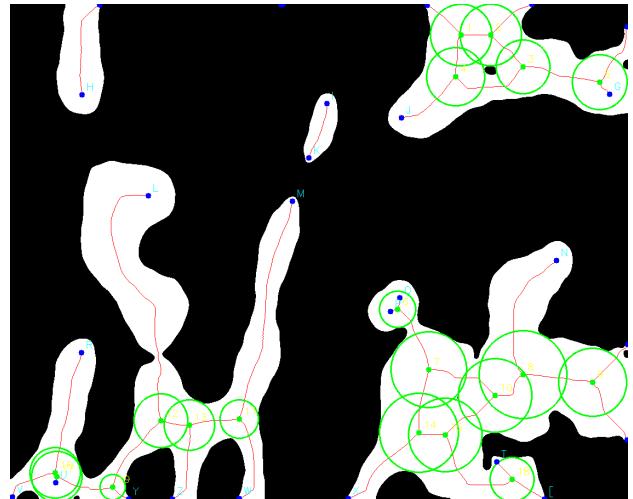
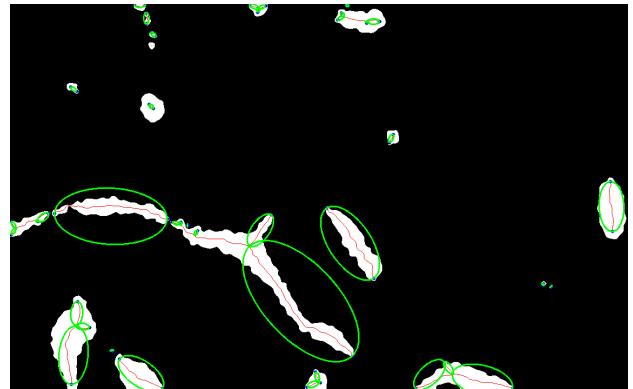
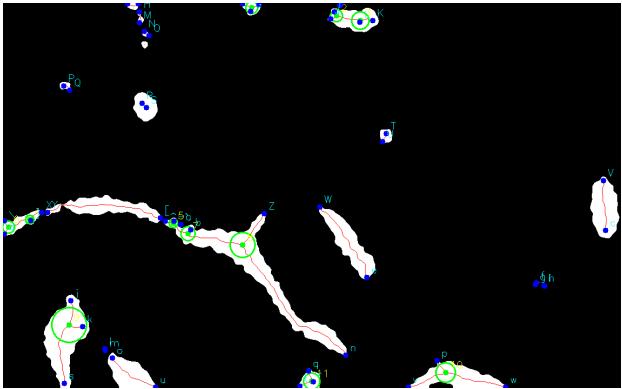


Fig. 22: circle



better adaptability than circles—especially for capturing directional spread—they are also more prone to instability under certain structural conditions.

Common mismatch cases include:

- **Too short or small segments:** Skeleton branches with fewer than 5 points do not provide sufficient geometric variation to define a reliable ellipse. This results in distorted shapes with extreme axis ratios or invalid fits.
- **Highly curved or non-convex segments:** Ellipse models assume smooth curvature. Segments with sharp bends,

loops, or irregular twists often deviate significantly from this assumption, resulting in misaligned or skewed ellipses.

- **Intersecting or bifurcating paths:** When nearby branches overlap or split too close together, the fitting algorithm may unintentionally include multiple structures, leading to ellipses that cross boundaries or misrepresent the actual shape.

Such cases introduce spikes in fitting error metrics and reduce the reliability of geometric analysis. To mitigate these

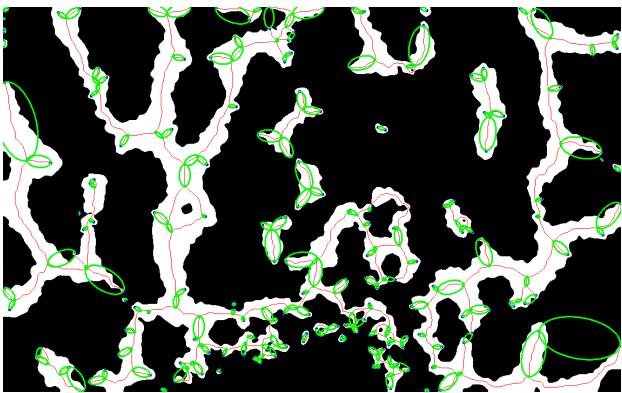
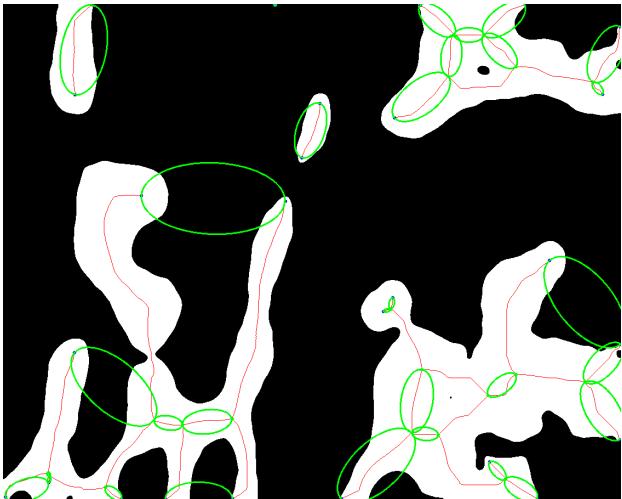


Fig. 23: ellipse fitting

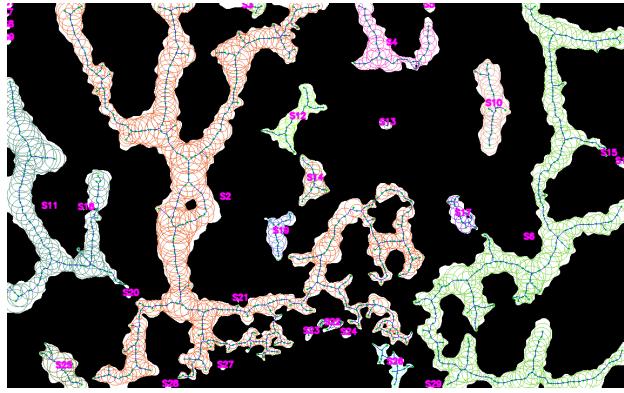
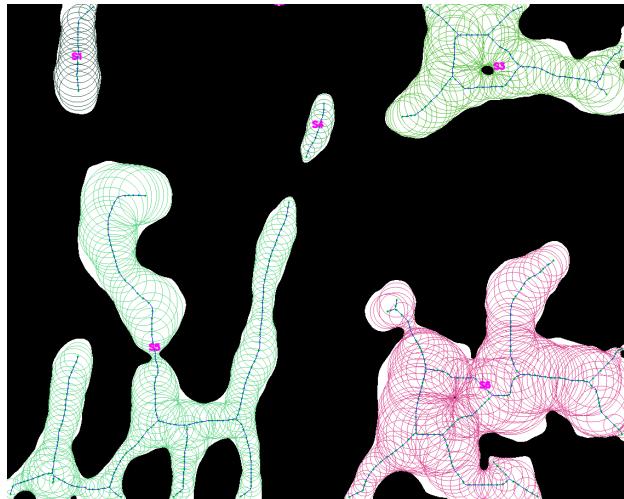
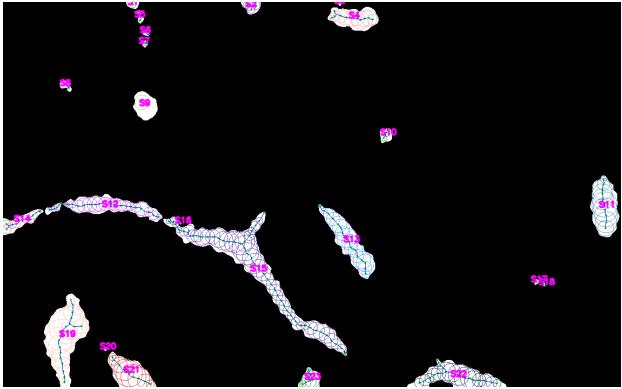


Fig. 24: circle fitting



issues, the following strategies were adopted in the pipeline:

- **Minimum point threshold:** Branches below a predefined length or pixel count are skipped or assigned alternate models (e.g., circles).
- **Post-fit filtering:** Ellipses are evaluated based on their eccentricity, angle variance, and residual fitting error. Outliers are flagged and excluded.
- **Fallback fitting:** When ellipse fitting fails validation, the pipeline automatically reverts to a simpler circle model for robustness.

These safeguards ensure that the geometric fitting module remains stable and meaningful, even under structurally complex or noisy input conditions.

B. Edge-case Failures

While the *Trabecular bone* pipeline performs robustly on well-prepared images, certain edge-case scenarios can break or significantly degrade the analysis pipeline. These cases usually arise from abnormal skeleton structures or poor input quality.

- **Overconnected skeletons:** When trabecular segments appear fused, overly dense, or overlapping in the binary mask, the skeletonization step produces large interconnected regions. This results in an excessive number of nodes and edges, causing an ambiguous topology and incorrect geodesic paths.
- **Discontinuous segments:** Low-contrast or fragmented inputs can lead to skeletons with missing pixels or breaks. These discontinuities create isolated skeleton fragments, which fail during graph traversal, path computation, and geometric fitting.
- **Highly noisy images:** Unfiltered background noise or speckle artifacts may be interpreted as legitimate structures, introducing spurious skeleton branches. These ir-

relevant branches distort both topological graphs and the accuracy of shape fitting.

To handle such cases, additional preprocessing steps (e.g., morphological filtering, region-of-interest isolation) and post-validation logic can be integrated. However, complete resolution often depends on image quality and source imaging conditions.

C. Sensitivity to Image Quality

The performance of the *Trabecular bone* pipeline heavily depends on the quality of the input Micro-CT images. Although the system is designed to handle minor artifacts and intensity variations, significant degradation in image quality can severely affect all downstream processes, from thresholding to geometric analysis.

Key sensitivity factors include:

- **Low-contrast Micro-CT Images:** Poor contrast makes it difficult to differentiate bone structures from the background during binarization, often resulting in incomplete or incorrect segmentation.
- **Uneven illumination and artifacts:** Vignetting, scanning noise, or medical annotations (e.g., labels, timestamps) can interfere with local intensity thresholds, causing artifacts in the skeletonized output.
- **Pixelation and low resolution:** When images are heavily compressed or captured at low resolution, fine structural details like thin branches, endpoints, or curvatures are lost. This reduces contour smoothness and affects both ellipse and circle fitting accuracy.

These quality-related challenges tend to cascade through the pipeline, degrading the reliability of skeleton generation, node identification, topological mapping, and geometric shape modeling. In such cases, optional enhancements like **contrast stretching**, **adaptive thresholding**, or **denoising filters** may improve stability.

D. Limitations of the Current Approach

Although the *Trabecular* framework performs robustly in many standard cases, several inherent limitations restrict its broader applicability and scalability:

- **2D Image Restriction:** The pipeline currently operates on single 2D Micro-CT images. Real-world medical datasets often involve volumetric scans (e.g., CT or MRI), which would require a complete overhaul of the processing stages, including 3D skeletonization, node tracking in 3D space, and surface-based geometric fitting.
- **Assumption of Thin Structures:** Skeletonization assumes that input structures are thin and well-separated. In cases of thick trabecular regions, overlapping bones, or fused areas, this assumption breaks down, leading to distorted medial axes and erroneous topological interpretation.
- **Hard-coded Threshold Parameters:** Several modules rely on fixed thresholds (e.g., minimum contour area,

branch length for ellipse fitting). These may work well for one dataset but often require manual tuning when applied to new imaging setups or magnification levels.

- **Lack of Automatic Fit Validation:** The system currently lacks automated post-processing checks for ellipse or circle fit quality. As a result, human intervention is sometimes required to discard poor fits or re-run the analysis with adjusted parameters.
- **No Adaptive or Learning Mechanisms:** The entire framework is rule-based and does not incorporate any form of adaptive learning. This makes it less resilient to domain shift, image noise variations, or novel morphological patterns not seen during development.

These limitations highlight potential directions for future work, including the incorporation of learning-based segmentation, dynamic parameter tuning, and extension to 3D skeleton analysis.

E. Summary

The errors and limitations described above are typical of morphological analysis systems that rely on deterministic skeletonization and geometric fitting. While the *Trabecular* framework demonstrates robust performance on standard input images, certain failure modes and accuracy bottlenecks highlight the need for further refinement.

Future improvements could significantly enhance system generalizability and reliability by:

- **Incorporating adaptive parameter tuning:** Dynamically adjusting thresholds and fit criteria based on image statistics or learned heuristics.
- **Leveraging deep-learning-based segmentation:** Replacing traditional thresholding with neural network-driven preprocessing pipelines to improve binary mask quality.
- **Extending to 3D data:** Applying the pipeline to volumetric imaging modalities such as CT or MRI for comprehensive anatomical analysis.
- **Post-fit geometric validation:** Introducing optimization loops or filters to discard, correct, or re-fit invalid ellipse or circle models.

These extensions would allow *Trabecular* to scale to larger and more diverse datasets, minimize manual intervention, and improve its applicability in clinical and research environments.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

This project presented **Trabecular bone**, end-to-end image processing pipeline designed for analyzing the topology and micro-architecture of trabecular bones using 2D radiographic images. Beginning with raw and often noisy ‘.tiff’ Micro-CT inputs, the system applies a structured sequence of steps, including image enhancement, skeletonization, graph-based

topology construction, and geometric modeling through circle and ellipse fitting.

The framework extracts a wide range of structural and morphological features such as:

- Overall skeleton structure and total tabular length,
- Density and distribution of nodes and endpoints,
- Segment-wise geometric parameters from circle and ellipse fitting,
- Fitting error percentages and geodesic connectivity metrics.

By building upon powerful Python libraries such as OpenCV, scikit-image, SciPy, and pandas, the pipeline maintains a modular, reusable architecture that is both interpretable and extensible. It supports batch automation for large datasets and produces well-organized outputs, including annotated images and CSV summaries.

Overall, complex bone images are converted into meaningful, quantifiable data suitable for medical analysis, morphological research, or machine learning applications.

B. Research Insights

Extensive experimentation and qualitative analysis during the development of the *Trabecular bone* pipeline led to several important research insights:

- **Ellipse fitting outperforms circle fitting** for modeling elongated and anisotropic bone segments. It captures structural orientation and aspect ratios more accurately, especially in irregular regions.
- **Skeleton-based topology extraction** proves effective for reducing complex structures into analyzable graphs. This abstraction allows measurement of node density, branch lengths, and enables connectivity modeling relevant to biomechanical behavior.
- **Input image quality significantly impacts performance.** Low-contrast, noisy, or improperly thresholded images impair skeletonization and result in unreliable geometric models. Preprocessing thus plays a critical role in pipeline success.
- **Geodesic distance metrics offer anatomically meaningful paths.** Compared to simple Euclidean metrics, geodesic routing better reflects true structural pathways through the trabecular network, making it suitable for biological and physical interpretations.

C. Future Work

While the current 2D implementation of the *Trabecular bone* pipeline performs reliably on standardized Micro-CT inputs, several promising directions exist to enhance its functionality, scalability, and clinical relevance:

3D Structural Analysis:

- Extend shape modeling and skeletonization techniques to volumetric data from CT or MRI scans.
- Apply 3D medial axis extraction and voxel-based graph construction for capturing full bone morphology.

- Develop visualization tools for analyzing surface curvature, orientation, and internal connectivity in 3D space.

Machine Learning-Based Shape Modeling:

- Train supervised models to classify skeletal segments into categories such as linear, curved, or branching.
- Employ convolutional neural networks (CNNs) or attention-based models to perform shape fitting in noisy or ambiguous regions.
- Use graph neural networks (GNNs) to capture topological patterns indicative of pathological changes or degeneration.

Clinical and Diagnostic Integration:

- Support export to medical standards such as DICOM and HL7 for compatibility with hospital EHR systems.
- Combine extracted morphological features with clinical indicators (e.g., BMD, T-scores) for osteoporosis risk analysis and research.

Closing Remark

The **Trabecular bone** pipeline establishes a robust foundation for automated, topology-aware analysis of trabecular bone structures from radiographic images. By integrating graph-based modeling, geometric shape fitting, and detailed visual and tabular outputs, the framework serves as a valuable tool for both researchers and clinical practitioners.

With future enhancements such as 3D volumetric support and AI-driven analysis *Trabecular* holds significant potential to advance diagnostic accuracy, support personalized treatment planning, and contribute to morphometric studies in bone health and disease.

REFERENCES

- [1] P. K. Zysset *et al.*, "Clinical use of quantitative computed tomography-based finite element analysis of the hip and spine in the management of osteoporosis in adults," *J. Bone Miner. Res.*, vol. 28, no. 6, pp. 1186–1204, 2013.
- [2] S. Boutroy *et al.*, "In vivo assessment of trabecular bone microarchitecture by high-resolution peripheral quantitative computed tomography," *J. Clin. Endocrinol. Metab.*, vol. 90, no. 12, pp. 6508–6515, 2005.
- [3] J. A. Kanis *et al.*, "European guidance for the diagnosis and management of osteoporosis in postmenopausal women," *Osteoporos. Int.*, vol. 30, no. 1, pp. 3–44, 2019.
- [4] A. J. Burghardt *et al.*, "High-resolution peripheral quantitative computed tomographic imaging of cortical and trabecular bone microarchitecture in patients with type 2 diabetes mellitus," *J. Clin. Endocrinol. Metab.*, vol. 95, no. 11, pp. 5045–5055, 2010.
- [5] E. M. Stein *et al.*, "MRI-based texture analysis of trabecular bone for opportunistic screening of skeletal fragility," *J. Clin. Endocrinol. Metab.*, vol. 106, no. 8, pp. e3000–e3010, 2021.
- [6] Q. Cao *et al.*, "Assessment of trabecular bone stiffness using radiomics and deep-learning features," *Med. Phys.*, vol. 48, no. 7, pp. 3643–3654, 2021.
- [7] T. M. Deserno, "Fundamentals of biomedical image processing," in *Biomedical Image Processing*, Berlin: Springer, 2011, pp. 1–51.
- [8] J. Y. Rho *et al.*, "Mechanical properties and the hierarchical structure of bone," *Med. Eng. Phys.*, vol. 20, no. 2, pp. 92–102, 1998.
- [9] R. Müller *et al.*, "Morphometric analysis of human bone biopsies: a quantitative structural comparison of histological sections and micro-computed tomography," *Bone*, vol. 23, no. 1, pp. 59–66, 1998.

- [10] M. J. Menten *et al.*, "A skeletonization algorithm for gradient-based optimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 8271–8280.
- [11] G. M. Blake *et al.*, "Role of dual-energy X-ray absorptiometry in the diagnosis and treatment of osteoporosis," *J. Clin. Densitom.*, vol. 10, no. 1, pp. 102–110, 2007.
- [12] Y. Kimori, "Mathematical morphology-based approach to the enhancement of morphological features in medical images," *J. Clin. Bioinform.*, vol. 1, no. 1, pp. 1–14, 2011.
- [13] P. Zhang *et al.*, "Prediction of trabecular bone architectural features by deep learning models using simulated DXA images," *Bone*, vol. 136, Art. no. 115348, 2020.
- [14] J. Chen *et al.*, "Deep learning-based high-resolution reconstruction of trabecular bone from clinical CT images," *Med. Image Anal.*, vol. 62, Art. no. 101659, 2020.
- [15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
- [16] K. B. Girum *et al.*, "Fast interactive medical image segmentation with weakly supervised deep learning method," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 9, pp. 1437–1444, 2020.
- [17] B. Sridhar, "Investigations of medical image segmentation methods with inclusion of mathematical morphological operations," *Tech. Sci.*, vol. 38, no. 5, pp. 511–520, 2021.
- [18] E. Selvi *et al.*, "Performance analysis of distance transform-based inter-slice similarity information on segmentation of medical image series," *Math. Comput. Appl.*, vol. 18, no. 3, pp. 511–520, 2013.
- [19] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies—a comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, 1992.
- [20] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. Cambridge, MA: MIT Press, 1967, pp. 362–380.
- [21] M. J. Menten *et al.*, "A skeletonization algorithm for gradient-based optimization," arXiv:2309.02527, 2023. [Online]. Available: <https://arxiv.org/abs/2309.02527>
- [22] G. Borgefors, "Distance transformations in digital images," *Comput. Vision Graph. Image Process.*, vol. 34, no. 3, pp. 344–371, 1986.
- [23] F. Paternostro *et al.*, "An application of the graph theory to the study of the human locomotor system," *Ital. J. Anat. Embryol.*, vol. 124, no. 3, pp. 353–363, 2019.
- [24] D. Macrini *et al.*, "Object categorization using bone graphs," *Comput. Vis. Image Underst.*, vol. 115, no. 8, pp. 1187–1206, 2011.
- [25] S. Khandelwal *et al.*, "ECM-aware cell-graph mining for bone tissue modeling and classification," *Data Min. Knowl. Discov.*, vol. 20, no. 3, pp. 416–438, 2010.
- [26] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least squares fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, 1999.
- [27] J. Yu, M. Lee, and J. Kim, "Robust fitting of ellipses and spheroids," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3544–3553, 2010.
- [28] P. L. Rosin, "A note on the least squares fitting of ellipses," *Pattern Recognit. Lett.*, vol. 14, no. 10, pp. 799–808, 1993.
- [29] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Software Tools*, vol. 25, no. 11, pp. 120–125, 2000.
- [30] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [31] A. Al-Sharadqah and G. Piga, "Fitting concentric elliptical shapes under general model," arXiv:2308.04970, 2023. [Online]. Available: <https://arxiv.org/abs/2308.04970>