2D Span Auto Software Manual

Technical Documentation Team

February 25, 2025

Contents

1 Introduction

2D Span Auto is an advanced image analysis software specifically designed for the automated detection and analysis of dendritic spines in neuronal images. This manual comprehensively guides the software's features, modules, and operational procedures. It is intended to assist users of all experience levels, from beginners to advanced researchers, in effectively utilizing the software for accurate and efficient spine analysis. The manual is structured to provide step-by-step instructions, ensuring that even users with no prior experience can navigate the software with ease.

2 Graphical User Interface Overview

2.1 Main Interface

Upon launching 2D Span Auto, users are greeted with an intuitive and user-friendly graphical interface. The main workspace is centred around a primary viewing area where images are displayed and analyzed. The interface is designed to provide quick access to essential tools while maintaining a clean and organized workspace. Key components of the interface include tool panels, visualization areas, and menu options, all of which are clearly labelled for ease of use.

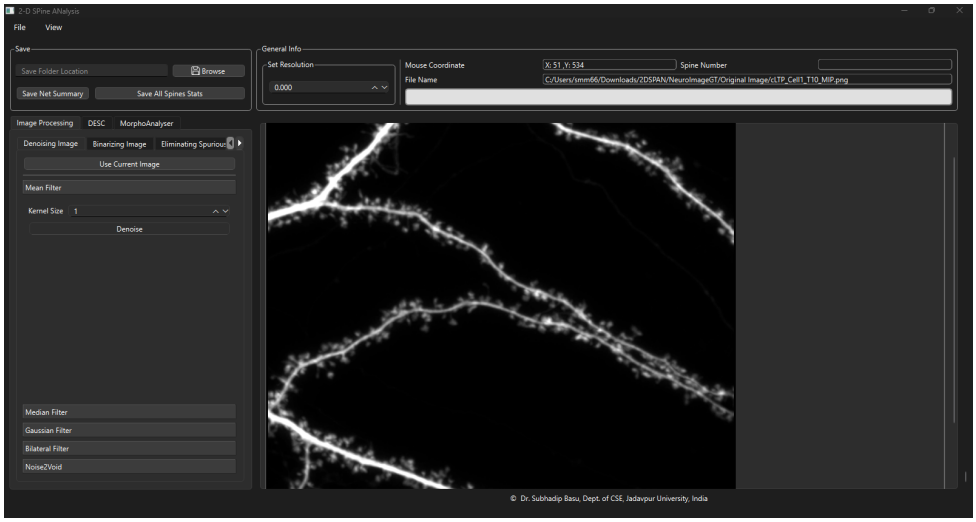

Figure 1: Main software interface displaying the primary viewing area after image loading. The interface provides a comprehensive view of the analysis workspace, with clearly organized tool panels and visualization areas. This layout facilitates efficient access to all major software functionalities while maintaining a clear view of the image being analyzed.

2.2 File Menu Operations

The File Menu is the starting point for all image analysis workflows. It provides essential operations for loading, saving, and managing images. Users can load individual images using the "Load Image" option, which supports common formats such as JPEG and PNG. For batch processing, the "Load Image Folder" functionality allows users to queue multiple images for sequential analysis.

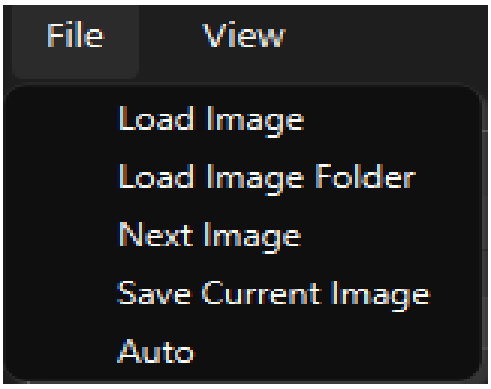

Figure 2: File menu interface showing available options including Load Image, Load Image Folder, Next Image, Save Current Image, and Auto processing features. This menu provides comprehensive file management capabilities, enabling both single-image and batch-processing workflows.

The navigation system within the File Menu offers two distinct workflows:

- **Next Image**: When working with multiple images, users must first click "Next Image" to access the initial image in a loaded folder. This allows for individual control over each image's processing.
- **Auto**: For uniform analysis across all images, users can use the "Auto" button to apply the same parameters to the entire batch.

2.3 General Information Panel

The General Information panel provides real-time feedback during the analysis session. It displays crucial information such as the active file name, live mouse coordinates, and spine identification numbers when hovering over analyzed regions. This panel is essential for monitoring the progress of the analysis and ensuring accurate results.

Figure 3: General Information panel displaying current file details, mouse coordinates, and spine identification data. Users can monitor real-time analysis parameters and set image resolution specifications through this interface. The panel provides immediate feedback on analysis operations and maintains a clear overview of current processing parameters.

2.4 Save Analysis System

The save functionality in 2D Span Auto is designed to preserve the complete state of the current analysis. When activated, this feature captures the exact state of the viewing area, including any active overlays, annotations, or analysis results. This ensures that users can save their work at any point and return to it later without losing any progress.

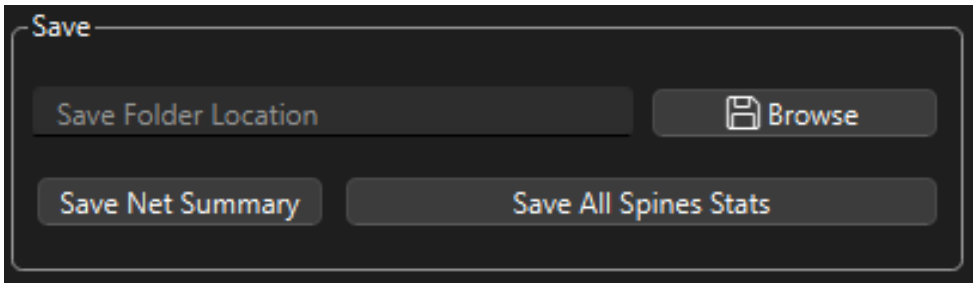

Figure 4: Save Analysis interface providing options for preserving the current analysis state. The panel enables users to capture and store the complete analysis setup, including all overlays and annotations present in the viewing area.

2.5 View Menu Capabilities

The View Menu offers essential visualization controls that enhance the analysis process. Users can toggle between viewing the original grayscale image and its processed binary counterpart. This flexibility allows users to compare the original image with the processed results, ensuring accurate analysis.

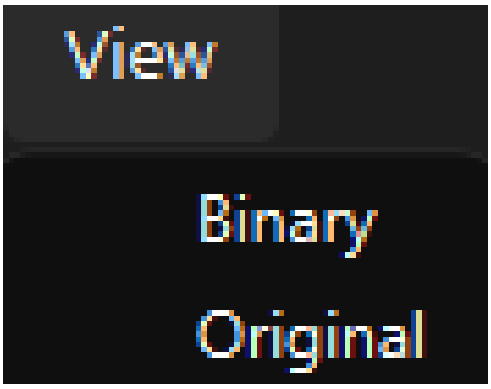

Figure 5: View Menu interface showing visualization options for spine segmentation and classification overlays. Users can toggle between original grayscale and processed binary image displays, providing flexibility in how analysis results are visualized.

3 Image Processing Module

3.1 Image Preprocessing: Denoising

Image preprocessing begins with the crucial step of denoising, which is highly recommended for optimal analysis results. The software offers multiple denoising methods, ranging from traditional filtering techniques to advanced deep learning-based solutions. Each method is designed to reduce noise while preserving important image features.

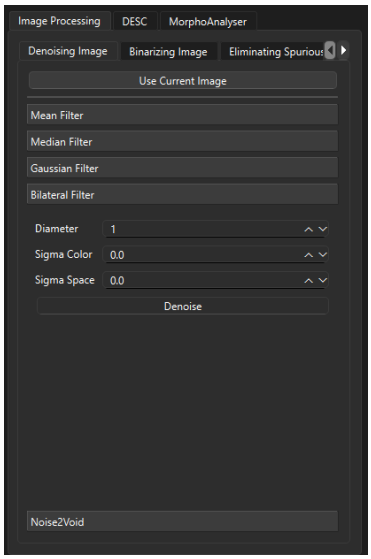

Figure 6: Bilateral Filter interface showing adjustable parameters for diameter, sigma colour, and sigma colour space. This advanced filtering method preserves edges while reducing noise in smooth regions.

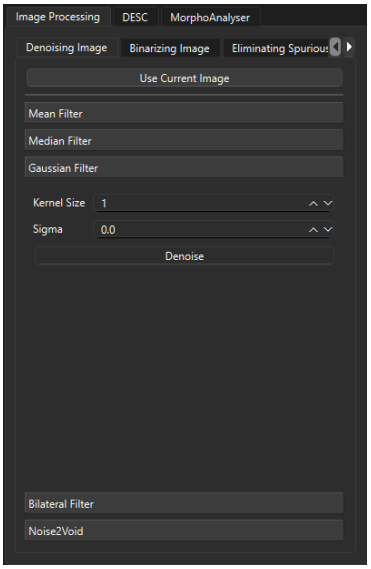

Figure 7: Gaussian Filter interface displaying kernel size and sigma parameter controls. This filtering method provides uniform smoothing across the image.

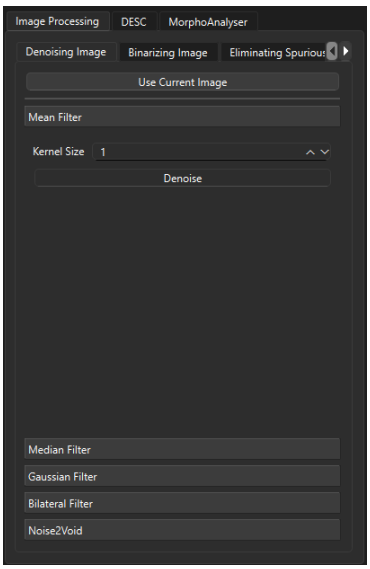

Figure 8: Mean Filter interface with adjustable kernel size parameter. This basic filtering approach provides straightforward noise reduction through local averaging.

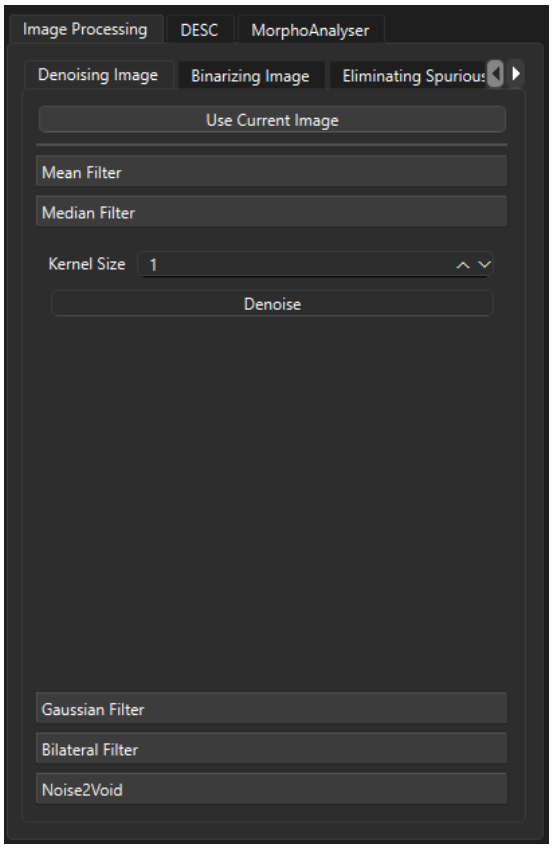

Figure 9: Median Filter interface showing kernel size adjustment. This non-linear filtering method is particularly effective at removing salt-and-pepper noise while preserving edges.

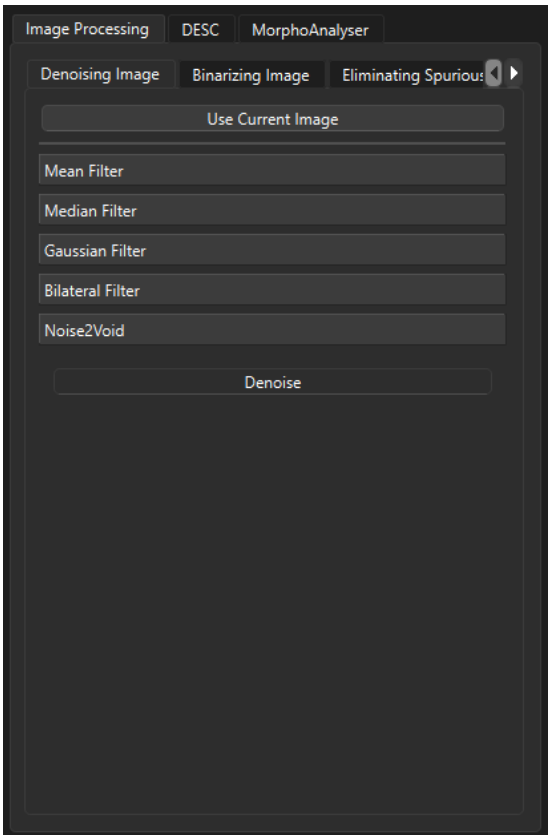

Figure 10: Noise2Void (N2V) interface providing deep learning-based denoising capabilities. This advanced method requires minimal user input while delivering superior noise reduction results.

3.2 Image Binarization

The binarization module converts grayscale images to binary format, a crucial step for spine detection. The software offers multiple binarization methods, each suited to different image characteristics and analysis requirements.

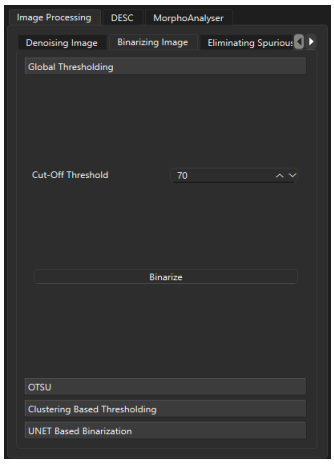

Figure 11: Global Threshold interface allowing manual threshold value selection. This method provides direct control over the binarization process with immediate visual feedback.

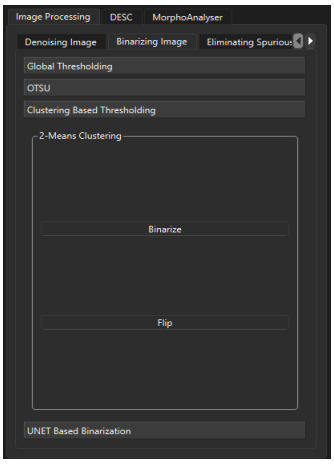

Figure 12: K-means Clustering interface with flip option for binary output inversion. This automated method provides sophisticated image segmentation through statistical clustering.

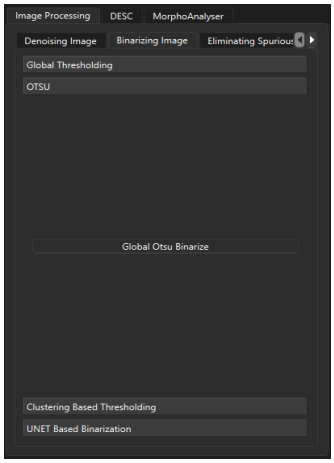

Figure 13: Otsu's Method interface implementing automatic threshold computation. This method excels at finding optimal threshold values for bimodal image distributions.

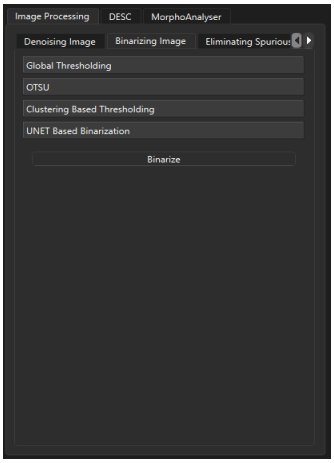

Figure 14: U-Net interface providing deep learning-based image segmentation. This advanced method leverages neural networks for complex feature detection and segmentation.

3.3 Post-Binarization Artifact Removal

The final preprocessing stage involves the removal of spurious artifacts that may persist after denoising and binarization. This module uses threshold-based techniques to clean up the binary image, ensuring that only relevant features are retained for analysis.

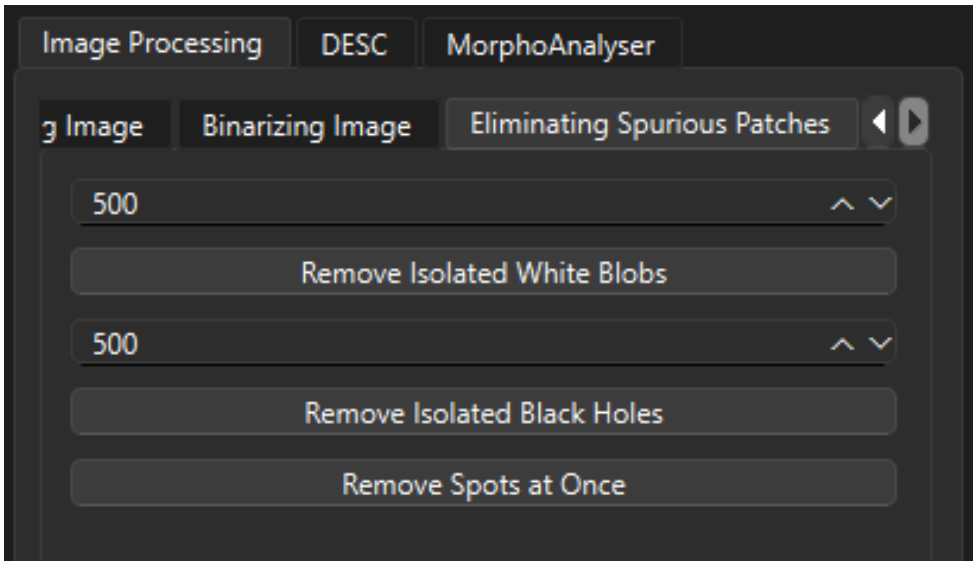

Figure 15: Artifact Removal interface showing threshold controls for hole filling and severed neck correction. These tools enable fine-tuned cleanup of binarization artifacts while preserving important image features.

Best Practices and Recommendations For optimal results, follow these steps:

1. Begin with denoising to reduce noise in the image.
2. Choose the appropriate binarization method based on image characteristics.
3. Use artifact removal to clean up the binary image, adjusting parameters conservatively to avoid removing valid features.

4 Dendrite Extraction and Spine Compartmentalization (DESC) Module

The DESC module is divided into three main parts:

1. Preprocessing
2. Skeletonization
3. Spine Localization

4.1 Distance Transform and Collision Impact Computation

The Preprocessing module identifies strong points for skeletonization using Blum's Grassfire algorithm. Users can adjust the strength threshold of detected points to optimize skeletonization results.

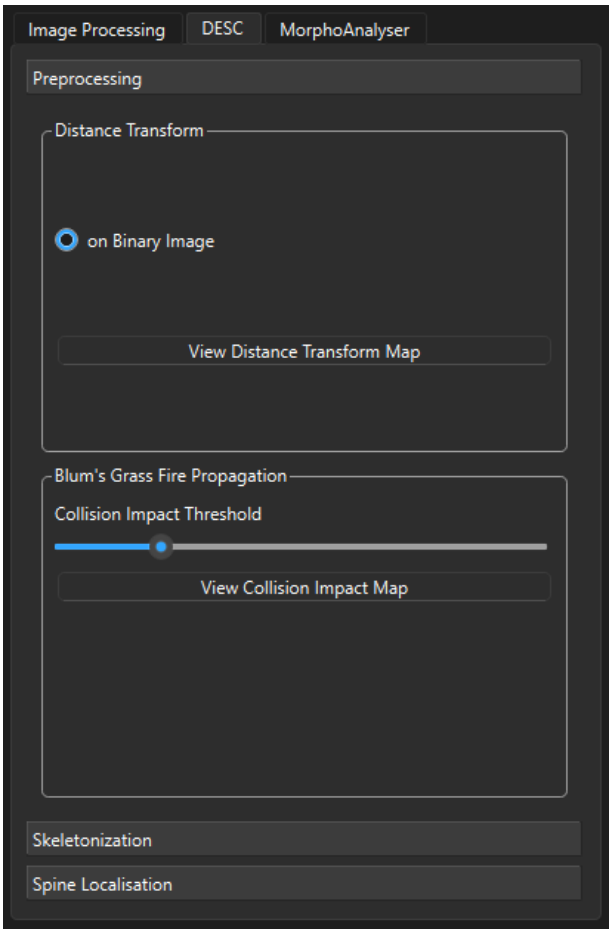

Figure 16: Visualization of the Distance Transform and Collision Impact computation. The displayed matrix represents the strength of points identified by Blum's Grassfire algorithm, which are crucial for the skeletonization process.

Operational Procedure

1. Click the "View Collision Impact" button to generate the initial matrix of strong points.
2. Adjust the strength threshold using the provided slider to filter out weaker points if necessary.
3. Higher threshold values retain only the most prominent strong points, reducing noise in the skeletonization process.

4.2 Skeletonization

The Skeletonization module offers two pathways for generating one-pixel-wide skeletal representations:

- **Expert Pathway**: Users can evaluate skeletal structures and adjust collision impact threshold values for optimal results.
- **Standard Pathway**: Provides direct access to the final skeleton for quick analysis.

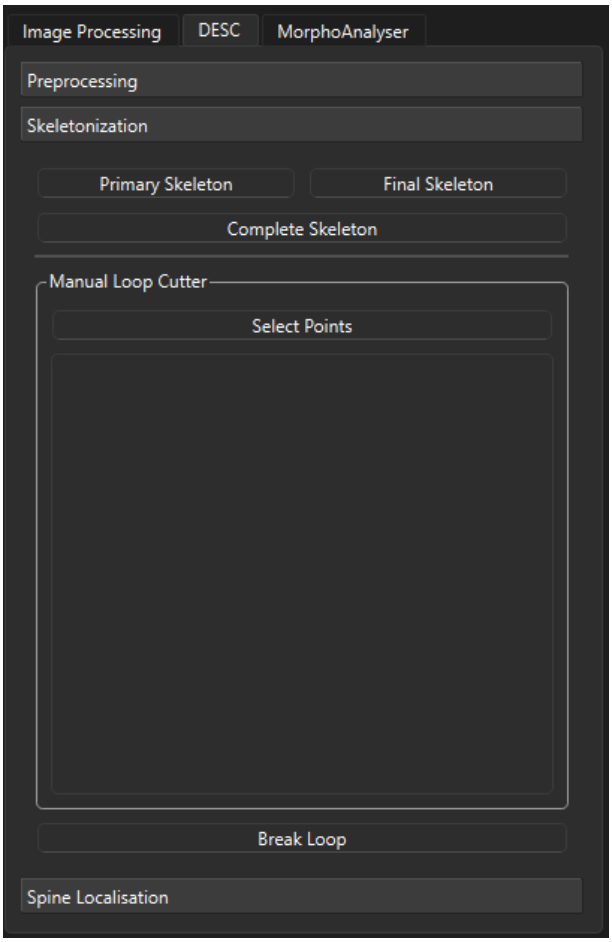

Figure 17: Skeletonization interface showing the Primary Skeleton, Final Skeleton, and Complete Skeleton options, along with the Manual Loop Cutter tool for precise skeleton refinement.

4.3 Spine Localization

This section is divided into two steps:

1. Dendrite Extraction
2. Spine Compartmentalization

4.3.1 Dendrite Extraction

The Dendrite Extraction module offers both automated and expert-guided approaches for extracting dendrite shafts. Users can adjust parameters such as the Alpha Parameter (α) and the number of iterations to optimize the extraction process.

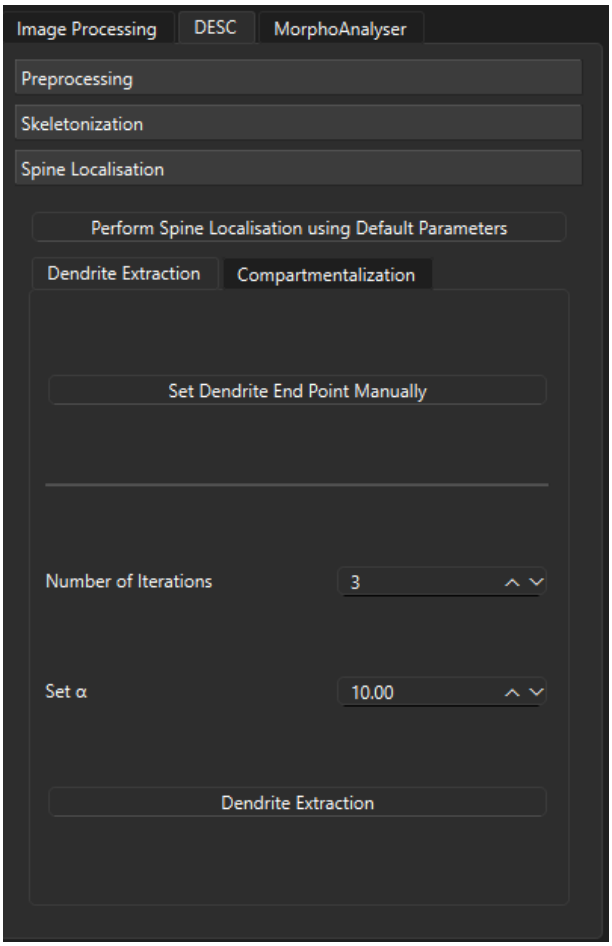

Figure 18: Dendrite Extraction interface displaying the automated processing option, expert parameter controls, and manual endpoint setting functionality.

4.3.2 Spine Compartmentalization

The Spine Compartmentalization module enables detailed segmentation of dendritic spines. Users can adjust parameters for Spine Base Point (SBP), Central Head Point (CHP), and Head Neck Junction (HNJ) detection to achieve accurate results.

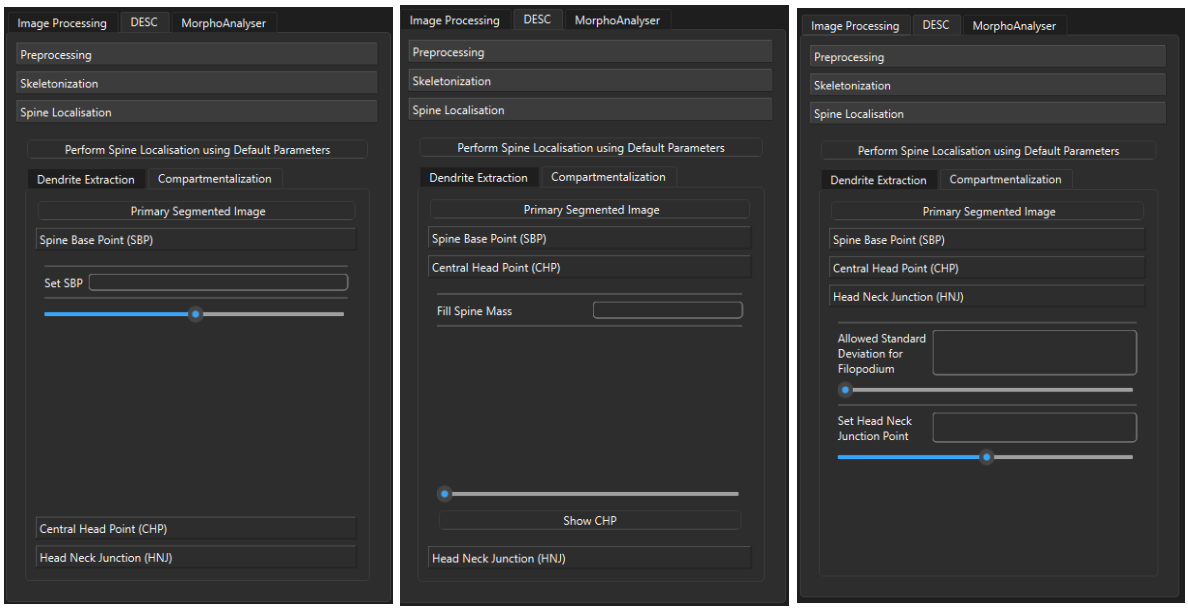

Figure 19: Spine Compartmentalization interface showing the sequential process of spine point detection, including SBP, CHP, and HNJ parameter controls.

5 MorphoAnalyser Module

The MorphoAnalyser module provides comprehensive statistical analysis of detected dendritic spines through two interfaces: Global Statistics and Individual Spine Statistics.

5.1 Global Statistics

The Global Statistics interface presents aggregate data across all detected spines. Users must configure proper resolution settings in the General Information tab before using this feature.

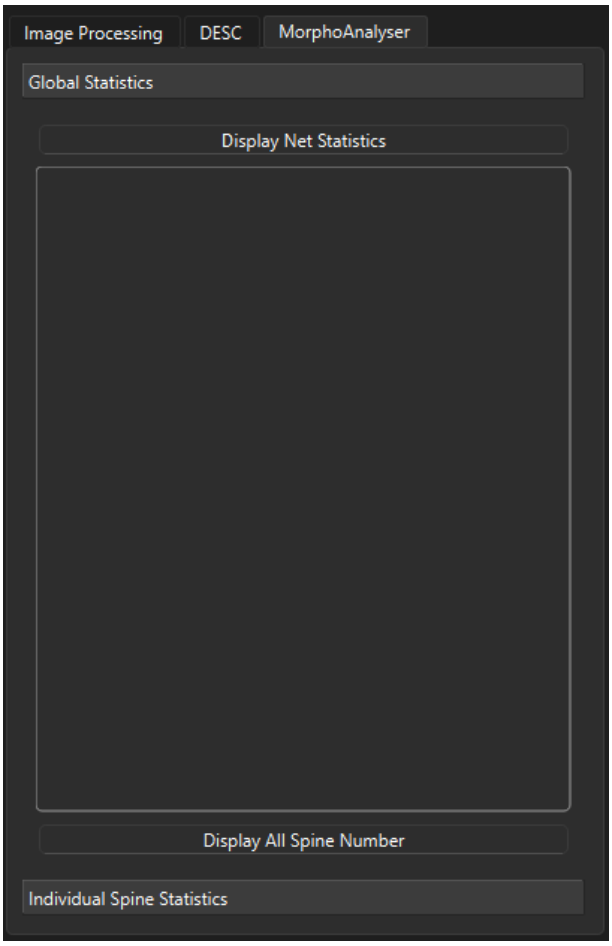

Figure 20: Global Statistics interface displaying the main analytics dashboard with "Display Net Statistics" button centered at the top and a large visualization area below for presenting aggregate spine metrics.

5.2 Individual Spine Statistics

The Individual Spine Statistics interface enables detailed analysis of specific spines. Users can select a spine by entering its number and clicking "Display Spine Information" to generate detailed metrics.

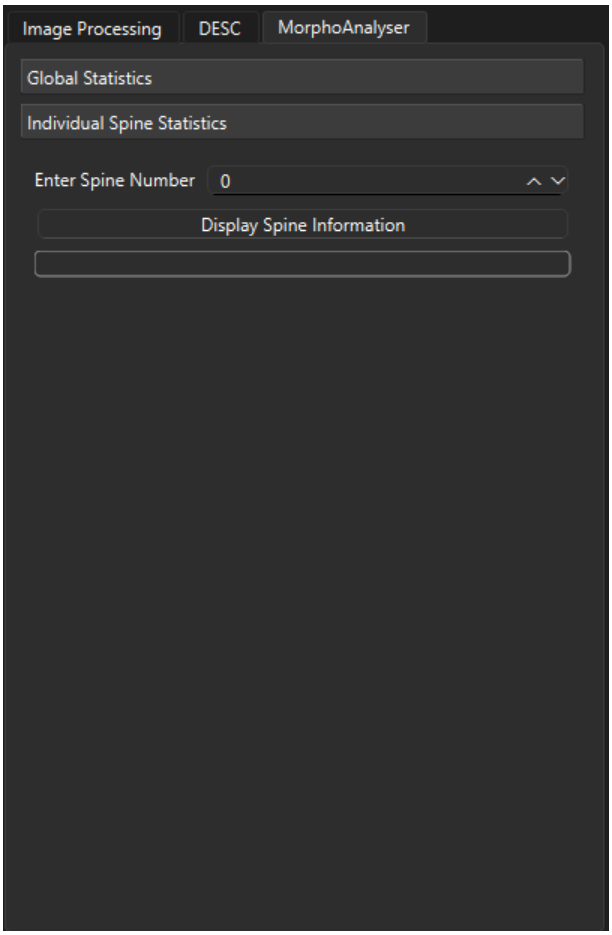

Figure 21: Individual Spine Statistics interface showing the spine selection controls at the top, including the spine number input field with increment/decrement arrows, and the "Display Spine Information" button.

6 Demonstrating the Workflow on an Example Image

This section provides a comprehensive, step-by-step demonstration of the complete 2D Span Auto software workflow using a representative neuronal image. By following these instructions meticulously, users will be able to conduct accurate and efficient dendritic spine analysis regardless of their prior experience with the software.

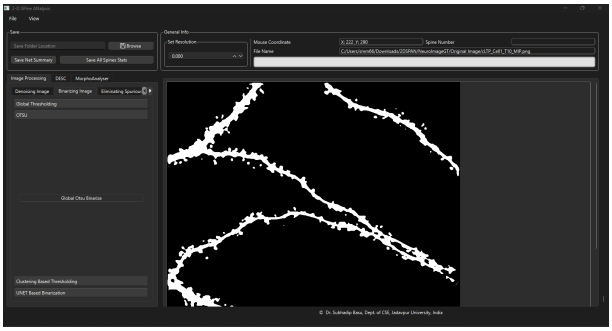

Figure 22: Application of Otsu's thresholding method to the original grayscale image. This automatic binarization technique effectively segments the neuronal structure from the background by computing an optimal threshold value based on the image's intensity histogram.

6.1 Initial Image Processing

6.1.1 Step 1: Image Binarization

Begin by loading your image through the File Menu. For this demonstration, we will apply Otsu's thresholding method directly to the grayscale image:

1. Navigate to the Image Processing Module. 2. Select the "Binarization" tab. 3. Choose "Otsu's Method" from the available binarization options. 4. Click "Apply" to execute the binarization. The result is shown in Figure 22.

Otsu's method automatically calculates an optimal threshold value based on the image histogram, making it suitable for images with bimodal intensity distributions such as fluorescent neuronal images.

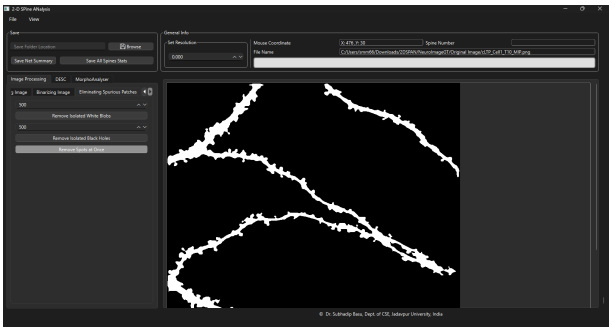

Figure 23: Binary image after artifact removal. Note how small spurious objects and noise have been eliminated while preserving the integrity of the neuronal structures. This cleanup step is crucial for accurate skeletonization and spine detection in subsequent stages.

6.1.2 Step 2: Artifact Removal

After binarization, it is essential to clean the image by removing any spurious artifacts that may interfere with accurate analysis:

1. Navigate to the "Artifact Removal" tab in the Image Processing Module. 2. Adjust the threshold sliders for hole filling and artifact removal. Begin with conservative values (around 30-40) to avoid removing valid features. 3. Click "Apply" to execute the artifact removal. The result is shown in Figure 23.

This step eliminates small noise objects while preserving the integrity of the dendrites and spines, creating a clean binary image for subsequent analysis steps.

6.2 Distance Transform and Skeletonization

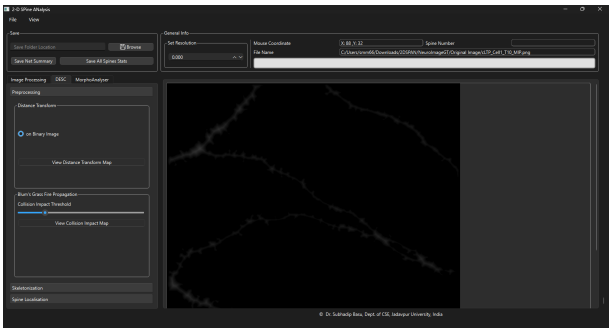

Figure 24: Visualization of the Distance Transform applied to the binary image. The color gradient represents the distance of each foreground pixel to the nearest background pixel, with brighter regions indicating greater distances from the edge of the structure.

6.2.1 Step 3: Distance Transform Computation

The Distance Transform is a crucial preprocessing step for accurate skeletonization:

1. Navigate to the DESC Module and select the "Preprocessing" tab. 2. Click "Generate Distance Transform" to compute the distance of each foreground pixel to the nearest background pixel. 3. The resulting visualization is shown in Figure 24, where brighter areas indicate greater distances from the structure's edge.

The Distance Transform provides essential information about the structure's thickness, which is vital for accurate skeletonization and spine detection.

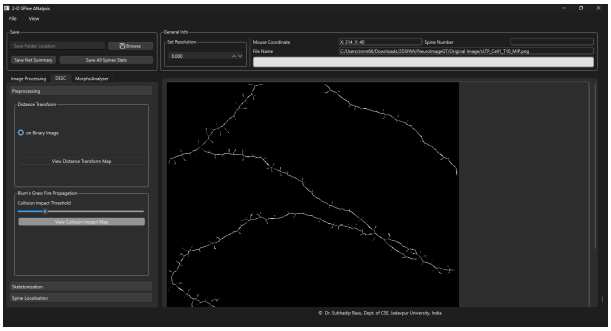

Figure 25: Collision Impact visualization showing strong points identified by Blum's Grassfire algorithm. These points represent potential skeleton locations where fire fronts from opposite edges would meet, forming the basis for accurate skeletonization.

6.2.2 Step 4: Collision Impact Computation

The Collision Impact computation identifies strong points for skeletonization:

1. In the "Preprocessing" tab, click "View Collision Impact" to generate the matrix of strong points. 2. If necessary, adjust the strength threshold using the provided slider to filter out weaker points. 3. The resulting visualization is shown in Figure 25, where brighter points indicate stronger collision impact values.

These strong points serve as the foundation for accurate skeletonization in the next step.

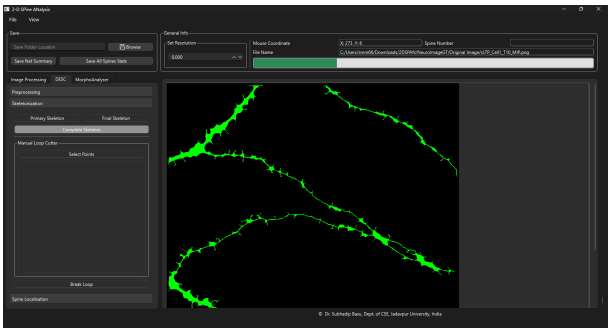

Figure 26: Initial stage of the skeletonization process using the Complete Skeleton method. The software begins by identifying the central axis of the neuronal structures based on the strong points identified in the Collision Impact computation.

6.2.3 Step 5: Skeletonization Process Initiation

Now proceed to the skeletonization process:

1. Navigate to the "Skeletonization" tab in the DESC Module. 2. Select the "Complete Skeleton" option for comprehensive skeletonization. 3. Click "Generate Skeleton" to initiate the process. 4. The initial stage of skeletonization is shown in Figure 26.

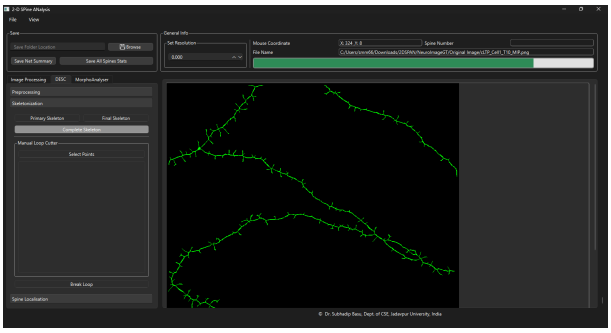

Figure 27: Intermediate stage of the skeletonization process. The software is progressively refining the skeleton by pruning redundant branches and connecting fragmented segments to create a continuous one-pixel-wide representation of the neuronal structure.

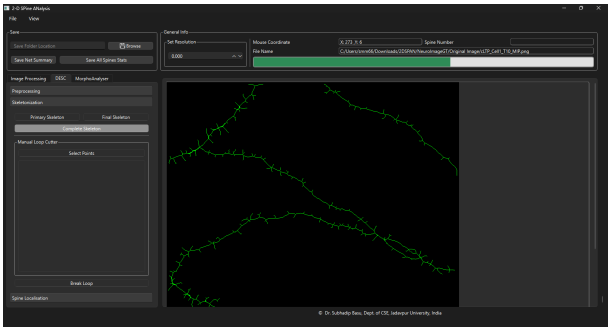

Figure 28: Final stages of the skeletonization process. The algorithm is completing the refinement of the skeletal structure, ensuring accurate representation of the neuronal morphology while maintaining topological integrity.

6.2.4 Step 6: Skeletonization Progression

The skeletonization process progresses through several stages automatically:

1. During this process, the software will display the ongoing refinement of the skeletal structure, as shown in Figures 27 and 28. 2. **IMPORTANT: Do not click any buttons or interact with the software during the skeletonization process, as this may cause the application to hang or crash.** 3. Allow the process to complete naturally, which may take several seconds to minutes depending on image complexity.

The skeletonization algorithm systematically thins the binary structure to a one-pixel-wide representation while preserving its topological characteristics.

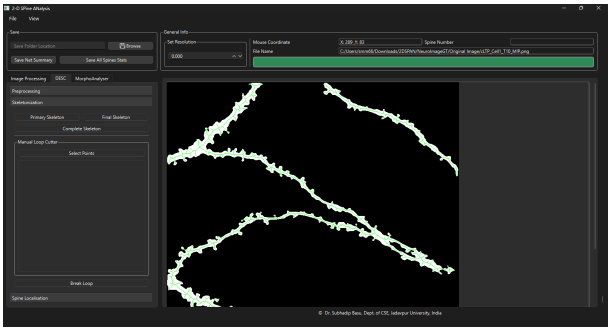

Figure 29: Final skeleton overlay (green lines) on the binarized background. The skeleton accurately represents the central axis of the neuronal structures, providing the foundation for dendrite extraction and spine identification in subsequent steps.

6.2.5 Step 7: Final Skeleton Visualization

Once the skeletonization process is complete:

1. The software will display the final skeleton overlay on the binarized image, as shown in Figure 29. 2. Verify that the skeleton accurately represents the central axis of the dendrites. 3. If the skeleton contains loops or other topological errors, use the "Manual Loop Cutter" tool to correct these issues before proceeding.

A proper skeleton is essential for accurate dendrite extraction and spine identification.

6.3 Dendrite Extraction and Spine Identification

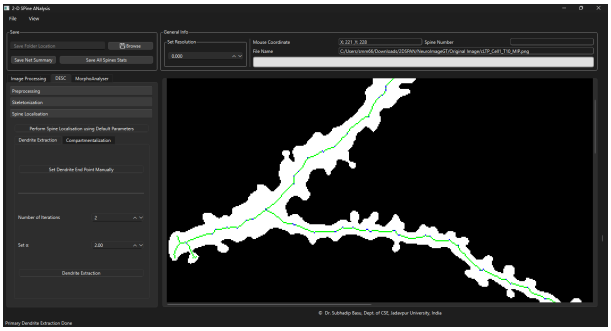

Figure 30: Dendrite extraction process in action. The software is identifying and isolating the dendrite shaft from the surrounding spines based on the skeleton and morphological characteristics of the neuronal structure.

6.3.1 Step 8: Dendrite Extraction Process

Next, proceed to extracting the dendrite shaft from the surrounding spines:

1. Navigate to the "Spine Localization" tab and select the "Dendrite Extraction" sub-tab. 2. Click "Extract Dendrite" to initiate the automated extraction process. 3. The ongoing extraction process is shown in Figure 30. 4. **IMPORTANT: Similar to skeletonization, do not interact with the software during the dendrite extraction process to avoid potential crashes.**

The dendrite extraction algorithm differentiates between the main dendrite shaft and the protruding spines, which is crucial for subsequent spine analysis.

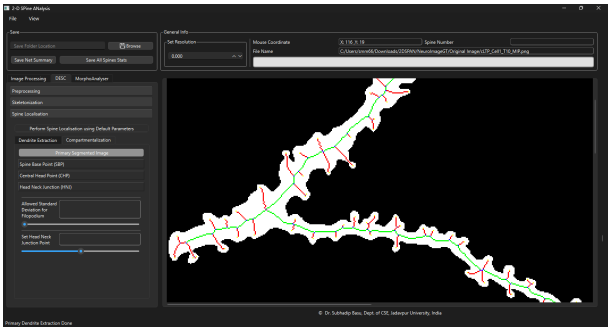

Figure 31: Primary segmentation of dendritic spines. The software has identified individual spines (colored regions) protruding from the dendrite shaft (white), providing the initial foundation for detailed spine analysis.

6.3.2 Step 9: Primary Spine Segmentation

After dendrite extraction completes, the software performs primary segmentation of spines:

1. The result of the primary segmentation is shown in Figure 31, with colored regions representing individual spines. 2. Each spine is now identified as a separate entity, distinct from the dendrite shaft (shown in white). 3. Review the primary segmentation to ensure all visible spines have been detected.

This primary segmentation serves as the foundation for detailed spine compartmentalization and subsequent morphological analysis.

6.4 Spine Compartmentalization

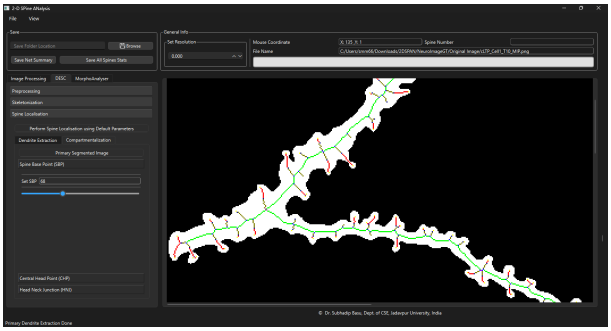

Figure 32: Setting Spine Base Points (SBPs) by adjusting the slider. SBPs (yellow dots) mark the junction points where spines connect to the dendrite shaft, providing crucial reference points for spine length calculations and morphological analysis.

6.4.1 Step 10: Setting Spine Base Points (SBPs)

The first step in spine compartmentalization is to identify the base points where spines connect to the dendrite:

1. Navigate to the "Spine Compartmentalization" sub-tab. 2. Locate the "Spine Base Point (SBP)" section and adjust the slider to optimize SBP placement. 3. Start with a moderate value (around 50) and incrementally adjust until all spine bases are accurately marked, as shown in Figure 32. 4. Click "Set SBPs" to confirm the placement of Spine Base Points.

SBPs mark the junction where spines connect to the dendrite shaft and serve as reference points for spine length calculations.

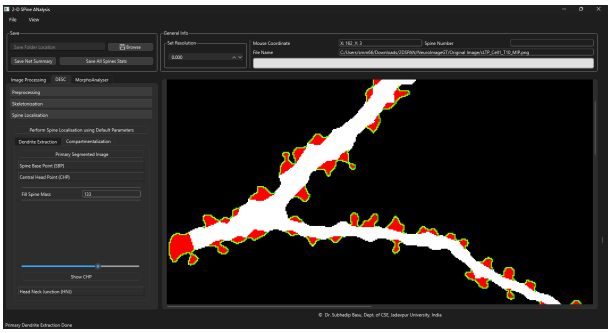

Figure 33: Spine filling process through slider adjustment. This critical step determines the extent of each spine's boundaries, requiring careful balance between complete filling and avoiding the merging of adjacent spines.

6.4.2 Step 11: Spine Filling

After setting SBPs, proceed to fill the spines:

1. Locate the "Spine Filling" controls and adjust the slider to determine the extent of filling. 2. Begin with a conservative value (around 30-40) and gradually increase until spines are adequately filled, as shown in Figure 33. 3. **IMPORTANT: Exercise careful judgment during this step. Excessive filling may cause nearby spines to merge, while insufficient filling may result in incomplete spine detection.** 4. The quality of spine filling directly impacts the accuracy of subsequent morphological measurements. 5. Click "Fill Spines" to execute the filling operation.

This step requires expert judgment to balance between complete spine filling and maintaining separation between adjacent spines.

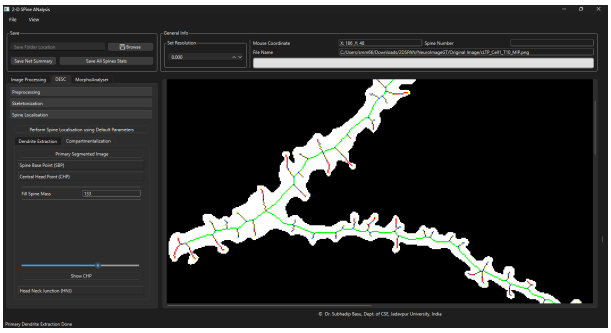

Figure 34: Central Head Points (CHPs) identification after spine filling. CHPs (magenta dots) represent the geometric centers of spine heads, providing essential reference points for spine head diameter measurements and morphological classification.

6.4.3 Step 12: Central Head Point (CHP) Identification

Once spines are filled, proceed to identify their central head points:

1. Locate the "Central Head Point (CHP)" section in the Spine Compartmentalization tab. 2. Click "Identify CHPs" to automatically detect the central points of each spine head. 3. The software will display the CHPs as magenta dots, as shown in Figure 34. 4. These points represent the geometric centers of spine heads and are crucial for accurate morphological analysis.

CHPs serve as reference points for measuring spine head diameters and play a key role in spine classification.

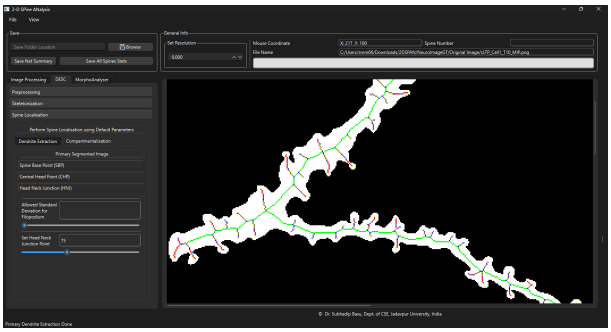

Figure 35: Head-Neck Junction (HNJ) setting for spine classification. HNJs (blue dots) mark the transition point between spine neck and head, which is critical for distinguishing mushroom-type spines from stubby spines.

6.4.4 Step 13: Head-Neck Junction (HNJ) Setting

The final step in spine compartmentalization is to identify the junction between spine heads and necks:

1. Locate the "Head Neck Junction (HNJ)" section in the Spine Compartmentalization tab. 2. Adjust the slider to optimize HNJ placement, starting with a value around 50. 3. Note that filopodia-type spines do not have defined heads, so HNJs are only marked for mushroom and stubby spine types. 4. The resulting HNJs are displayed as blue dots, as shown in Figure 35. 5. Click "Set HNJs" to confirm the placement.

HNJs are critical for distinguishing between mushroom and stubby spine types, as they mark the transition between spine neck and head.

6.5 Morphological Analysis

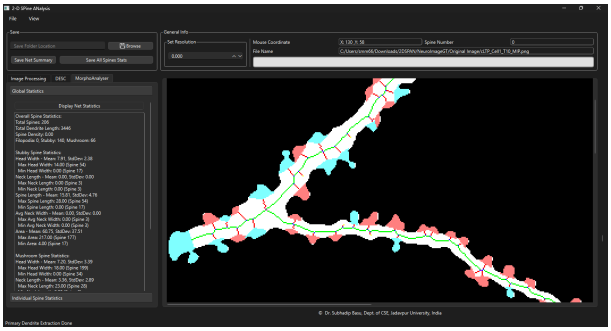

Figure 36: Global spine statistics summary displaying comprehensive metrics including spine density, type distribution, and aggregate morphological measurements across all detected spines. This overview provides valuable insights into the overall dendritic spine population.

6.5.1 Step 14: Global Statistics Analysis

After completing spine compartmentalization, proceed to analyze spine statistics:

1. Navigate to the MorphoAnalyser Module and select the "Global Statistics" tab. 2. Ensure that proper resolution settings have been configured in the General Information panel. 3. Click "Display Net Statistics" to generate comprehensive metrics across all detected spines. 4. The resulting statistics display is shown in Figure 36, providing insights into spine density, type distribution, and aggregate morphological measurements.

Global statistics offer a broad overview of dendritic spine characteristics across the entire image, enabling population-level analysis.

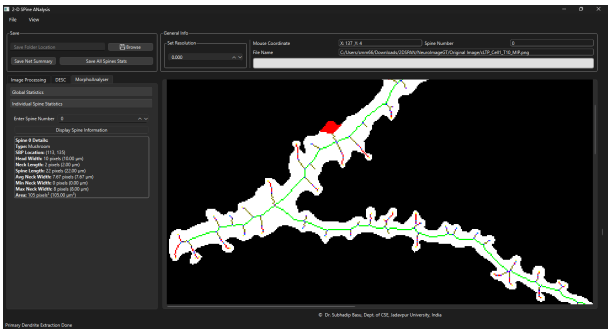

Figure 37: Individual spine statistics display for Spine 0, showing detailed morphological measurements including length, head diameter, neck diameter, and classification type. This detailed analysis enables precise characterization of each spine's unique properties.

6.5.2 Step 15: Individual Spine Analysis

For detailed analysis of specific spines:

1. Navigate to the "Individual Spine Statistics" tab in the MorphoAnalyser Module. 2. Enter the spine number (e.g., 0) in the provided field or use the increment/decrement arrows to select a spine. 3. Click "Display Spine Information" to generate detailed metrics for the selected spine. 4. The resulting display, as shown in Figure 37, provides comprehensive measurements including spine length, head diameter, neck diameter, and classification type.

Individual spine statistics enable in-depth analysis of specific spines of interest, facilitating detailed morphological characterization.

6.6 Results Visualization

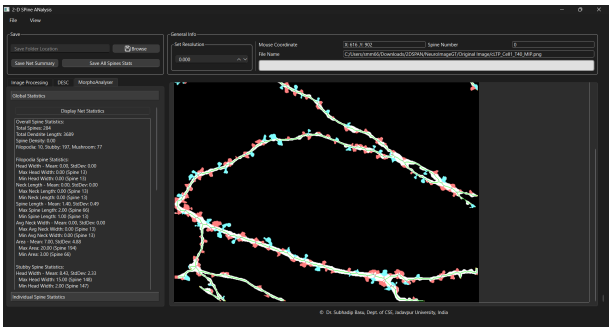

Figure 38: Classification results overlaid on the binary image. Different spine types are color-coded (mushroom: blue, stubby: yellow, filopodia: red), providing a clear visualization of spine type distribution throughout the dendrite.

6.6.1 Step 16: Visualizing Classification on Binary Image

To visualize the classification results on the binary image:

1. Navigate to the View Menu. 2. Select "Classification on Binary" to display the spine classification overlay on the binarized image. 3. The resulting visualization, shown in Figure 38, color-codes different spine types (typically mushroom: blue, stubby: yellow, filopodia: red). 4. This visualization provides a clear overview of spine type distribution throughout the dendrite.

The binary overlay offers a clean, high-contrast visualization of spine classification results against the simplified binary background.

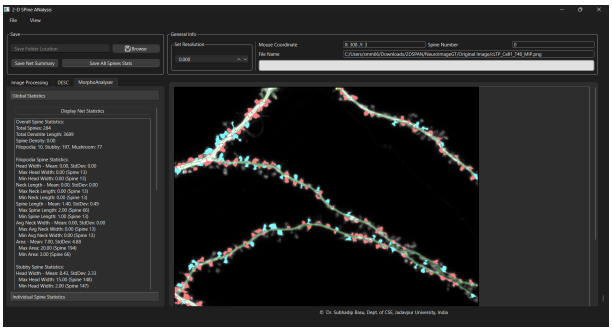

Figure 39: Classification results overlaid on the original grayscale image. This visualization provides context for spine classification within the original neuronal structure, facilitating validation of analysis results against the raw image data.

6.6.2 Step 17: Visualizing Classification on Original Image

For context-rich visualization of results on the original image:

1. In the View Menu, select "Classification on Original" to display the spine classification overlay on the grayscale image. 2. The resulting visualization, shown in Figure 39, presents spine classifications in the context of the original neuronal structure. 3. This visualization is particularly useful for validating analysis results against the raw image data.

The original image overlay provides biological context for the classification results, allowing researchers to verify spine detection and classification in relation to the actual neuronal morphology.

6.7 Saving Analysis Results

To preserve the complete analysis results for future reference or further analysis:

1. Navigate to the File Menu and select "Save Current Image" or use the Save Analysis System panel. 2. Choose the appropriate file format and location for saving. 3. The saved file will preserve all analysis overlays, annotations, and results present in the current view. 4. Additionally, consider exporting

statistical data by first setting the path of the folder where one wants to store the analysis result in the Save Panel. Then by mere clicking the Save Net summary and Save Individual Spine Summary, csv files in the mentioned folder path are created.

By following this comprehensive workflow, users can effectively utilize 2D Span Auto to conduct thorough and accurate dendritic spine analysis on neuronal images, regardless of their prior experience with the software.