A Platform for the Semi-automatic Segmentation, Characteristic Shape Analysis and Simulation of Neuronal Somas

Shidan Javaheri, Siger Ma, Athmane Benarous, Thibaut Baguette, Amir Shmuel

***Abstract*—Segmenting neuronal somas in 3D brain images through techniques such as tissue clearing and Light Sheet Fluorescence Microscopy promises to advance our understanding of healthy brains and the pathophysiological roots of neurological diseases, but it is costly and time-consuming. Automating the segmentation through deep neural networks shows promise but requires labeled ground-truth data that is labor-intensive and expensive to obtain. This work provides a platform within the open-source 3D medical image analysis software Vaa3D to quickly and precisely label and automatically segment neuronal somas. The binary segmented image is generated using a region-growing algorithm on somas that have been marked and bounded by a sphere around their center of mass, guided by a local Otsu threshold that can be manually overridden. The platform further supports principal component analysis of the segmented somas, allowing them to be rotated about their center of mass and aligned and for their characteristic shape to be captured in a probabilistic model. Finally, the platform supports the simulation of ground truth data, allowing segmented somas to be reoriented and repositioned according to probabilistic data, generating synthetic ground truth data that can be used to train deep neural networks. The platform's source code is released along with this paper, allowing future work to build on its initial architecture: https://github.com/orgs/ImagiNeuron/repositories**

***Index Terms*—Neuronal somas, Cell Segmentation, Characteristic Shape Analysis, Simulation**

I. Introduction

THE advent of tissue clearing [1] and Light Sheet Fluorescence Microscopy (LSFM) [2] as 3D imaging techniques allows for the high-resolution imaging of the entire brain of animals like rats, and its microscopic structure, including isolating the brain's neuronal somas. Analyzing and segmenting these somas at a large scale allows us to uncover more about the structure and properties of neurons of the brain, advancing our understanding of the functions of a healthy brain and the pathophysiological roots of neurological diseases. At this scale, terabytes of data are practically impossible to label manually, necessitating the automation of the segmentation process. Various efforts have shown the promise of deep neural networks and convolutional neural networks (CNN) in automating this task [3]–[8]. However, the most accurate of these methods require ground truth data to train the machine learning models, which is labor-intensive and time-consuming to obtain [3], [4], and software that supports it is often expensive and inaccessible to the broader research community.

This paper was produced by ImagiNeuron, a group of 4 McGill Students working on their 2024-2025 Engineering Capstone project.

LSFM is an imaging technique that uses a plane of light to image transparent tissue [9]. Used in conjunction with tissue clearing, it is an effective way to isolate and image neuronal somas in the whole brain of various species, including rats. Another common imaging technique for analysis at this scale is Fluorescence Micro-Optical Sectioning Tomography (fMOST). This physical imaging technique is more common than LSFM in literature as the basis of the analysis of neuronal structures in the brain and in deep learning pipelines for the automated segmentation of neuronal somas [3], [5]–[7]. However, for both imaging techniques, labeled ground truth data consisting of a binary segmentation of neuronal somas in the brain is rarely released. While the Big Neuron project [10] has focused on the automated reconstruction of singular neurons in Optical Microscopy Images and releases some ground truth data, none of it consists of binary labels for every voxel in a 3D image. Very recent work has focused on using a collaborative platform to quickly label the location of neuronal somas in the brain [11], successfully doing so for 180,000 somas in 122 different fMOST brains [12]. However, the work only cites 4 LSFM brains, in which no somas are identified. And again, while the project releases data of 3D soma locations in the brain, it and many other efforts in this area fail to provide access to binary segmentation of neuronal somas that can be used as ground truth data to train deep neural networks. There is a clear gap in accessible ways to quickly and accurately obtain ground truth data of this nature, particularly for neuronal somas imaged with LSFM.

This platform aims to provide an accessible tool for fast and precise segmentation of neuronal somas in an entire brain. In particular, the focus is on automatically segmenting neuronal somas that can be encapsulated by a sphere around their center of mass, without that sphere overlapping with other somas. In addition to supporting the correction of the automatic segmentation, the tool aims to support advanced visualization of segmentation results, including an overlay on the original image and a clear outline of the segmentation for its qualitative evaluation.

The platform further aims to support a tool to analyze the characteristic shape of somas with principal component analysis (PCA) on segmentation results. For comparison, intensity-weighted PCA on the original soma intensities is also

supported, and both results can be visualized. PCA results allow all segmented somas to be aligned at their center of mass and along their principal components, and for a resulting probabilistic shape of a neuronal soma to be built and visualized. Finally, the platform supports the simulation of data, generating a realistic synthetic image with intensities and its corresponding binary segmentation. This simulation seeks to address the scarcity of available ground truth data, aiming to increase its quantity and diversity for the training of deep learning networks.

II. Related Work

A. Image Processing for Cell Segmentation

Given a 3D microscopy image of cells such as neuronal somas, factors such as intensity variations and noise make it difficult to segment and separate those bodies from their background. To help accomplish this complex task without the need for manual labeling of individual voxels, image processing techniques have been developed to automatically or semi-automatically segment cells.

Some image processing techniques have been proven effective without manual labeling, such as the watershed algorithm, which treats image intensity as a topographical map [13]. While relevant, the algorithm's segmentation can be prone to overflow from their intended soma regions, which is especially true in noisy areas [14]. The algorithm's global approach thus often struggles with local areas.

Alternatively, algorithms such as region-growing can use manually labeled seed points to initiate the segmentation of a cell. At its core, a region-growing algorithm iterates by recursively segmenting neighbors of the currently segmented voxels if they are below a certain intensity threshold. The most important element of region-growing is selecting an appropriate threshold to stop the expansion of the segmented volume and to accurately determine the 3D boundary of the cell body, separating it from the image background. There are various methods for selecting the best threshold. Choosing a global threshold for the whole image will not lead to an effective segmentation, as the intensity and boundary of every cell are different. Most cases necessitate selecting a specific threshold for each cell body.

One approach explores the use of 'iterative thresholding' [15]. This method involves starting with a very high threshold for each body, successively lowering it, and applying region growing in sequence until the segmentation's center of mass strays past a limit based on its distance from its previous position - the limit being used as a heuristic for over-flooding. This approach does not always succeed, necessitating further heuristics to keep or discard a segmentation before over-flooding is detected, which makes it very inefficient. However, the architecture of its implementation implementation is a helpful reference for image segmentation techniques in open source software, and is discussed further in Section III.

Research indicates the effectiveness of local Otsu thresholding as an approach to separate the voxels of a cell from the background [16]. This method uses a histogram to separate foreground from background voxels and seeks to choose an optimal threshold by minimizing intra-class variance and maximizing inter-class variance [16]. This approach was chosen as the basis of our design.

B. 3D Visualization-Assisted Analysis (Vaa3D)

3D Visualization-Assisted Analysis (Vaa3D) is an open-source software platform for the visualization and quantitative analysis of large-scale biological image datasets [17]–[19]. The platform supports viewing and labeling images in 2D and 3D. Labeling is supported by placing markers on cells in the image to contain each cell within a sphere around its center of mass. The platform also supports TeraFly, which provides Google Earth-like functionality that supports navigation through terabytes of data of the entire brain. The data is loaded on the fly, and the resolution is adjusted as the user navigates through it at different levels of resolution [20]. TeraFly also comes with a tool called TeraConverter [21], that allows the automated conversion of 2D microscopy images taken at various slices in the z-axis into the 3D format supported by TeraFly, allowing seamless integration. Furthermore, Vaa3D allows developers to add plugins to it, supporting the development of additional functionality. It provides a strong basis to build robust and accessible pipelines to facilitate medical image analysis, particularly given its recent update [22].

C. Deep Learning based Soma Segmentation

Deep learning and CNNs have been gaining popularity for computer vision tasks like biomedical image classification and segmentation [23], and they often outperform more traditional computational and machine-learning methods. However, they rely on the availability of enormous labeled datasets to prevent overfitting and improve their generalization. There is often limited biomedical image training data, and producing more well-annotated labeled data can be expensive and time-consuming [3], [5].

Several efforts exist that have successfully applied deep learning to soma segmentation. Many have explored applying deep neural networks to combat the high cost of obtaining labeled data for segmenting somas from 3D microscopy data, by avoiding using a complete binary segmentation of the brain as ground truth data. A following brief overview of a few of these methods indicates that while they are somewhat successful, they are less effective than those trained more extensive ground truth data. Most of the work in this area uses fMOST 3D images of the brain, with no studies done on LSFM data to our knowledge.

Recent work [6] explores using deep neural networks to locate the centers of neuronal soma centers in fMOST images of the brain rather than explicitly segmenting them, given that manual labels of the centers of mass of somas are faster to obtain. The method uses a 2-stage neural network and a custom loss function to define the objective of correctly locating somas centers. While it achieves a high precision of 98%, its design ensures a low false positive rate, and it does not classify neuronal somas in dense brain areas, reflected in its low F1 score of 64%.

Another paper [7] explores what it refers to as a "weakly supervised" approach. Ground truth data consists of labels for each neuronal soma and a radius defining a sphere that completely envelopes each soma, which is also relatively easy to obtain. For training, voxels are classified as "foreground" within the sphere and "background" if they are far enough from all other spheres, with other voxels ignored. This approach leads to lower levels of precision at 78%, but higher levels of recall at 83%.

One method [5] attempted to do away with ground truth data altogether. Working with fMOST images of the brain, it uses a "hessian analysis based enhancement filter" to automatically create weakly supervised labels for all voxels. A deep neural network is trained on this data to segment neuronal somas, with reported results with a high F1 score of 97%. However, further research into the project's code-base revealed that the results may be misleading and not reproducible, as little evidence supported the paper's claims.

Methods that use a binary segmentation of somas as ground truth data have achieved better results. The UNET CNN architecture is well known to have been well-performing for medical image analysis [8]. Recent work [3] cites the promise of building on this architecture for soma segmentation applications, and adapts UNET to include multi-scale feature extraction to enhance finer details, adaptive weighting to measure feature importance, and dynamic convolution to combat the sparsity in neuronal soma data. It achieves an F1 score of 91.8%, an improvement compared to other UNET architectures it is compared to, including 3D UNET [24], SegNet [25], HRNET [26], Attention UNET [27], Res UNET [28], and RAUNET [29]. Despite its high performance, its code and ground truth data are not made available.

A brief survey of methods for soma segmentation indicates the importance of obtaining a binary segmentation of neuronal somas in the brain as ground truth data. However, data of this nature is expensive to obtain and rarely made publicly available, particularly for LSFM data. Our platform seeks to address these limitations by providing a means to quickly obtain a binary segmentation of neuronal somas to be used as ground truth data, as well as to enhance its quantity and diversity through simulation.

D. Data Augmentation

Data augmentation is essential for combating the high cost of obtaining a large amount of labeled data necessary to build high-performing deep learning and CNN pipelines [30]. Data augmentation methods alter the original image with a well-defined operation and add it to the dataset the model is built on, improving the data's quality and diversity. This imitates human imagination and dreaming, helping the model experience different scenarios and learn from them. Different image data augmentation techniques exist, including basic, deformable, and deep learning augmentation techniques [30]. Table I summarizes these data augmentation techniques and their categories.

Deep learning approaches for segmentation tasks require image-label pairs. Data augmentation can generate a substantial amount of synthetic image-label pairs semi-automatically, using a limited number of well-labeled examples encompassing a diverse range of neuronal somas from different brain regions.

GAN-based methods are gaining popularity among the data augmentation techniques to generate realistic synthetic image-label pairs suitable for image segmentation [30]. GAN-based data augmentation methods necessitate a significant amount of initial data for training [31], which poses practical challenges for our dataset that contains raw image data but lacks sufficient labeled examples.

Deformable data augmentation techniques align well with our dataset, segmentation, and augmentation objectives. From a limited labeled dataset, statistical shape models can combine the available data to describe neuronal somas' shape, position, and orientation variability with probability models. These models can generate realistic synthetic image-label pairs through deformations within plausible parameters. Previous studies have used this data augmentation technique to improve the biomedical image segmentation performance when the available training data is limited [33]–[36].

III. Method

A. Overall Architecture

The platform we built is designed and provided as a plugin to Vaa3D, making the most of this open-source platform's visualization, navigation, labeling, and image processing features. We further ensure that the functions in our plugin are compatible with TeraFly, allowing for simple conversion of 2D slices of microscopy data into a 3D format, and the seamless navigation through and analysis of terabytes of data. We name our plugin Machine Intelligence for Neuron Detection (MIND); all functions are accessible under it.

Our platform uses Vaa3D's built-in markers to provide semi-automatic segmentation of somas with minimal manual labeling, allowing users to contain a soma within a bounding sphere around its center of mass. These labels are faster to obtain than manually labeling each voxel, and provide the basis for a region-growing-based segmentation algorithm to automatically generate a binary segmented image, which is then also visualized in detail for qualitative analysis of results. The platform is thus built on the premise that it will only be able to label and process somas that can be isolated in this way, leaving extensions to somas in denser areas of the brain for future work.

The platform further supports analysis of the characteristic shape of somas. When the binary segmented image is generated, the outputs include files containing the PCA results on each segmented soma and a probability model of a soma shape built from all the segmented somas. These outputs can be visualized through separate functions provided through our plugin. Finally, the platform uses the segmentation files and analysis of soma shapes to generate realistic simulations of soma data and their corresponding binary segmentation. This feature provides data augmentation that is critical for building deep learning pipelines to completely automate the soma segmentation task. This plugin's main workflow and use are depicted and described in Figure 1. The details of each

TABLE I
SUMMARY OF DATA AUGMENTATION TECHNIQUES AND THEIR CATEGORIES [30]–[32]

Category	Technique	Description
Basic	Geometric Transformations	Scaling, translating, rotating, reflecting, shearing, skewing
	Cropping	Select a random patch from the image
	Occlusion	Remove random patches from the image
	Intensity Operations	Pixel/voxel manipulation; gamma correction; contrast enhancement
	Noise Injection	Add Gaussian, uniform, or salt-and-pepper noise
	Filtering	Apply kernels to sharpen, blur, or smooth the image
	Combination	Combine two or more original images
Deformable	Displacement Field	Random value shifts each pixel/voxel
	Spline Interpolation	Use piecewise polynomials to estimate values between data points
	Image Registration	Align moving image to match a fixed image
	Shape Models	Generate shape variability models across the dataset
Deep Learning-Based	GANs	Generate realistic images via adversarial learning
	Feature Augmentation	Augment representations in feature space

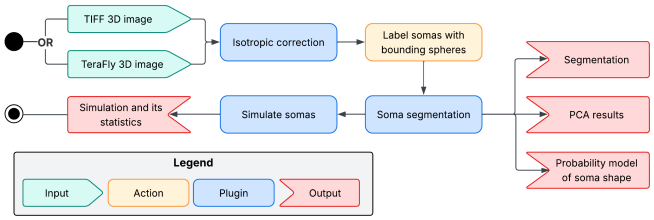


Fig. 1. The main workflow and architecture of the segmentation and simulation pipeline. Once a user is viewing an image in Vaa3D, either through TeraFly or by loading a 3D image into Vaa3D directly, the plugin's functions are accessible. The isotropic correction function is used to adjust for the actual dimensions of a voxel, determined by the resolution of the microscopy technique and the separation between subsequent Z slices, for 3D visualization and calculations. The user can then label each soma, bounding each one within a spheres of particular radius, centered at each the soma's center of mass. Once a sufficient number of somas have been labeled, the image can be segmented with the soma segmentation function, producing an output of the binary segmented image of the labeled somas, the results of PCA on the segmented shapes, and the probabilistic model of soma shape. With these results, simulation of soma data can be generated with a custom number of synthetic somas.

function are outlined further during our discussion of results in Section IV-B.

Our platform thus has four key features: the semi-automated segmentation of somas, PCA of segmented somas, probabilistic modeling of soma shape, and soma simulation. The method used to achieve each of these functionalities is discussed in detail in the following sections. During processing and analysis, the outputs of each function are always automatically saved. A different approach is used to save files processed with and without TeraFly, and both are clearly documented.

B. Segmenting Somas

Our solution for semi-automated soma segmentation uses an existing Vaa3D plugin as a basis, drawing from its implementation of region growing to segment cells [15], [37], but vastly improving on its iterative approach to using local Otsu thresholding to find the optimal threshold to segment each soma - as highlighted and discussed in Section II. This function of the plugin is designed to run once the user has isolated all the somas they want to segment within a sphere around their center of mass, using the markers in Vaa3D.

The soma segmentation function allows the user to preprocess the 3D image using an efficient histogram-based 3D median filter, using a user-defined window size. This aims to reduce salt-and-pepper noise, which can poorly affect soma segmentation results, without blurring the edges of somas [38]. Once the segmentation process is initiated, a 3D region-growing algorithm is employed with starting from the center of each soma labeled by the Vaa3D markers. Along with its center, each marker has a radius defining the volume of the bounding sphere, which, by default, also defines the maximal segmentation volume. The user has the option to select one of three methods to determine the optimal intensity threshold used to flood soma voxels. The first two options are provided for comparison, but the optimal one is the third:

1) Iterative Thresholding: This method was initial provided by the reference plugin [15], and it determines the threshold for each soma iteratively. It and its connected hyperparameters were left as part of the plugin for reference and comparison, but its use is not recommended.
2) Global Otsu Thresholding: This method determines an optimal threshold to separate the foreground and background of the whole image after being given a histogram of the entire 3D image. This method is also for reference and shows the weakness of using a global thresholding method, which is not recommended.
3) Local Otsu Thresholding: Instead of considering the whole 3D image, this method determines an Otsu threshold specific to the spherical volume defined by the Vaa3D marker around each soma. This allows each calculated threshold to better reflect each soma's local intensities. This specificity means the local Otsu threshold produces the best results and is the method provided by default.

By default, the behavior of the segmentation algorithm is to constrain the segmented region of each soma to remain within the spherical volume defined by each soma marker - consistent with the fundamental assumption behind our application about the way somas are labeled. However, the user can disable the "marker constraint" option in the segmentation functions configurations if desired. This feature was added for flexibility, but its use is not recommended.

After completing the function, the segmentation results can be adjusted and corrected to increase accuracy. The intensity

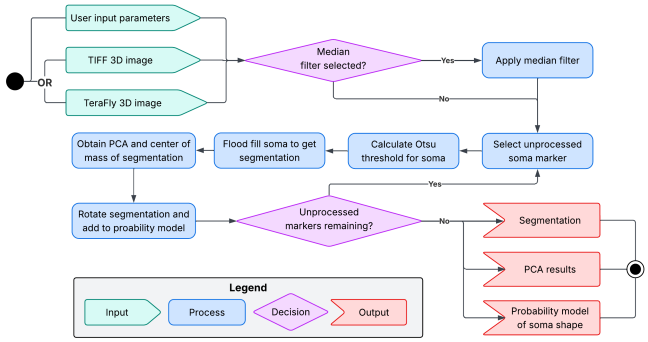


Fig. 2. The main workflow and architecture for soma segmentation, using local Otsu thresholding on each soma

threshold value determined for each segmented soma using the user-selected thresholding method is automatically saved into the comments field of the Vaa3D marker. For example, a soma with an optimal threshold of 30 would have the following comment: "threshold:30". Should the user notice an under-flooded or over-flooded soma segmentation through visual inspection, they can manually change the numerical threshold value stored in the marker's comment. Selecting the "Manual Thresholding" option and running the segmentation plugin once again will enforce the new threshold constraint for the specific soma. This allows the user to direct correction efforts without affecting other somas.

When the segmentation processing step within our plugin is complete, the segmentation visualization is automatically opened in a new image window. This image contains three channels, shown as red, green, and blue by default. The first channel contains the original image data, the second, the segmented binary image data, and the third, the image gradient or outline of the segmented image. All three channels can have their color customized, such that, for instance, the original image is shown in greyscale, the segmentation is in red, and the outline is in cyan. The channels can also be individually hidden if desired. This new image can also be linked to the original to show the same location in the 3D image as the user scrolls through one of the axes. Finally, the image can also be opened in the 3D viewer to have a global view of the segmentation. The image gradient is computed using a Sobel filter [39] applied in three dimensions, implemented with the OpenCv library for C++ [40]. An additional step of non-maximum suppression is applied to get a thinner edge, allowing the user to see the shape of the segmentation more easily on top of the original image data. These results are explored in Section IV.

C. Principal Component Analysis of Somas

As one of its steps, the soma segmentation pipeline includes 3D PCA of the shapes of the segmented areas of neuronal somas. PCA is performed on the coordinates of the neuronal soma voxels to capture and understand their shape. The first principal component PC_1 describes the orientation of the shape axis with the most variance, the longest axis in the shape. The second principal component PC_2 is the second longest axis orthogonal to the first, and the third principal component PC_3 is the third longest axis orthogonal to the two others. These three components explain the spatial variance of the soma shape around its center of mass to its boundary [15], and are captured in a shape feature vector composed of normalized eigenvectors eig_n and their eigenvalues σ_n^2, where $n \in \{1, 2, 3\}$. The direction of each principal component is thus represented by its eigenvector, and its length captured in its eigenvalue.

The platform also supports intensity-weighted PCA on unsegmented images to use information on voxel intensity to characterize soma shape. We use PCA without intensity-weight on the binary segmented image to focus solely on spatial and shape information, the platform still supports intensity-weighted PCA. The statistics gathered for each neuronal soma through the former approach serve the probability model of soma shapes, as well as the simulation processes. The statistics of the latter approach are included for reference and comparison.

The PCA data can be visualized using our plugin's "visualize PCA" function. Through this, a new color image window is opened with the original image and, for each labeled soma, a representation of its three principal components. The longest principal component is colored green, the second longest red, and the third cyan. In this case, the colors cannot be changed. The new image will be opened with the default 2D viewer of Vaa3D, but due to the nature of the visualization, in which each principal component is a thin 3D vector, it is preferred to view it in 3D. The eigenvectors are drawn as lines starting from the center of mass using Bresenham's line algorithm [41], which allows fine line representations in a low-pixel-density scenario. To obtain an estimate of the length of each principal component in 3D, the full width at half maximum measure is used [42], which for a principal component represented by the eigenvector eig_n and the eigenvalue σ_n^2, assumed to result from a gaussian distribution, can be calculated as $\sigma \times \sqrt{8ln2} \approx 2.35\sigma$. These results are further explored in Section IV.

D. Probability Model of Soma Shape

The eigenvectors defining the principal components of the binary volume from the segmentation of each soma, eig_1, eig_2 and eig_3, as well as the soma's center of mass, allow us to understand a soma's orientation in 3D space. With this information, we can take the binary volume of every segmented soma and place its center of mass at the origin of a 3D cartesian coordinate system. We can then rotate it about its center of mass to align its principal components with the cartesian axes, so that eig_1 aligns with the y-axis, eig_2 aligns with the x-axis, and eig_3 aligns with the z-axis. Once all the somas have been aligned and overlapping, we can sum all the binary soma volumes to get a count value for each voxel, indicating the number of somas each voxel is a part of. By dividing by the total count of somas, we obtain a probability model that characterizes soma shape in one area of the brain.

A simple heuristic is used to check whether the user placed any labels poorly during the segmentation process. If, after

rotating the binary segmentation of a soma about its center of mass, the value of the voxel at the center of mass is 0 (i.e not part of a soma), the soma is flagged as having poor segmentation, and a notification is displayed to the user. Output in the Vaa3D terminal displays a slice through the center of the binary volume of the rotated soma, allowing the poor segmentation to be visualized approximately. The user can then adjust the marker of the corresponding soma to address the issue before segmenting again.

The rotation of the soma from a source image into a target image where its principal components are aligned with the cartesian axes is approached as follows. Once a single soma is segmented, its segmented volume can be extracted to create a source image, in which the soma's principal components are not aligned with the cartesian axes. The coordinates of every point in the volume of the source image relative to the soma's center of mass can be calculated and used in all future operations. A rotation matrix A can then be defined, whose columns a_1, a_2, a_3 are the eigenvectors representing each principal component of the soma, $a_1 = eig_2, a_2 = eig_1, a_3 = eig_3$. Multiplying by A^T maps coordinates in the source image to the target, and multiplying by A maps coordinates in the target image to the source.

This rotation operation is prone to numerical errors that need to be mitigated. This possibility for error arises from the fact that the principal components defining a soma's orientation support floating point precision, and can be in any three orthogonal directions. However, the volume in which somas are represented is discrete, with only integer coordinates. This means that any mapping between the coordinates of the source and target matrices using matrix A maps integer coordinates to floating points that need to be rounded, creating numerical imprecision.

A simple method was employed to combat numerical errors in generating the target image, with further improvements left for future work. The target image is generated by iterating over every voxel in it to ensure that every one of its voxels has a value. Each coordinate in the target image is mapped to the source image by multiplication with the matrix A. Since the resulting coordinate in the source image is a floating point, we take eight samples from the $2 \times 2 \times 2$ sub-volume that captures the source coordinate. The value assigned to the target voxel in the target image is the average value of the voxels in this sub-volume. The target image generated thus contains a 3D volume of floating points with values between 0 and 1, describing the shape of the soma, which is now aligned with the cartesian axes. The floating points values are kept to avoid the numerical error of using a thresholding function to obtain integer values again.

Each aligned soma is added sequentially to construct a model that describes the overall soma shape probabilistically. Dividing the value of all voxels by the value at the center of mass gives the probability of each voxel in the 3D volume being part of a soma. The probability model is saved as a binary file in a simple format: the first 24 bytes contain three 64-bit integers representing the image's x, y, and z dimensions. The rest of the file contains double-precision floating point values representing count of the number of somas that were found overlapping with each voxel.

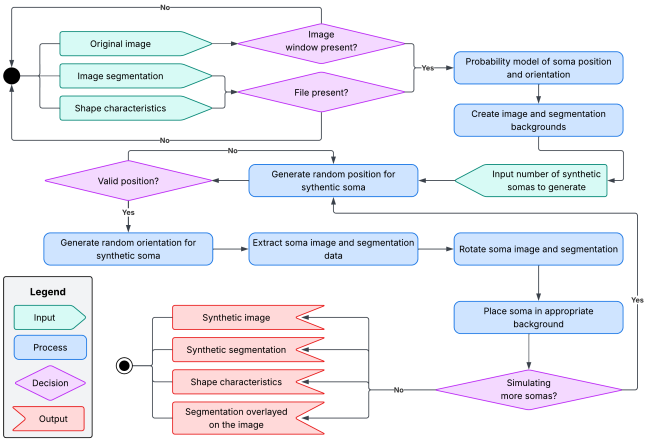


Fig. 3. Workflow and architecture of the soma simulation pipeline

We can visualize the probability model's results using the plugin's "visualize probability model" function, which loads a user selected probability model in a new image window and represents it as a heatmap. This will open by default in Vaa3D's 2D viewer, but can also be opened in a 3D view. A legend will also open in a separate window to present the color mapping of the heatmap. These results are explored in Section IV.

E. Simulation of Somas

The platform being proposed also allows augmentation of existing ground truth data through statistical shape models to increase the quantity and diversity of realistic image-label pairs. Figure 3 illustrates the methodology of the simulation pipeline. The simulation takes the original image as input, its segmentation, and the characteristics of the neuronal somas, including each soma's ID, marker coordinates and radius, center of mass, and principal components. We obtain the normal distribution of the soma's position and orientation from each soma's center of mass and principal components. Segmented somas are randomly selected, reoriented, and placed into two synthetic images that are output, based on the learned probability distributions. The first output image is the realistic simulated image with the original somas's intensities, while the second contains the corresponding binary segmentation of that image.

To operate, the simulation first generates two synthetic backgrounds, one for the synthetic image and one for its segmentation, to form the image-label pairs. The background with realistic intensities is not uniformly defined globally for the image, which would fail to capture the complex background intensity variations and patterns that can be observed in real data. Instead, local intensity variations in the image are accounted for by dividing the image volume into 16 smaller, local 3D chunks. An intensity histogram is computed within each chunk, and an Otsu threshold [43] is determined and applied to each chunk to separate soma voxels from their background. Gaussian modeling, which has been used in previous works to synthesize neuron images [44],

is then calculated using the mean and standard deviation of the intensity values of the identified background voxels from each chunk. To ensure a coherent final background image, instead of simply obtaining voxel intensities from the Gaussian distribution of each chunk, we blend the statistical parameters of surrounding chunks using distance-based weighting, which has been used to stitch microscopy imagery in previous work [45]. This allows us to calculate blended Gaussian statistics for each voxel by taking its distance to each neighboring chunk up to a set blending radius, which we chose to set as a relative measure, at 1 chunk. That weight is also set to fall off quadratically as the distance decreases, to preserve a chunk's local intensities. Once normal distributions based on the blended Gaussian parameters are calculated for each voxel, we generate its pixel value using the Box-Muller transform [46].

The simulation pipeline generates new soma images with random positions from the normal distribution of labeled soma positions, which the method checks against each other for 3D overlap avoidance using the radius data. Then, the method generates the intensities and corresponding segmentation for these synthetic somas. The method then rotates the synthetic somas to a random orientation according to eigenvectors generated from the normal distribution of principal components orientation. The three eigenvectors need to be orthogonal to each other for a single soma. The method employs QR decomposition with Householder Reflections to obtain three orthogonal vectors, ensuring numerical stability [47], done using the Eigen library in C++ [48]. The pipeline places these rotated synthetic soma images and their corresponding segmentation within their respective generated background.

The simulation pipeline finally saves the generated image and segmentation image-label pair separately. The pipeline also saves its byproducts, including marker information and PCA data on synthetic neuronal somas. The pipeline displays the results in a new window showing the segmentation overlayed on the image, with the same view as the simulation results. All files output by the simulation function are saved within a timestamped folder, allowing multiple simulations to be generated from the same segmented image. These results are discussed further in Section IV

There remains room for future work to improve limitations with the current simulation pipeline. The generated neuronal soma shapes are not new synthetic values. Instead, they are directly taken from existing somas identified in the original image. Future work can explore deforming them based on the shape probability model computed as part of the soma segmentation pipeline. The raw intensities are also not new synthetic values. Instead, they are directly taken from existing somas. Deformable data augmentation methods like spline interpolation could simulate the synthetic intensity texture [35].

IV. Results

A. Dataset

For this project, we had access to three datasets provided by Dr. Amir Shmuel. They consisted of 3D images of the brains of two rats at a magnification of 3.6x, one of which had an additional 3D image of part of the brain at 9x. The dimensions of each voxel in the 3.6x datasets are $1.8\mu\text{m} \times 1.8\mu\text{m} \times 4.0\mu\text{m}$ (x, y, z), and the voxel dimensions of 9x magnification dataset are $0.71\mu\text{m} \times 0.71\mu\text{m} \times 1.0\mu\text{m}$ (x, y, z). All of this data was obtained through LSFM and tissue clearing, and used was used to test and qualitatively evaluate our results.

B. Vaa3D Plugin

Our plugin for Vaa3D is called MIND, and contains five main features, available as different entries within the plugin's menu: isotropic correction, principal component analysis, soma segmentation, visualization, and soma simulation. The paragraphs below will go through each functionality, briefly explaining what it does and how to use it.

1) Isotropic Correction: The isotropic correction feature allows the user to input the dimension of the image voxels along the three dimensions in micrometers, to correct for anisotropic data. This affects 3D visualizations, which use interpolations to show how the data should truly look, and also affects any calculations that rely on accurate voxel dimensions.

2) Principal Component Analysis: This function performs an intensity-weighted PCA of each labeled soma in the current image. For each soma, the PCA data generated will include the soma's ID, the x, y, z coordinates of the label, the radius of the soma, the x, y, z coordinates of the center of mass of the soma, the three eigenvalues, and the nine components of the three eigenvectors. All these results are automatically saved to a comma separated value (CSV) file that can be used for further analysis or by other parts of our plugin, such as the PCA visualization.

3) Soma Segmentation: This feature uses the labels placed by the user to segment somas automatically. It produces a binary output image where each voxel will show whether the associated voxel in the input image was part of a neuronal soma. A pop-up window will appear when selecting this option, asking the user to select the parameters for the analysis and whether or not to use pre-processing. By default, this function uses median filtering with a moving window size of 3 and local Otsu thresholding with flood filling, where the flooding is restricted to remain within the spherical volume defined by each marker. The processing will start when the user clicks the "run" button. Logging information will be printed to the terminal, and once the processing is finished, a pop-up window will open showing where the output files have been saved. A new image window will also be opened to show the results of the soma segmentation.

4) Visualization: The plugin features two visualization functionalities: one for the PCA data (the output of the PCA function or the soma segmentation function), and one for the probability model (the output of the soma segmentation function). When selecting the "visualize PCA" option, the user will be prompted to select the CSV file containing the data. Note that the original input image must first be open. Then, the plugin will open the visualization, as previously described. The "visualize probability model" will prompt the user to select the probability model file. Then, the plugin will open a new image window to present the probability model as a heatmap.

5) Soma Simulation: The soma simulation feature will use the method described in Section III-E to create new synthetic data. An image containing real data must first be opened to run this function. Then, the user must input how many synthetic somas they wish to generate, and the data will be generated and automatically saved before being shown to the user using the 2D viewer.

C. Segmentation

Due to a lack of available ground truth datasets for 3D brain imagery, particularly for LSFM imagery of neuronal somas in rats' brains, quantitatively evaluating our segmentation algorithm's accuracy proved to be beyond the scope of our project. As a result, segmentation performance was primarily assessed through qualitative visual inspection using tools integrated into the Vaa3D plugin. Given the known drawbacks of global and iterative thresholding approaches, this qualitative analysis was done with the plugin's default settings, using local Otsu thresholding.

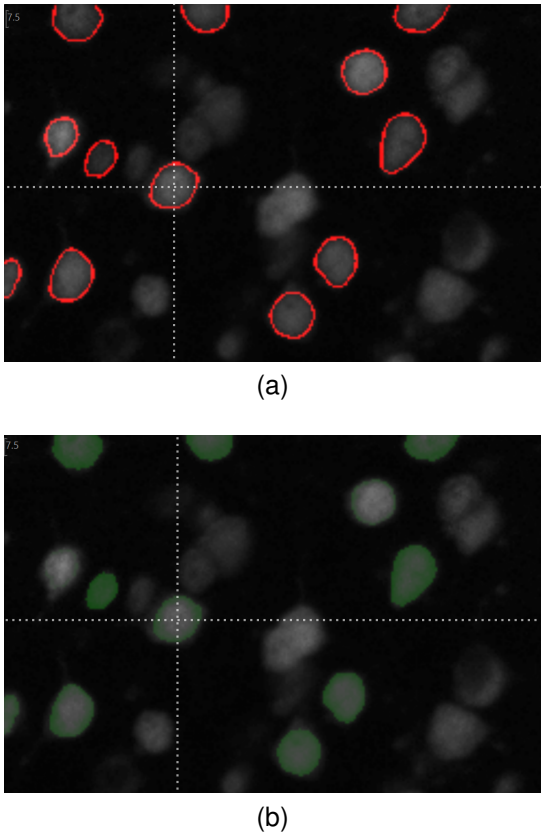
Qualitative inspection using the aforementioned visualization tools shows that our segmentation algorithm tends to produce high-quality results that closely follow the visible boundaries of the neuronal somas (see Figure 4). In contrast, the iterative thresholding method performs significantly less well, being prone to under-flooding or over-flooding depending on its input parameters, and the initial shape of the soma. This is because if a segmentation uniformly over-floods out of its soma, its center of mass will remain consistent, meaning the over flooding will avoid being detected. Similarly, if a soma has an odd shape, its center of mass movement will be captured too early, meaning the soma will not be allowed to be fully segmented. Global Otsu thresholding also yields inaccurate results, prone to under-flooding somas whose intensities are lower than average, and over-flooding those somas whose intensities are higher than average, reflcting the fact that a global threshold is too generic to describe the unique intensities of each soma.

Overall, the visual tools prove critical in determining which specific somas produced poor results and guiding the user through manual threshold corrections. Most notably, the overlay with thin lines effectively assessed segmentation boundaries and their accuracy. These results suggest that a semi-automatic approach can significantly improve the speed and usability of labeling methods compared to purely manual ones while striking a balance between fidelity and quantity of generated data.

The results of the PCA of the somas in the binary segmentation can also be visualized to confirm that the components correctly capture each soma's characteristic shape. Figure 5 shows these results in 3D, depicting their accuracy through visual inspection.

D. Probability Model of Soma Shape

The characteristic shapes of somas in different brain parts are captured within the probability model of soma shape. This model is returned as a binary file as part of the segmentation



(a)

(b)

Fig. 4. 2D View of Segmented Somas from data at a resolution of 9x, with contrast increased to 67%. (a) Results displayed with an outline of the segmentation obtained using the Sobel filter. (b) The binary segmentation transparently overlaid over the original image.

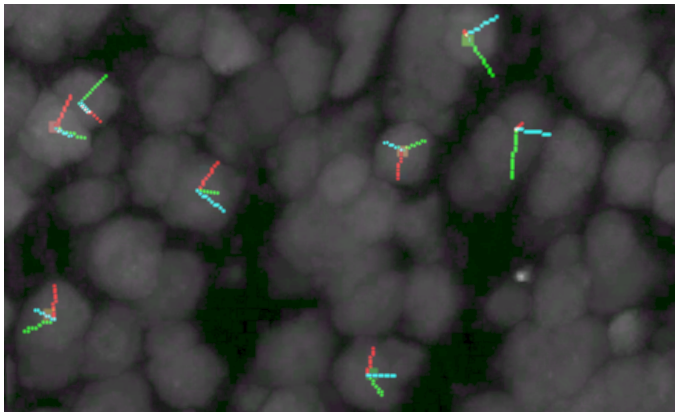


Fig. 5. 3D view of PCA Results performed on the segmentation results of somas at resolution of 9x.

results and can be visualized as a heatmap by the plugin. Qualitative inspection of results, as shown in Figure 6, demonstrates that the model effectively depicts a soma's characteristic shape.

E. Simulation

Quantitative evaluation of the effectiveness of the data augmentation process will involve future work where a deep learning model will be trained with and without the simulated data obtained through this pipeline. The data augmentation's performance evaluation will depend on the impact of the

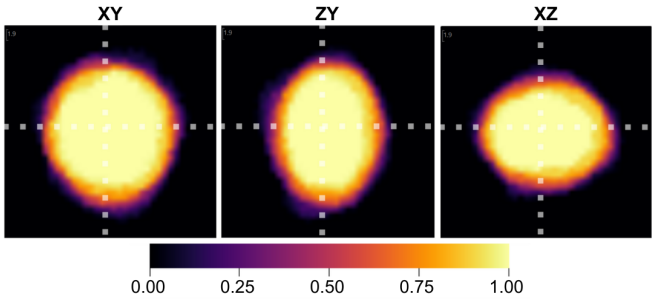


Fig. 6. The probability model demonstrates the characteristic shape of a neuronal soma. Results were obtained by segmenting 10 similar-sized neuronal somas of the 9x dataset in the same region of the brain as Figure 4.

simulated data on the model's accuracy. We currently can qualitatively evaluate the performance of the simulation through visual inspection.

Qualitative metrics used to evaluate simulation quality include how similar the simulation is to the original image, how accurate the segmentation is, and how realistic the synthetic image and its deformations are. As seen in Figure 7, the visual inspection tools integrated into Vaa3D show that our simulations tend to produce high-quality synthetic images that are realistic deformations of the original image. The segmentation label results also closely follow the performance obtained from our segmentation algorithm. However, the numerical errors that result from soma segmentation are visible in a slightly grainier outline of each soma in the synthetic image in Figure IV-E. More advanced methods for combating these numerical errors are left to future work.

Due to current limitations to the neuronal soma simulation pipeline, the resulting soma intensity texture, shape, and segmentation accuracy heavily depend on the prior segmentation's performance, as synthetic somas are reoriented and positioned copies of real somas and are not created with the probability model of neuron shapes, or with synthetic intensities. Future works to resolve these limitations can improve the simulation's performance. This work is intended to serve as an initial basis for a simulation pipeline and is made open-source to welcome improvements to its performance.

V. Conclusion

The functions in the Machine Intelligence for Neuron Detection (MIND) plugin for the open-source software platform Visually-assisted Analysis 3D (Vaa3d) successfully deliver a platform to semi-automatically segment neuronal somas, as well as to model their characteristic shape and to simulate additional ground truth data. The plugin integrates with Vaa3D and TeraFly, supporting the seamless navigation and analysis of terabytes of data, ensuring scalability towards large image datasets. The platform's primary goal is labeling and analyzing neuronal somas in 3D imagery of entire brains of rats obtained through LSFM. However, its results likely apply to cell segmentation of any intensity-based microscopy data.

The platform focuses on segmenting somas that can be isolated within a sphere around their center of mass using the

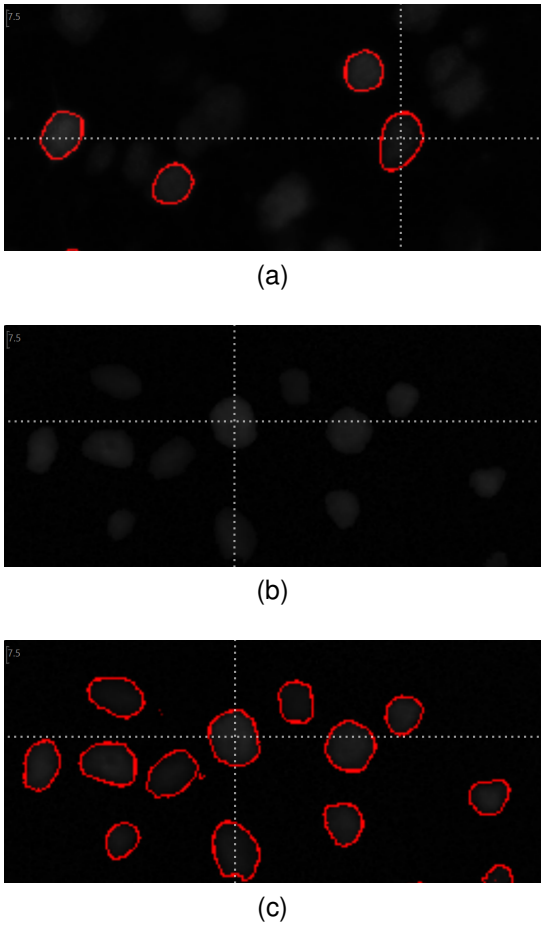


Fig. 7. Comparison of real and synthetic neuronal soma data. Synthetic data was generated from 10 labeled neuronal somas in the 9x data. (a) The original image and a few of its segmented somas. (b) The simulated image. (c) The segmentation of the simulated image.

markers that Vaa3D provides. A local Ostu thresholding algorithm is used to determine the optimal threshold defining the boundary of each soma, and a 3D region-growing algorithm starting from a soma's center of mass is applied to segment each one. Segmentation results can be analyzed qualitatively through the plugin's visualization tools, including an outline and overlay of segmentation results against the original image. These tools visually indicate a high-performing segmentation. A soma's segmentation can be corrected by manually overriding the intensity threshold that defines its boundary. Overall, these features promise to significantly speed up the time researchers spend obtaining ground truth data by labeling every voxel in the image as part of a soma, making deep learning pipelines to automate the soma segmentation more accessible.

The soma segmentation plugin also produces PCA results for each segmented soma, providing insight into their orientation and characteristic shape. The plugin supports visualizing these principal components to confirm their accuracy, and uses them to build a probability model of soma shape in one part of the brain. This provides further insight into somas' characteristics and lays the foundation for future work to create advanced simulations of neuronal somas in the brain.

Along this line, the plugin also provides a soma simulation pipeline, which creates synthetic ground truth data by taking segmented somas and reorienting and repositioning them into corresponding synthetic binary and realistic images. Qualitative inspection of simulated data reveals realistic images whose binary segmentation and intensities match the accuracy of the initial segmented data, with some minor distortion due to numerical errors. The plugin supports multiple simulations of any image in the brain, promising to be an effective data augmentation technique to increase the quality and diversity of labeled data. However, the effectiveness of this simulated data in a deep-learning pipeline remains to be seen and tested.

The platform's current implementation also contains several well-known limitations we leave to future work. Its main drawback is its focus on segmenting somas that are isolated enough to be bounded by a spherical marker. Isolating each soma is necessary for accurate analysis of soma shape and characteristics. However, with this drawback, our tool will be harder to use in high neuron-density regions of the brain. Thus, our future work could improve performance by exploring extensions that allow the isolation of somas even in areas of high density.

Future work can also be dedicated to improving the probability model of soma shapes by combating numerical errors that occur during rotation in more sophisticated ways. The simulation architecture can then be further improved to allow the deformation of soma shapes based on the probability model, which would generate realistic but new and unseen soma shapes for deep learning models to be trained on. In addition, the intensities of simulated somas can be generated through a spline method, further increasing the power of the synthetic data generation pipeline.

The ultimate goal remains to integrate deep learning models into this platform that can use both the semi-automatically labeled and the simulated ground truth data so that the entire process of soma segmentation can be automated with increasing accuracy, in a way that is accessible and cost-effective for all researchers.

REFERENCES

[1] D. S. Richardson, W. Guan, K. Matsumoto, C. Pan, K. Chung, A. Ertürk, H. R. Ueda, and J. W. Lichtman, "Tissue clearing," *Nature Reviews Methods Primers*, vol. 1, no. 1, p. 84, 2021, publisher: Nature Publishing Group UK London.

[2] E. H. K. Stelzer, F. Strobl, B.-J. Chang, F. Preusser, S. Preibisch, K. McDole, and R. Fiolka, "Light sheet fluorescence microscopy," *Nature Reviews Methods Primers*, vol. 1, no. 1, p. 73, Nov. 2021. [Online]. Available: https://doi.org/10.1038/s43586-021-00069-4

[3] L. Ma, Q. Zhong, Y. Wang, X. Yang, and Q. Du, "A 3D semantic segmentation network for accurate neuronal soma segmentation," *Journal of Innovative Optical Health Sciences*, p. 2450018, 2024, iSBN: 1793-5458 Publisher: World Scientific.

[4] X. Liu, B. Hu, M. Li, W. Huang, Y. Zhang, and Z. Xiong, "A soma segmentation benchmark in full adult fly brain," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7402–7411.

[5] R. Xiao, L. Zhu, J. Liao, X. Wu, H. Gong, J. Huang, P. Li, B. Sheng, and S. Chen, "DDeep3M+: adaptive enhancement powered weakly supervised learning for neuron segmentation," *Neurophotonics*, vol. 10, no. 3, pp. 035 003–035 003, 2023, iSBN: 2329-423X Publisher: Society of Photo-Optical Instrumentation Engineers.

[6] X. Wei, Q. Liu, M. Liu, Y. Wang, and E. Meijering, "3D soma detection in large-scale whole brain images via a two-stage neural network," *IEEE Transactions on Medical Imaging*, vol. 42, no. 1, pp. 148–157, 2022, iSBN: 0278-0062 Publisher: IEEE.

[7] M. Dong, D. Liu, Z. Xiong, C. Yang, X. Chen, Z.-J. Zha, G. Bi, and F. Wu, "3D CNN-based soma segmentation from brain images at single-neuron resolution," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 126–130.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, _eprint: 1505.04597. [Online]. Available: https://arxiv.org/abs/1505.04597

[9] P. A. Santi, "Light sheet fluorescence microscopy: a review," *Journal of Histochemistry & Cytochemistry*, vol. 59, no. 2, pp. 129–138, 2011, publisher: SAGE Publications Sage CA: Los Angeles, CA.

[10] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli, "BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images," *Neuron*, vol. 87, no. 2, pp. 252–256, 2015, publisher: Elsevier.

[11] L. Zhang, L. Huang, Z. Yuan, Y. Hang, Y. Zeng, K. Li, L. Wang, H. Zeng, X. Chen, H. Zhang, J. Xi, D. Chen, Z. Gao, L. Le, J. Chen, W. Ye, L. Liu, Y. Wang, and H. Peng, "Collaborative augmented reconstruction of 3D neuron morphology in mouse and human brains," *Nature Methods*, vol. 21, no. 10, pp. 1936–1946, Oct. 2024. [Online]. Available: https://doi.org/10.1038/s41592-024-02401-8

[12] Y. Liu, S. Jiang, Y. Li, S. Zhao, Z. Yun, Z.-H. Zhao, L. Zhang, G. Wang, X. Chen, L. Manubens-Gil, Y. Hang, Q. Gong, Y. Li, P. Qian, L. Qu, M. Garcia-Forn, W. Wang, S. De Rubeis, Z. Wu, P. Osten, H. Gong, M. Hawrylycz, P. Mitra, H. Dong, Q. Luo, G. A. Ascoli, H. Zeng, L. Liu, and H. Peng, "Neuronal diversity and stereotypy at multiple scales through whole brain morphometry," *Nature Communications*, vol. 15, no. 1, p. 10269, Nov. 2024. [Online]. Available: https://doi.org/10.1038/s41467-024-54745-6

[13] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam, "A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks," *Cytometry Part A*, vol. 56A, no. 1, pp. 23–36, 2003, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.10079. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.10079

[14] Christ, "Segmentation of Medical Image using Clustering and Watershed Algorithms," *American Journal of Applied Sciences*, vol. 8, no. 12, pp. 1349–1352, Dec. 2011. [Online]. Available: http://www.thescipub.com/abstract/10.3844/ajassp.2011.1349.1352

[15] X. Li, Z. Zhou, P. Keller, H. Zeng, T. Liu, and H. Peng, "Interactive exemplar-based segmentation toolkit for biomedical image analysis," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, pp. 168–171.

[16] L. Nichele, V. Persichetti, M. Lucidi, and G. Cincotti, "Quantitative evaluation of ImageJ thresholding algorithms for microbial cell counting," *OSA Continuum*, vol. 3, no. 6, p. 1417, Jun. 2020. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=osac-3-6-1417

[17] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnology*, vol. 28, no. 4, pp. 348–353, Apr. 2010. [Online]. Available: https://doi.org/10.1038/nbt.1612

[18] H. Peng, A. Bria, Z. Zhou, G. Iannello, and F. Long, "Extensible visualization and analysis for multidimensional images using Vaa3D," *Nature protocols*, vol. 9, no. 1, pp. 193–208, 2014, iSBN: 1754-2189 Publisher: Nature Publishing Group UK London.

[19] H. Peng, J. Tang, H. Xiao, A. Bria, J. Zhou, V. Butler, Z. Zhou, P. T. Gonzalez-Bellido, S. W. Oh, J. Chen, A. Mitra, R. W. Tsien, H. Zeng, G. A. Ascoli, G. Iannello, M. Hawrylycz, E. Myers, and F. Long, "Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis," *Nature Communications*, vol. 5, no. 1, p. 4342, Jul. 2014. [Online]. Available: https://doi.org/10.1038/ncomms5342

[20] A. Bria, G. Iannello, L. Onofri, and H. Peng, "TeraFly: real-time three-dimensional visualization and annotation of terabytes of multidimensional volumetric images," *Nature Methods*, vol. 13, no. 3, pp. 192–194, Mar. 2016. [Online]. Available: https://doi.org/10.1038/nmeth.3767

[21] A. Bria, G. Iannello, and H. Peng, "An open-source VAA3D plugin for real-time 3D visualization of terabyte-sized volumetric images," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, pp. 520–523.

[22] J. Liang, L. Deng, S. Chen, Y. Wang, Z. Ruan, and L. Zhang, "Vaa3D-x for cross-platform teravoxel-scale immersive exploration of multidimensional image data," *Bioinformatics*, vol. 39, no. 1, p. btac794, 2023, iSBN: 1367-4811 Publisher: Oxford University Press.

[23] L. Cai, J. Gao, and D. Zhao, "A review of the application of deep learning in medical image classification and segmentation," *Annals of Translational Medicine*, vol. 8, no. 11, pp. 713–713, Jun. 2020. [Online]. Available: http://atm.amegroups.com/article/view/36944/html

[24] Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," 2016, _eprint: 1606.06650. [Online]. Available: https://arxiv.org/abs/1606.06650

[25] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[26] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," 2019, _eprint: 1902.09212. [Online]. Available: https://arxiv.org/abs/1902.09212

[27] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," 2018, _eprint: 1804.03999. [Online]. Available: https://arxiv.org/abs/1804.03999

[28] X. Xiao, S. Lian, Z. Luo, and S. Li, "Weighted Res-UNet for High-Quality Retina Vessel Segmentation," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, 2018, pp. 327–331.

[29] Z.-L. Ni, G.-B. Bian, X.-H. Zhou, Z.-G. Hou, X.-L. Xie, C. Wang, Y.-J. Zhou, R.-Q. Li, and Z. Li, "RAUNet: Residual Attention U-Net for Semantic Segmentation of Cataract Surgical Instruments," 2019, _eprint: 1909.10360. [Online]. Available: https://arxiv.org/abs/1909.10360

[30] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *Journal of Medical Imaging and Radiation Oncology*, vol. 65, no. 5, pp. 545–563, Aug. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/1754-9485.13261

[31] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Dec. 2019. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[32] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image Data Augmentation for Deep Learning: A Survey," 2022. [Online]. Available: https://arxiv.org/abs/2204.08610

[33] J. Corral Acero, E. Zacur, H. Xu, R. Ariga, A. Bueno-Orovio, P. Lamata, and V. Grau, "SMOD - Data Augmentation Based on Statistical Models of Deformation to Enhance Segmentation in 2D Cine Cardiac MRI," in *Functional Imaging and Modeling of the Heart*, Y. Coudière, V. Ozenne, E. Vigmond, and N. Zemzemi, Eds. Cham: Springer International Publishing, 2019, vol. 11504, pp. 361–369. [Online]. Available: http://link.springer.com/10.1007/978-3-030-21949-9_39

[34] D. Karimi, G. Samei, C. Kesch, G. Nir, and S. E. Salcudean, "Prostate segmentation in MRI using a convolutional neural network architecture and training strategy based on statistical shape models," *International Journal of Computer Assisted Radiology and Surgery*, vol. 13, no. 8, pp. 1211–1219, Aug. 2018. [Online]. Available: http://link.springer.com/10.1007/s11548-018-1785-8

[35] Z. Tang, K. Chen, M. Pan, M. Wang, and Z. Song, "An Augmentation Strategy for Medical Image Processing Based on Statistical Shape Model and 3D Thin Plate Spline for Deep Learning," *IEEE Access*, vol. 7, pp. 133 111–133 121, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8835020/

[36] R. Bhalodia, A. Goparaju, T. Sodergren, R. Whitaker, A. Morris, E. Kholmovski, N. Marrouche, J. Cates, and S. Elhabian, "Deep Learning for End-to-End Atrial Fibrillation Recurrence Estimation," Dec. 2018. [Online]. Available: http://www.cinc.org/archives/2018/pdf/CinC2018-160.pdf

[37] "vaa3d_tools/released_plugins/v3d_plugins/cellseg_quickfind at master · Vaa3D/vaa3d_tools." [Online]. Available: https://github.com/Vaa3D/vaa3d_tools/tree/master/released_plugins/v3d_plugins/cellseg_quickfind

[38] B. Charmouti, A. K. Junoh, W. Azman, W. Z. A. Wan Muhamad, M. Mansor, Z. Hasan, and M. Mashor, "Extended Median Filter For Salt and Pepper Noise In Image," *International Journal of Applied Engineering Research*, vol. 12, pp. 12 914–12 918, Jan. 2017.

[39] O. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," 01 2009.

[40] G. Bradski, "The opencv library." *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[41] J. E. Bresenham, "Algorithm for computer control of a digital plotter," in *Seminal graphics: pioneering efforts that shaped the field*, 1998, pp. 1–6.

[42] O. Rainio, J. Hällilä, J. Teuho, and R. Klén, "Methods for estimating full width at half maximum," *Signal, Image and Video Processing*, vol. 19, no. 4, p. 289, Feb. 2025. [Online]. Available: https://doi.org/10.1007/s11760-025-03820-6

[43] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[44] M. Abdolhoseini, M. G. Kluge, F. R. Walker, and S. J. Johnson, "Neuron Image Synthesizer Via Gaussian Mixture Model and Perlin Noise," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 530–533.

[45] X. Ruan, M. Mueller, G. Liu, F. Görlitz, T.-M. Fu, D. E. Milkie, J. L. Lillvis, A. Kuhn, J. Gan Chong, J. L. Hong, C. Y. A. Herr, W. Hercule, M. Nienhaus, A. N. Killilea, E. Betzig, and S. Upadhyayula, "Image processing tools for petabyte-scale light sheet microscopy data," *Nature Methods*, vol. 21, no. 12, pp. 2342–2352, Dec. 2024. [Online]. Available: https://doi.org/10.1038/s41592-024-02475-4

[46] G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *Annals of Mathematical Statistics*, vol. 29, pp. 610–611, 1958. [Online]. Available: https://api.semanticscholar.org/CorpusID:119971394

[47] C. Karpfinger, "The QR-Decomposition of a Matrix," in *Calculus and Linear Algebra in Recipes: Terms, phrases and numerous examples in short learning units*, C. Karpfinger, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, pp. 189–199. [Online]. Available: https://doi.org/10.1007/978-3-662-65458-3_19

[48] G. Guennebaud, B. Jacob, and others, "Eigen v3," 2010. [Online]. Available: http://eigen.tuxfamily.org