SKOOTS: Skeleton oriented object segmentation for mitochondria

Christopher J Buswinka[1,2,3], Hidetomi Nitta[1], Richard T. Osgood[1,2], Artur A. Indzhykulian[1,2]

1 - Eaton Peabody Laboratories, Mass Eye and Ear, Boston, MA, USA.
2 - Department of Otolaryngology, Head and Neck Surgery, Harvard Medical School, Boston, MA, USA.
3 - Speech and Hearing Biosciences and Technology graduate program, Harvard University, Cambridge, MA, USA.

ABSTRACT

The segmentation of individual instances of mitochondria from imaging datasets is informative, yet time-consuming to do by hand, sparking interest in developing automated algorithms using deep neural networks. Existing solutions for various segmentation tasks are largely optimized for one of two types of biomedical imaging: high resolution three-dimensional (whole neuron segmentation in volumetric electron microscopy datasets) or two-dimensional low resolution (whole cell segmentation of light microscopy images). The former requires consistently predictable boundaries to segment large structures, while the latter is boundary invariant but struggles with segmentation of large 3D objects without downscaling. Mitochondria in whole cell 3D EM datasets often occupy the challenging middle ground: large with ambiguous borders, limiting accuracy with existing tools. To rectify this, we have developed **sk**eleton **o**riented **o**bjec**t** **s**egmentation (SKOOTS); a new segmentation approach which efficiently handles large, densely packed mitochondria. We show that SKOOTS can accurately, and efficiently, segment 3D mitochondria in previously difficult situations. Furthermore, we will release a new, manually annotated, 3D mitochondria segmentation dataset. Finally, we show this approach can be extended to segment objects in 3D light microscopy datasets. These results bridge the gap between existing segmentation approaches and increases the accessibility for three-dimensional biomedical image analysis.

INTRODUCTION

The dynamics of mitochondria, size and number, fission and fusion, may be important to the pathology of many diseases[1]. Mitochondria dysfunction has been implicated in Alzheimer's[2,3], Huntington's[4], Parkinson's[5], cardiovascular diseases[6], cancer, auditory neuropathy[7,8], and many other conditions. Mitochondria are abundant in the pre-synapse[9,10], help regulate Ca^{2+} dynamics[11-13], and function in apoptotic signaling cascades[14-17]. In metabolically active cells they often occupy a significant percentage of total cellular volume[18], yet their relatively small size and variable morphologies make resolving single instances, as well as automated image analysis and quantification, very difficult[19]. Recent advances in light microscopy can allow for remote, individual mitochondria to be differentiated[20], however the 3D electron microscopy techniques, such as focused ion beam scanning electron microscopy (FIB-SEM)[21] or serial blockface SEM, enable the visualization of mitochondria morphology at nanometer scale and are more suitable for studying, sometimes subtle, morphological changes. These techniques, when imaging entire or multiple cells, often generate huge volumes of data over large spatial areas[22]. Imaging potentially many-thousands of mitochondria in great detail poses a great analysis challenge: there are simply too many to analyze by hand. To properly analyze their volume, morphology, as well as spatial distributions within a cell, and they change with experimental conditions, one must accurately segment each instance of mitochondria.

Yet mitochondria are difficult to segment[19]. Connectomics – the analysis of the connections of neurons through high resolution 3D electron microscopy – necessitates segmentation with near perfect accuracy over huge areas[23] and has led to the development of three-dimensional algorithms, most of which predict via affinity graphs[24-26], to enable segmentation based on object boundaries[27-31]. Boundaries are inferred by a predicted graph of affinities, i.e. a graph of the connections between 3D pixels, known as voxels[24]. If two adjacent voxels are a part of the same underlying object (a single mitochondrion, neuron, or other structure), their affinities are high, while if two adjacent voxels belong to different objects, their affinities are low. Through this affinity graph, sub-voxel boundaries can be inferred and

segmented by watershed segmentation, which groups the connected components. This, and other similar approaches[32], are fast, efficient, and accurate when affinities are correctly predicted. However, individual mitochondria may be touching, have some regions with unclear boundaries, leading to poor affinity graph prediction resulting in under-segmentation[30] (i.e. where several mitochondria are merged into a single segmentation mask). Ambiguous boundaries between densely packed objects may occur independent of imaging modality. This is a particular challenge for segmenting densely packed cells with cytoplasmic labels visualized by light microscopy. Cellpose[33], later Omnipose[34], the two most popular instance segmentation approaches for 2D light microscopy data, solve this with a different approach: by predicting optical flows. Optical flows are vectors which push pixels in a useful direction. In practice, when segmenting dense cells with ambiguous borders, pixels of different cells flow away from their border.

Optical flows are only useful when the pixels flow somewhere. For Cellpose, pixels flow against a diffusion gradient, toward the cell's center of mass. In some cases, the center of mass may not lie within the cell, as in curled bacteria[34], leading to the core innovation of Omnipose: flowing up the gradient of an Eikonal distance function[35], towards the local center of mass. Applying these flows creates clusters of pixels for each instance, which must be identified with a clustering algorithm (usually DBSCAN[36]). These approaches are designed, and well suited, for two-dimensional instance segmentation of biomedical imaging data. While existing implementations work in 3D, they are in practice limited by the clustering algorithm, which scales poorly in memory and slows exponentially in time [$O(N^2)$][37]. Omnipose, for example, is purportedly limited to processing a 78(x) 78(y) 78(z) voxel cube on the memory of a 12 GB consumer-grade graphics card[34]. Therefore, to segment anything larger than 78 px wide would require either downscaling of high-resolution data, excessive and inefficient tiling of a large dataset, or both. This is not suitable for mitochondria segmentation within high resolution 3D EM datasets. These limitations are less in 2D, and therefore may be mitigated by evaluating 2D segmentation models on multiple planes of a 3D image, then reconstructing a 3D mask. This is a common workaround, employed by Cellpose and Empanada[38], a novel mitochondria segmentation platform, however reconstruction introduces and additional point of failure for segmentation errors.

There are additional challenges that also limit the development and implementation of a 3D machine learning segmentation approach. Training data are time consuming to generate, and existing datasets are unlikely to be suitable for a specific analysis task. Furthermore, the adoption of any segmentation analysis is limited by speed and hardware. To be accessible, 3D segmentation must be not just accurate but also efficient. Approaches needing a supercomputer will be used less than one suitable for a personal computer or laptop. Approaches that run slowly will be used less than one that executes instantly. In this work, we take strides to alleviate these limitations.

In this study, we present a new approach for instance segmentation of mitochondria within high-resolution 3D EM datasets: **s**keleton **o**riented **o**bjec**t** **s**egmentation (SKOOTS). We show that SKOOTS is fast, efficient, and accurate even for cases with densely packed mitochondria of various morphologies. We created a dataset of manually annotated mitochondria which was used to train and validate our algorithm. We trained SKOOTS on a variant of the U-Net deep learning architecture[39] with modifications allowing for better accuracy per parameter over convention[40]. We compare SKOOTS to several other existing mitochondria segmentation approaches. Finally, we show that this algorithm can also be applied to developing 3D segmentation solutions for objects within high resolution light microscopy datasets.

RESULTS

Skeleton embedding allows for segmentation of mitochondria

Modern graphics cards quickly evaluate neural networks; however, many segmentation approaches require the usage of further algorithms to translate these predictions into segmentation masks. These algorithms are the limiting factor in 3D segmentation speed. Cellpose and Omnipose, for example, rely on clustering algorithms to assign unique ID labels. The most common, DBSCAN, has a time complexity of $O(N^2)$ [37] and therefore is exponentially slower when adding a third dimension. Segmenting a 100(x) 100(y) 100(z) voxel image cube processed with DBSCAN is 100,000x slower than segmenting a single 100(x) 100(y) pixel image square. In contrast, boundary detection networks use a version of the watershed algorithm on predicted affinity graphs to assign ID labels. Affinity graphs are the connections between voxels within the same object; adjacent voxels of two different objects do not connect, forming an implicit sub-voxel boundary between them. When all affinities are correctly predicted, the collection of sub-voxel boundaries perfectly delineates

the object. This scales linearly in time and is therefore faster for 3D processing[41]. However, small errors in the affinity graph, causing a "hole" to open between two boundaries, can have a dramatic effect, leading to under-segmentation and poor segmentation performance.

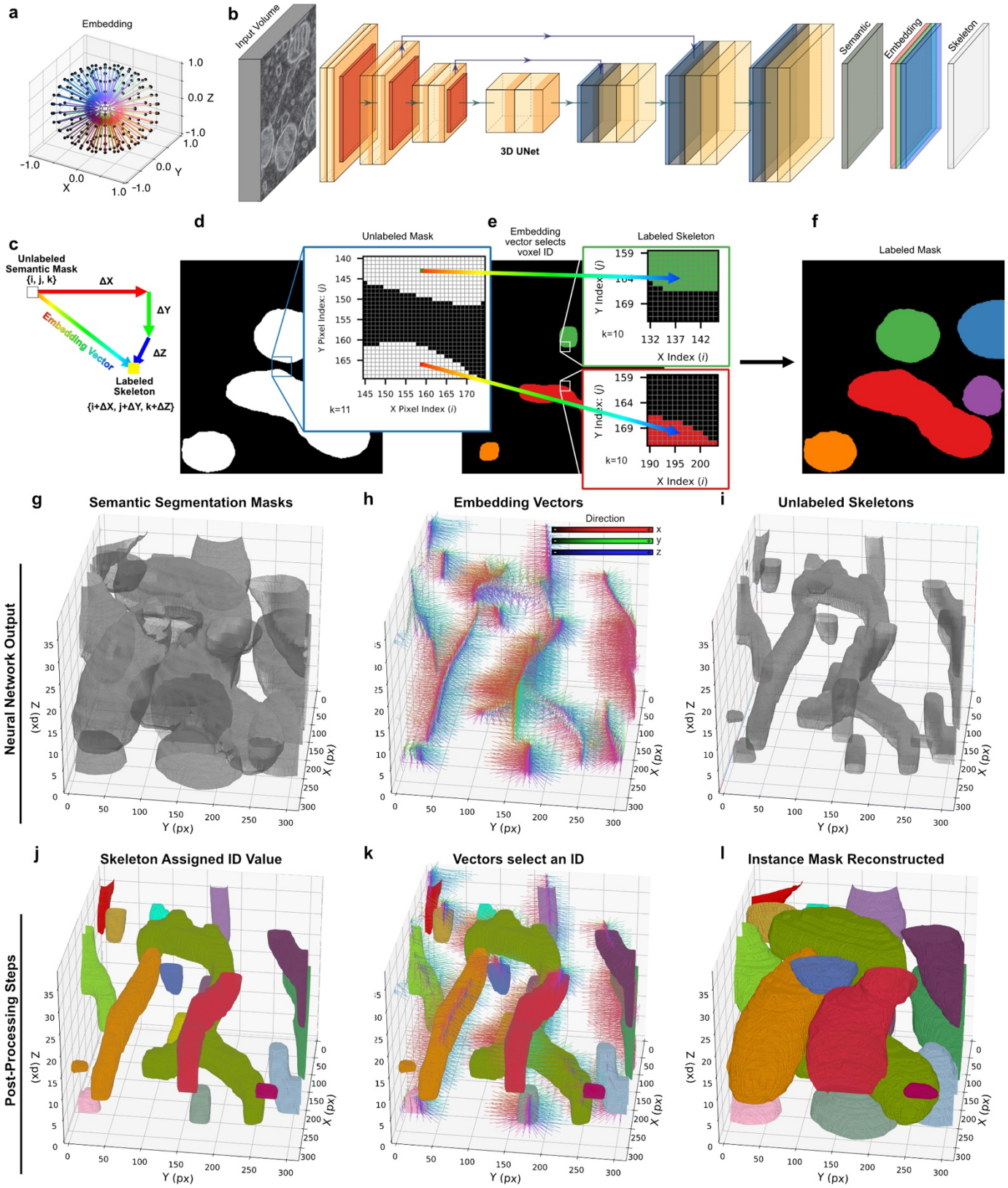

Figure 1. Overview of instance segmentation procedure via neural network skeleton embedding. (**a**) A schematic representation of spatial embedding. Voxels of a sphere in 3D space (black dots) are associated with an object by vectors (color-coded based on their orientation) pointing from their spatial location to the object's local mean (grey). This principle is utilized for mitochondria

segmentation. (**b**) An input volume is passed through a trained 3D UNet neural network model which predicts a semantic segmentation mask, unlabeled skeletons, and 3D skeleton embedding vectors. (**c**) Each unlabeled voxel at location ijk with embedding vector $V_{ijk} = [\Delta X, \Delta Y, \Delta Z]$ points to a skeleton at position $i + \Delta X, j + \Delta Y, k + \Delta Z$. (**d-f**) Simplified 2D representation of model's key steps. (**d**) Each voxel of a mitochondrion semantic mask is assigned a 3D embedding vector, predicted by the neural network, which points to an un-labeled skeleton. (**e**) All predicted skeletons are then labeled using an efficient flood fill function. (**f**) An unlabeled voxel therefore points, via the vector, to a labeled skeleton, thus skeleton's ID can be used to assign that voxel a label. (**g-i**) 3D representations of the model's key steps. A pretrained deep-learning model directly infers the following, rendered in 3D for illustration purposes: (**g**) semantic segmentation mask, (**h**) embedding vectors with color-coded orientation, and (**i**) unlabeled skeletons. Using the deep neural network outputs, the following steps generate instance segmentation masks. (**j**) A flood fill function assigns each skeleton a unique ID. (**k**) The embedding vectors project onto labeled skeletons, assigning each voxel an ID. (**l**) 3D render of fully labeled mitochondria instance masks.

Densely packed mitochondria may have unclear boundaries, leading to unreliable affinity graph prediction, in turn resulting in poor segmentation outcomes. In whole cell FIB-SEM datasets they are also commonly imaged at too high a resolution for existing 3D optical flow techniques. Our approach of **sk**eleton **o**riented **o**bjec**t s**egmentation (SKOOTS) solves these problems by recognizing that while mitochondria may be densely packed, with their membranes in contact with other neighboring mitochondria, their skeletons remain distinct. Object skeletons are defined as a connected local center of mass[42]. Intuitively, when one shakes hands with another, skin comes in contact but the bones within each finger do not. With the same principal, the skeletons of neighboring mitochondria do not contact regardless of packing density. (The principal steps of SKOOTS segmentation are outlined in **figure 1**). This solves a critical problem of some optical flow-based methods: knowing where to flow if the model cannot see the entire object. In contrast to Omnipose, which made a similar observation, we explicitly predict these skeletons as a volume. Hence, their ID labels may be assigned by a flood fill[43], which is more computationally efficient than the clustering approaches often used for this task. Once ID values are assigned to skeletons, they must be associated with the voxels that make up the rest of the object. We do this by predicting a set of spatial embedding vectors (**figure 1a**), like Omnipose, which point from any voxel of a mitochondria to its skeleton. However, instead of inferring clusters by iteratively flowing up the gradient of an Eikonal distance function, we simply predict these vectors outright. Therefore, each unlabeled voxel has a predicted embedding vector which "points" to a labeled skeleton, and is ultimately assigned the same ID. This direct assignment eliminates the need for an Euler integration employed in Omnipose and contributes to SKOOTS' increased segmentation speed and memory efficiency.

Following successful implementation of these principles (**figure 1b**), SKOOTS, when tasked to segment mitochondria within a 3D FIB-SEM dataset approaches the speed of affinity-based segmentation while maintaining the boundary invariance of flow-based methods. SKOOTS, in part, works by predicting an optical flow in 3D, sometimes called a spatial embedding, using a deep neural network (**figure 1a-c**) for each voxel of a mitochondrion semantic segmentation mask (**figure 1d**). Rather than "flowing" to a center of mass, or point of maximum distance, these vectors embed a voxel to an explicitly predicted, and uniquely labeled, skeleton (**figure 1e**). Voxels embedded within a labeled skeleton are simply assigned the skeleton's label, while reversing the embedding results in an instance mask (**figure 1f**). Specifically, SKOOTS's steps are outlined as follows. First, a pretrained neural network predicts a semantic segmentation mask of mitochondria (**figure 1g**), a set of 3D skeleton embedding vectors (**figure 1h**), and an unlabeled segmentation mask of distinct mitochondria skeletons (**figure 1i**). Embedding vectors are predicted for each voxel in a volume and represented as a 3D RGB image, where the red, blue, and green values reflect the embedding strength in X, Y, and Z, respectively. Next, a flood fill algorithm is applied to the unlabeled skeletons, assigning an ID to each instance (**figure 1j**). Following this step, skeleton embedding vectors at each voxel point to a labeled skeleton with a unique ID (**figure 1k**) and therefore are assigned that ID. From these assignments, an instance mask is reconstructed (**figure 1l**). For large images, this process may be parallelized, further improving efficiency on large datasets. Movies showing each step in 3D are available in the supplementary materials.

Most datasets generated using 3D EM or fluorescence imaging do not have isotropic voxels: the pixel size in X-Y is often smaller than the Z step. Embedding vectors are predicted by SCOOTS in the range of -1 to 1 for all spatial dimensions. By scaling these vectors proportionally to the anisotropy of the dataset, SKOOTS can segment both isotropic and anisotropic datasets generated using FIB-SEM, serial blockface SEM, confocal microscopy, or any other 3D imaging techniques.

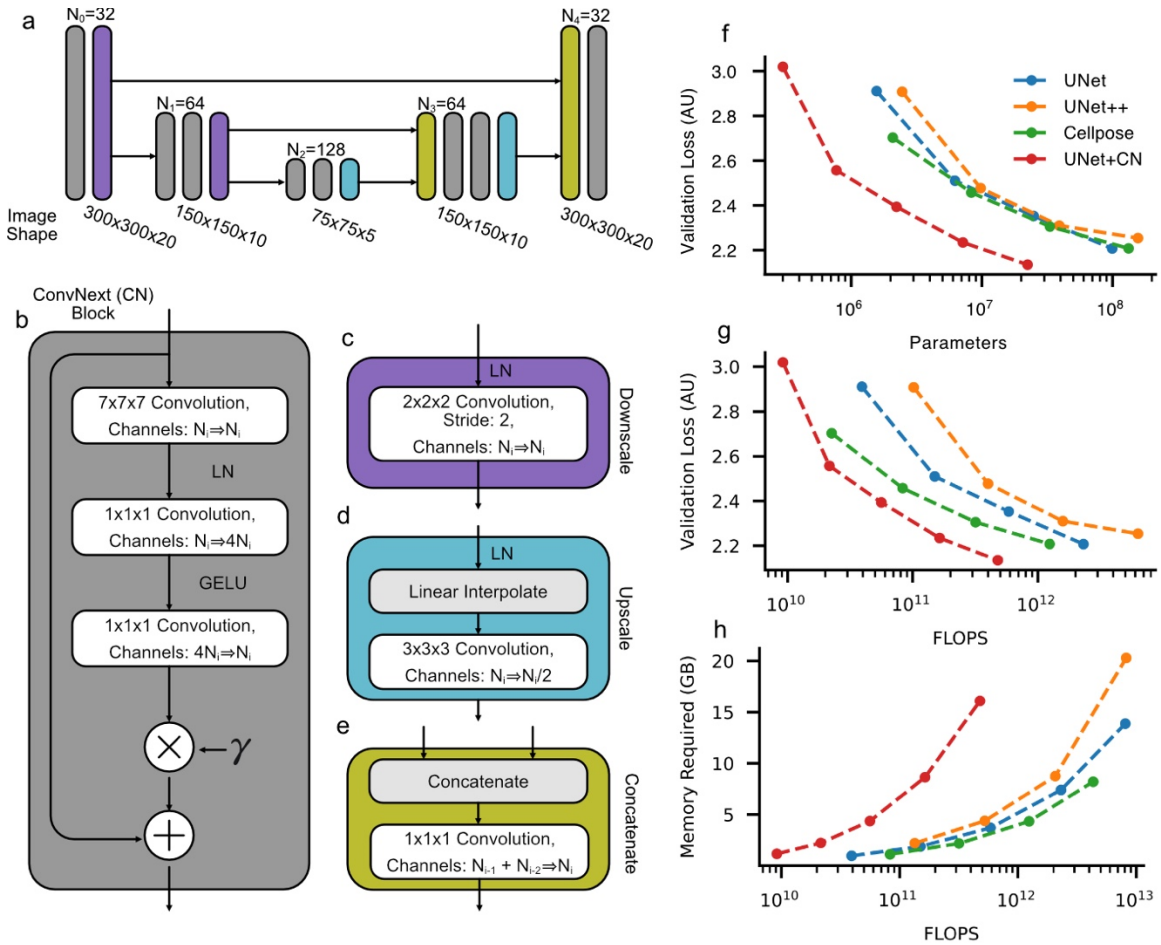

Figure 2. UNet Architecture with ConvNext computational blocks offers superior accuracy per floating-point operations/second (FLOPS) in a semantic segmentation task. (**a**) Structure of a UNet-like model, composed of four major neural-network computational blocks, summarized in panels b-e, and color-coded across a-e for clarity. (**b**) a ConvNext residual block replaces the standard two stacked convolutions of UNet with a single convolution with increased kernel size (7x7x7) followed by two pointwise convolutions and a learnable scaling parameter (1/2×), (**c**) a downscale block composed by a 3D convolutional kernel with a 2x2x2 shape and a stride of 2, (**d**) an upscale block achieved by linear interpolation followed by convolution, and (**e**) a concatenation block. (**f-h**) Upon direct comparison, this architecture (**Red**) achieves superior performance semantic segmentation compared to currently reported best alternative architectures (UNet, UNet++ and Cellpose), with (**f**) a lower number of parameters, (**g**) a lower number of FLOPS, however, needs slightly more (**h**) video memory to process identically sized volumes when compared to alternatives**.**

A highly efficient UNet-like architecture for segmentation is more computationally efficient than standard 3D UNet

For many cases, deep neural networks are limited by computational resources. Accuracy tends to change with the number of parameters of a model, the specific architecture (structure and operations within the model), as well as the number of floating-point mathematical operations (FLOPS). As SKOOTS requires embedding vectors and skeletons to be well defined for ultimate segmentation accuracy to be high, we sought to develop a more efficient neural network architecture, aiming to maintain high accuracy with less resources. To do this, we took advantage of recent advances in other areas of computer vision. ConvNext, a neural network architecture for 2D object detection, achieved state-of-the-art performance by applying a system of best practices to convolutional neural network design[40]. We applied the same techniques to the standard UNet architecture (**figure 2a-e**), trained, and compared the resulting model against other published architectures for biomedical image segmentation. We trained each model on a semantic segmentation task (for speed), varying the number of parameters (and therefore FLOPS), and compared accuracy (**figure 2f,g**), and graphic memory requirements (**figure 2h**). By replacing the double convolution at each stage of a UNet with these ConvNext blocks, we achieve superior segmentation performance compared to competing segmentation architectures. This architecture improves accessibility of 3D segmentation. We employed this UNet + ConvNext architecture for all

subsequent segmentation task, and when applying the SKOOTS algorithm, it can process a 300(x) 300(y) 20(z) voxel volume in 5.6 GB of video memory (data not shown).

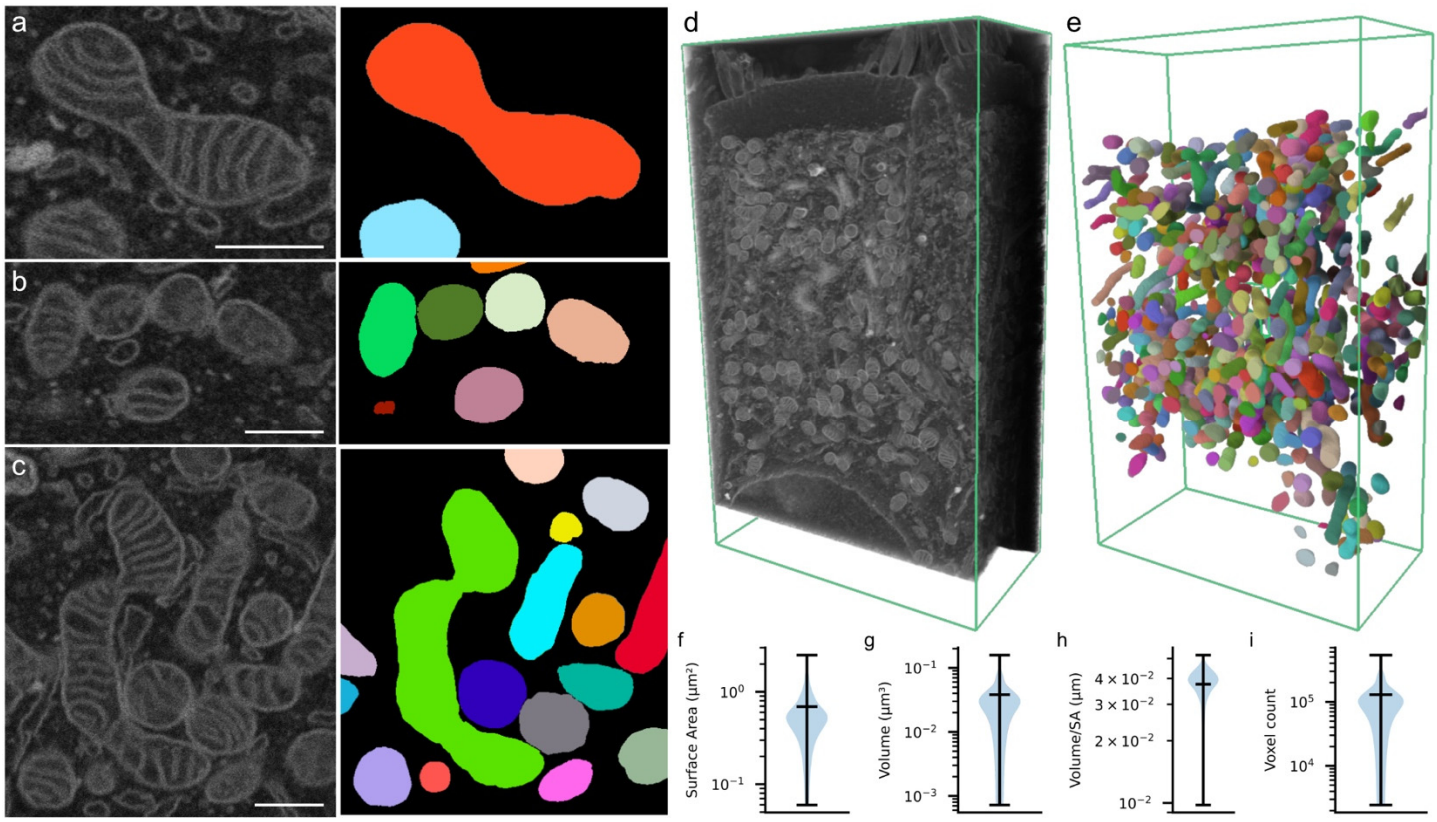

Figure 3. Manually annotated FIB-SEM mitochondria instance segmentation training dataset of an early post-natal murine outer hair cell. (**a**) Two-dimensional slice of exemplar mitochondria (left) and associated segmentation mask (right). Distinct mitochondria are represented by unique colors. (**b**) Exemplar image and an accompanying mitochondria mask from a region of densely packed mitochondria with ambiguous borders between outer mitochondria membranes. (**c**) Exemplar mitochondrion (lime green) with atypical morphology. (**d**) A three-dimensional render of the entire training dataset. (**e**) Three-dimensional render of ~700 manually annotated ground truth mitochondria instance masks. From this dataset of mitochondria we calculated distributions of: (**f**) surface area, (**g**) volume, (**h**) volume to surface area ratio, and (**i**) total voxel count per mitochondrion. All scale bars are 500 nm. Data are presented as Mean, 15% and 85% quantile.

A new annotated dataset for FIB-SEM mitochondria segmentation

A major limitation in the development of accurate 3D segmentation approaches is the limited number of open datasets by which to train with. Some high quality, 3D segmentation datasets exist[19], however as morphologies can be highly heterogenous between cell types, often training on such datasets can be limited in its usefulness.

We have generated a high-resolution FIB-SEM dataset of murine cochlear sensory hair cells with a volume of 23.4 µm, 18.5 µm, and 11.1 µm in X, Y, and Z respectively, with a pixel size of 3.82 nm in X and Y, and 20 nm milling step, leading to an anisotropy ratio of ~1/5, for an image dimension of 6194(x) 4849(y) 553(z). The mitochondria contained within the image have varied morphologies, sizes, and packing densities. For training and validation, we manually annotated 722 mitochondria from a single outer hair cell in an image area of 1968(x) 3528(y) 201(z). We employed a human in the loop annotation paradigm in which we corrected the instance masks produced by a semi-supervised instance segmentation approach. Exemplar mitochondria are presented in **figure 3**. The average volume of these mitochondria is 0.038±0.023 μm^3 with a surface area of 0.691±0.381 μm^2. Special attention to accuracy was provided to densely packed mitochondria, as these are typically the most difficult to segment. We split the annotated dataset into a training set and a validation set by z-slice. The first 151 slices were used for training, while the remaining 50 were used for validation. The validation set contained 198 manually annotated mitochondria, representing 27% of manually annotated instances and 24.8% of volume used for annotation.

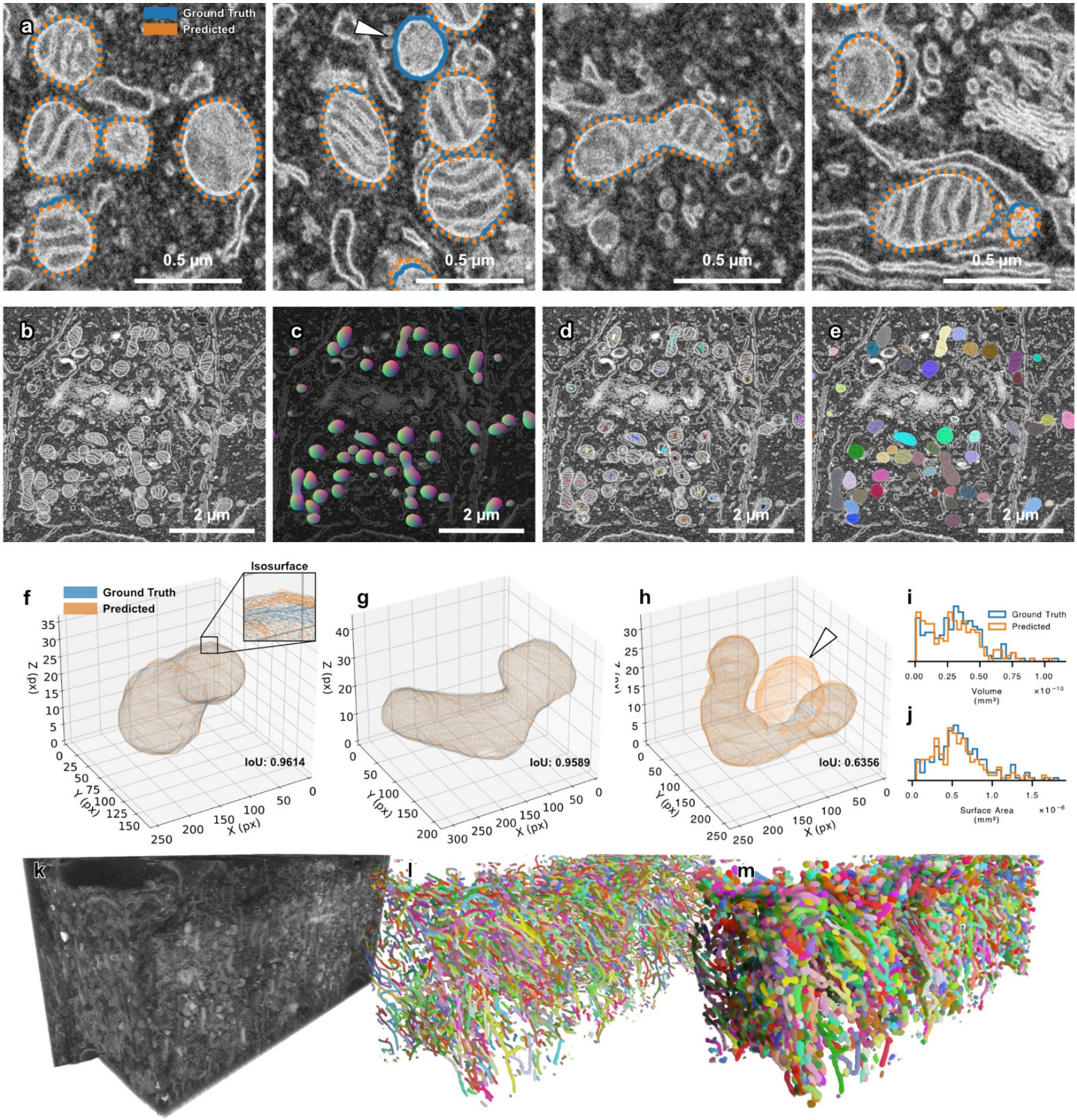

Figure 4. SKOOTS enables accurate instance segmentation on large-scale datasets. (**a**) 2D slice of exemplar segmentation result. Predicted segmentation outlines (orange dashed) are overlayed on ground truth segmentation outlines (blue). A missed mitochondrion instance is denoted by the white arrowhead. (**b-e**) Visual representation of SKOOTS workflow: (**b**) Single FIB-SEM slice, (**c**) Visualized slice of 3D skeleton embedding vectors, (**d**) labeled skeletons, (**e**) the resulting segmentation masks. Different colors denote individual instance IDs. (**f**) 3D mesh render of an exemplar ground truth (blue) and predicted (orange) segmentation mask isosurfaces. (**g**) SKOOTS accurately segments mitochondria with varied morphology. (**h**) An example of an under-segmentation error. (**i**) Histogram of mitochondria volume calculated from manually annotated (blue) and predicted (orange) segmentation masks. (**j**) Histogram of mitochondria surface area calculated from manually annotated (blue) and predicted (orange) segmentation masks. (**k**) 3D render of FIB-SEM input volume. (**l**) 3D render of predicted skeleton instances. (**m**) 3D Render of mitochondria instance segmentation masks.

SKOOTS enables accurate 3D segmentation on a large scale in FIB-SEM volumes

Using our mitochondria dataset and neural network architecture, we trained a deep learning model to segment mitochondria using SKOOTS. When compared to our validation set of 198 manually annotated mitochondria, SKOOTS achieves high segmentation accuracy (**figure 4a**) across a variety of mitochondrial sizes and morphologies. A single 2D slice of the predicted skeletons and resulting segmentation masks is presented in **figure 4b-e**. While most mitochondria were segmented accurately (exemplar 3D renders in **figure 4f, 4g**), errors in skeleton prediction can lead to segmentation errors (**figure 4h**). However, predicted segmentation masks are sufficient to predict identical distributions of morphological statistics when compared to the manually annotated dataset (**figure 4i, 4j**). To demonstrate applicability on large datasets, we applied SKOOTS to our large, multi-cellular FIB-SEM dataset (**figure 4k-m**)**.** Using consumer-grade hardware, SKOOTS segmented 6897 mitochondria within a 6194(x) 4849(y)553(z) px 3D EM dataset at a resolution of 3.82 nm/px (x and y) in 25.8 minutes, segmenting an average of 267 mitochondria per minute.

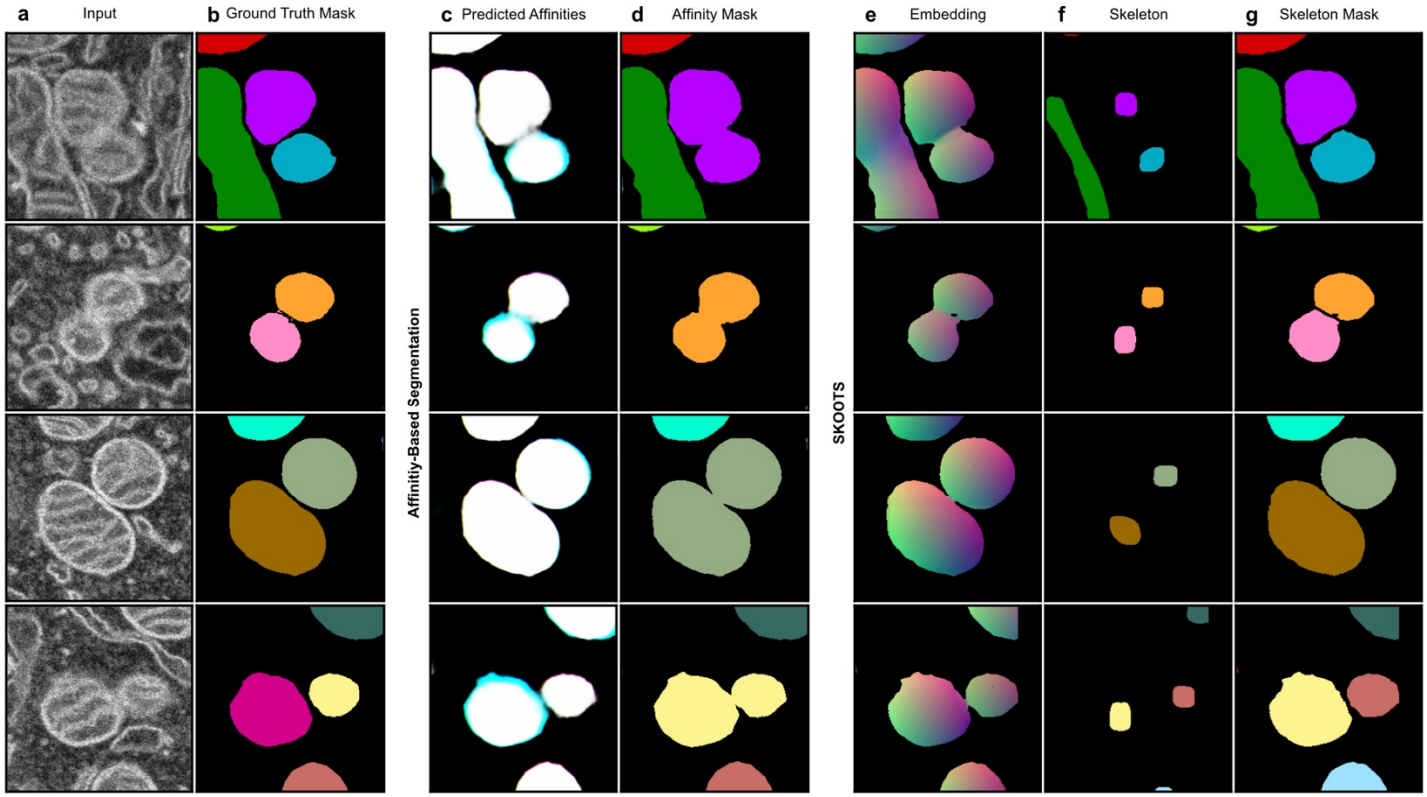

Figure 5. SKOOTS outperforms affinity graph segmentation in challenging to segment mitochondria. (**a**) Images denoting challenging instances of mitochondria segmentation, (**b**) with associated ground truth masks, are input into two segmentation pipelines for segmentation mask generation: (**c-d**) affinity-based segmentation and (**e-g**) SKOOTS. (**c**) Predicted affinity graphs were agglomerated to form (**d**) instance masks, however weak boundary prediction led to under-segmentation. SKOOTS predicts (**e**) embedding vectors and (**f**) skeletons leading to better (**g**) segmentation outcomes.

SKOOTS enables more accurate mitochondria segmentation when compared to other approaches

To validate SKOOTS, we also trained an affinity-based segmentation model using an identical architecture on the same dataset using a common set of parameters to facilitate unbiased comparison. We applied the resulting model on our manually annotated validation dataset, calculated instance masks, and compared segmentation results **(figure 5)**. When evaluating the outcome of the affinity model, we observed poor affinity graph predictions at the contact points between different, touching mitochondria, leading to multiple mitochondria being labeled as one, resulting in under-segmentation errors (**figure 5c, 5d**). By predicting skeletons for each individual object, SKOOTS was able to correctly separate each mitochondrion, resulting in fewer under segmentation errors compared to the affinity-based approach (1.04% vs 1.17% respectively **figure 5e-g**).

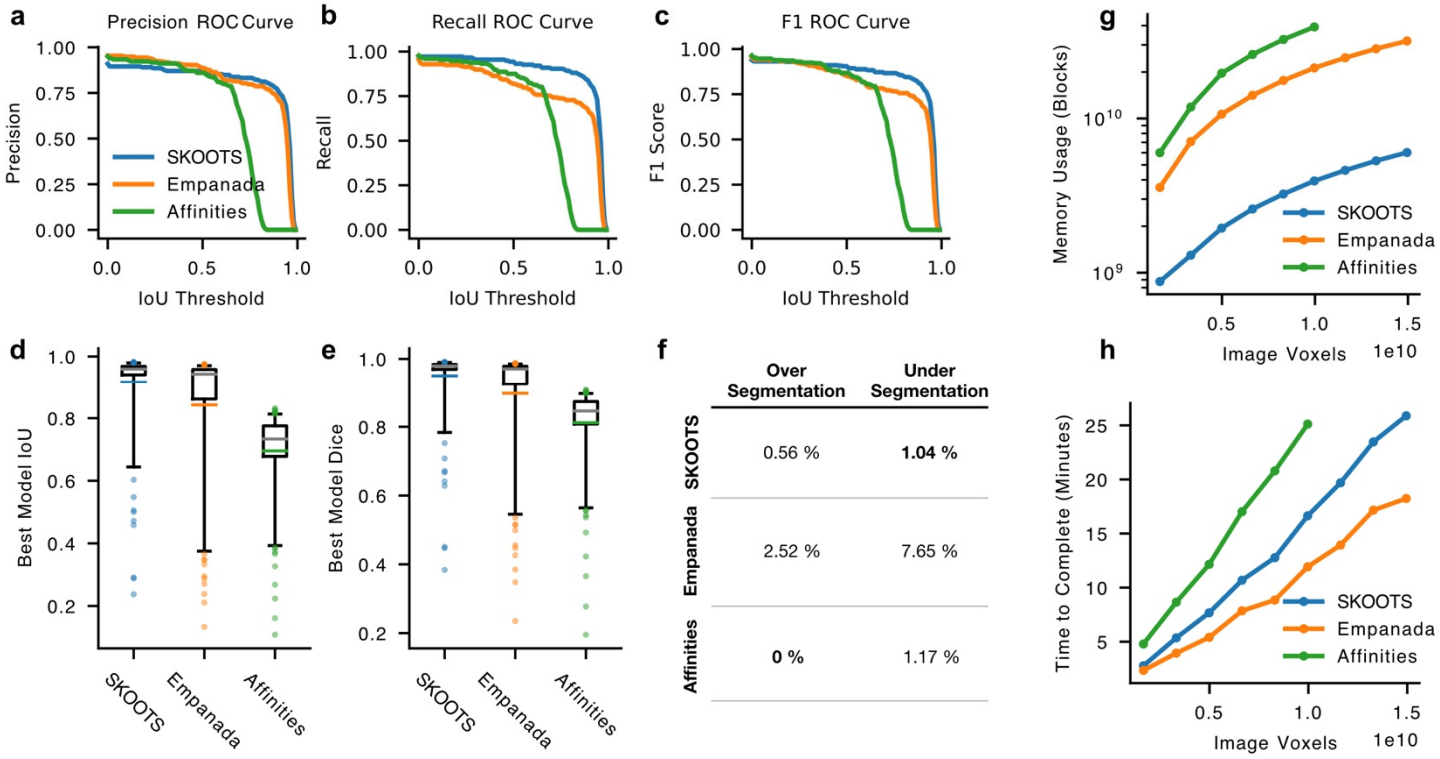

Figure 6 – SKOOTS is accurate, fast, and efficient at segmenting mitochondria compared to other approaches. Mitochondria in our validation FIB-SEM dataset were segmented with SKOOTS (**blue**), Empanada (**orange**), and with Affinity-based (**green**) segmentation approaches and compared for accuracy against the manually annotated ground-truth segmentation masks. The overall performance was assessed by calculating (**a**) precision, (**b**) recall, and (**c**) F1 score for each approach at a variety of rejection thresholds to create region operating characteristic (ROC) curves for each metric. The average intersection over union (IoU, **d**) and Dice (**e**) score were calculated for all correctly identified mitochondria; colored horizontal line represents the mean value, while the grey line reports the median value. The number of over- and under-segmentation errors generated by each approach were calculated (**f**); with best performers shown in bold. Finally, the total memory (**g**), and time (**h**) required to segment a volume of varying total size for each approach were also calculated. Each approach was trained for a similar amount of time using an identical dataset. SKOOTS and Affinity segmentation used identical model architectures and model evaluation scripts, while Empanada was trained and evaluated using its Napari plugin.

We also fine-tuned Empanada[38], a pseudo-3D mitochondria segmentation pipeline that utilizes optical flow in 2D before 3D reconstruction, on our dataset and similarly evaluated its performance on our validation dataset. It is worth noting, unlike for comparisons to the affinity-based approach, Empanada was pretrained on other data and not implemented with the same deep-learning architecture. For each method, SKOOTS, affinities, and Empanada, we calculated detection and segmentation accuracy metrics (**figure 6**). SKOOTS achieved highest precision, recall and F1 scores for mitochondria detection (**figure 6a, 6b, 6c**) as well as best average intersection over union (IoU) and Dice metrics (**figure 6d, 6e**), suggesting superior segmentation accuracy. Furthermore, SKOOTS offered the lowest rate of under-segmentation errors (**figure 6f**). Finally, we found that SKOOTS requires less computer memory to segment when compared to Empanada and Affinities (**figure 6g**). While faster than Affinity-based segmentation, SCOOTS was slower than Empanada, taking 25.8 min compared to 18.2 min for 1.49e10 voxels in this validation task (**figure 6h**), perhaps due to better acceleration of 2D deep-learning operations as compared to 3D.

SKOOTS can be applied for efficient segmentation of fluorescence light microscopy datasets

While SKOOTS was originally designed for segmentation of mitochondria in FIB-SEM volumes, it is a general algorithm for high-resolution 3D segmentation and can be applied to other datatypes across different imaging modalities. To demonstrate this, we trained and applied SKOOTS to segment a 1000(x) 1000(y) 60(x) voxel confocal microscopy Z stack of neonatal auditory hair cells immunolabeled for cytoplasmic hair cell marker, MYO7A (**figure 7a**). These cells are densely packed in an epithelium in four rows, and when visualized with a cytoplasmic label often lack clear borders, yet SKOOTS enables prediction of accurate 3D segmentation masks. We further trained SKOOTS to segment another

fluorescence dataset of the lateral root primordia of Arabidopsis thaliana, stained with DAPI and previously deposited by authors of the original study[44](**figure 7b**). These results demonstrate SKOOTS can be an accurate and efficient tool for instance segmentation of objects in a range of three-dimensional imaging datasets.

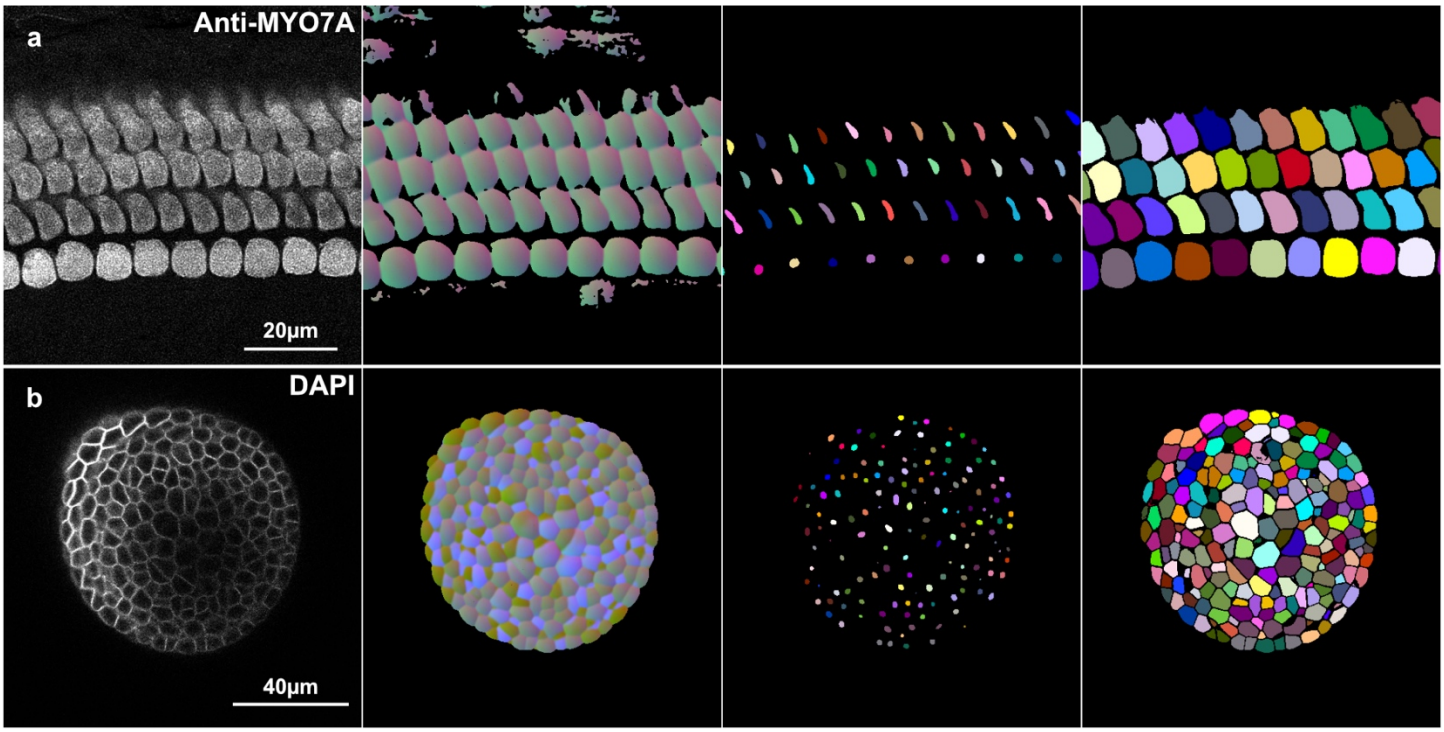

Figure 7. SKOOTS can segment cells within fluorescence microscopy datasets. SKOOTS is not limited to electron microscopy volumes. We successfully trained, and applied, SKOOTS to segment (**a**) auditory hair cells, stained with an anti-MYO7A immune label, as well as (**b**) cells of the lateral root primordia cells of Arabidopsis thaliana; cell walls stained with DAPI. Both samples were imaged with confocal microscopy. The input image, predicted vectors, skeletons, and resulting segmentation mask are displayed from left to right, respectively.

DISCUSION

Here, we present SKOOTS: a 3D deep learning approach for segmenting mitochondria. Based on explicitly predicting skeletons, SKOOTS is fast, memory efficient, and accurate. SCOOTS was developed to facilitate the segmentation of objects too large to be predicted by native 3D optical flow approaches, yet tightly packed with unreliable features limiting accuracy for boundary detection approaches. While designed for mitochondria segmentation within EM datasets, it is not limited by microscopy modality or object type, and represents a significant advancement for 3D biomedical image segmentation.

This work was motivated by the desire to study mitochondrial population dynamics in our own lab. The auditory hair cells are post-mitotic cells with high energetic demands. Hence, mitochondria are abundant and densely packed, leading to poor performance with existing 3D electron microscopy segmentation approaches, both semi-automatic and automatic. The semi-supervised approach of manually correcting watershed segmentation following a deep learning-based prediction of semantic masks was accurate (and faster than fully manual annotation) but still impractical for our use case. We found affinity-based segmentation and Emapanda contained too many under-segmentation errors on our datasets. The use of Omnipose optical flow approach would have required substantial down sampling to fit Omnipose's purported 78(x) 78(y) 78(z) voxel size limitation for 3D analysis, leading to an impermissible loss of precision for our uses.

Each pre-existing algorithm is ideally suited for the use case they were designed for, and SKOOTS does supersede either affinity-based or Omnipose segmentation algorithms. Rather, we draw inspiration from both algorithms, offering superior performance in a use case poorly served by either: segmentation of high resolution, densely packed objects with poor delineation. There are many factors which influence a neural network's accuracy, and therefore its

segmentation outcomes. A segmentation approach may appear better suited, not from superiority in design, but due to differences in neural network training, execution, or post-processing workflow. Therefore, we attempted to mitigate these factors, when possible, when comparing SCOOTS to other approaches therefore attributing differences in performance to approach alone. However, without re-implementing each, it is unclear if any differences at all are due to subtle approach-specific optimizations.

SKOOTS succeeds in high resolution instance segmentation where object diameters are large enough to predict a well-defined skeleton. Therefore, there is a functional minimum object diameter in pixels, at which points skeletons will either merge or separate, causing under-segmentation or over-segmentation respectively. In practice, with sufficiently high-resolution images this does not occur. Furthermore, while each operation necessary for SKOOTS segmentation is dimensional agnostic and therefore may be performed in 2D, the speedup of using a flood fill (SKOOTS) over DBSCAN (Omnipose) becomes functionally negligible. Therefore, there is no advantage to SCOOTS over Omnipose for the segmentation of 2D datasets. In 3D, SKOOTS requires significantly less total video memory than Omnipose. However, SKOOTS needs four pieces of information per voxel (embedding strength in x, y, z and a skeleton) to segment objects, compared to only three (affinities in x, y and z) for affinity-based segmentation. This constitutes a 33% increase in computer memory requirements. Our dataset at native resolution and bit-depth is 32.6 GB and therefore a 33% increase in memory requirement was acceptable. This may cease to be the case for much larger datasets without further feature development, such as streaming data directly to the disk.
While we implemented SKOOTS in python due to its flexibility and the robust ecosystem of deep learning tools, python is a slow language. Most of the execution time in our segmentation process is not deep learning prediction, rather, it is spent running conventional algorithms which may have faster implementations in compiled languages such as C++ or Rust. With dedicated implementations in such languages, significant speed increase s may be achievedover our native python code and therefore further increase usability.

The FIB-SEM dataset we generated was sufficient to train an accurate segmentation model for our dataset. However, variations in sample, preparation, and microscope parameters, affecting resolution and signal-to-noise ratio, may lead to diminished performance of our pretrained model in other cases. Mitochondria have highly variable morphologies; it is therefore reasonable to assume that the mitochondria in auditory hair cells may not be fully representative to other tissues. Furthermore, our dataset does not contain any mitochondria in obvious disease states, such as extreme swelling. Therefore this model may be unsuitable for the detection of pathologic mitochondria without further dedicated training using such datasets.

We have made the SKOOTS source code, model source code, and all pretrained models publicly available. The code is implemented as a library which can easily be adapted into any segmentation pipeline. Furthermore, we have released detailed documentation on training, evaluation, and code functionality. Due to the inherent variability of 3D imaging datasets, preparation, anisotropy, and image quality, we expect that users will have to train their own models with data they have manually annotated. While annotation may be prohibitive, our pretrained models serve as a suitable starting point, which will significantly shorten the amount of data necessary to train a model with high segmentation accuracy. Our training scripts and detailed procedures are included in the documentation for external use.

In summary, SKOOTS bridges the gap between performance and accuracy for previously difficult segmentation conditions. Here we have outlined the segmentation procedure, created a new neural network architecture, and released a new 3D mitochondria segmentation dataset. By expanding on previous flow techniques with skeleton prediction, SKOOTS demonstrates a step forward in high-resolution instance segmentation.

MATERIALS AND METHODS

FIB-SEM Data Collection

The hair cell FIB-SEM dataset was collected as part of a separate study, detailed fully elsewhere[45]. In summary, early postnatal mouse organotypic cochlea cultures were utilized for an aminoglycoside damage repair assay, prior to fixation, processing, and imaging by FIB-SEM. Briefly, cultures were prepared from wild-type CD-1 mice at postnatal-day 2 (P2) as described previously[46] and maintained in vitro for 1 day. Following a 15-minute exposure to 1 mM neomycin in HEPES-

buffered HBSS (HBHBSS), cultures were washed with cold HBHBSS then incubated on ice at 4°C in HBHBSS with 0.5 mg/ml cationized ferritin for 15 minutes to track endocytosis. Cultures were washed with HBHBSS and allowed to recover in culture medium at 37°C/5% CO_2 for 15 minutes, prior to fixation. It is not anticipated that this assay will disrupt mitochondria imaging or analysis. Animal work was carried out at the University of Sussex in full accordance with United Kingdom Home Office regulations.

Cultures were fixed for 2 hours at room temperature with 2.5% glutaraldehyde, 2 mM calcium chloride in 0.1 M sodium cacodylate buffer; followed by secondary fix with 1% osmium tetroxide, 1.5% potassium ferrocyanide in 0.1 M sodium cacodylate buffer for a further 2 hours. Samples were dehydrated by ethanol series and transferred to propylene oxide prior to embedding in an Epoxy Embed812/Araldite 502 resin mix.

Following tissue exposure by ultramicrotomy, FIB-SEM was carried out at the Center for Nanoscale Systems, Harvard University, on a FEI Helios Nano lab 660 operated at 3 kV. Milling and imaging was carried out in an orientation along the apical/basal axis of the organ of Corti, orthogonal to the reticular lamina, travelling through multiple outer hair cells in row one simultaneously. Three full outer hair cells in row one in the mid region of the coil were imaged in a serial FIB-SEM volume, 23.4(x) 18.5(y) 11.1(z) µm, collected at 3.82 nm/px with 20 nm milling steps. The resulting image stacks were aligned in Dragonfly Software, Object Research Systems (ORS) Inc, Montreal, Canada, 2022.

Mitochondria dataset annotation

The mitochondria training and validation dataset consists of an 1968(x) 3528(y) 201(z) px excerpt of the larger FIB-SEM hair cell dataset. Data were annotated using Dragonfly Software, Object Research Systems (ORS) Inc, Montreal, Canada, 2022, through a “human in the loop” approach by predicting semantic segmentation masks, assigning labels via watershed, and manually correcting the results.

In brief, using the Dragonfly segmentation wizard, 3D semantic segmentation masks of mitochondria were manually generated for a small part of the training and validation data subset and used to train the in software neural network model which generated masks for the entire subset volume. , separating each voxel into two labels: foreground (mitochondria) and background. A distance map was then generated from the semantic mask. This served two purposes: to aid in the generation of seed pixels for mitochondria, and to serve as the basins of watershed segmentation. Thresholds of the distance map served as seed pixels, which were then manually eroded to reduce over-segmentation. From these seeds, the watershed algorithm was used to fill the distance map basins and generate instance maps of mitochondria. While this approach was accurate for isolated, large mitochondria, the outcome was not reliable enough to serve as training data. When using this method, densely packed mitochondria tend to be under-segmented with poor borders, while small mitochondria were missed outright. Therefore, the resulting masks were visually inspected for errors, and manually corrected.

Ground truth skeleton calculation for training

To predict skeletons of objects using a neural network, we first needed to explicitly calculate ground truth skeletons for each manually annotated instance. We therefore calculated a ground truth skeleton for each unique instance in our segmentation datasets. Prior to skeletonization, we performed a linear interpolation on the binary mask of each image, scaling the instance closer to an isotropic pixel size The skeleton was a binary image, where each pixel represents the local mean of the object. We applied scikit-image’s skeletonize function to each scaled instance of our annotated dataset. The nonzero values of each instance skeleton were saved as a file for later training. In the case of small instances where skeletonization fails, we substituted the skeleton for the object’s local center of mass. We found the quality of calculated skeletons affects the accuracy of segmentation, and therefore up/down-scaling the instance mask of an object prior to skeletonization could produce skeletons which are better defined and lead to improved segmentation.

Data pre-processing and augmentation for training

Our training dataset consisted of multiple large, manually annotated images, and their associated ground truth labeled mitochondria instance masks, and precomputed skeletons. These images were too large to pass through a deep learning model at once. Rather, crops of 300 x 300 x 20 voxels were randomly centered at an instance of mitochondria. To improve generalizability of our model, we applied a suite of augmentations to these crops which are broadly separated into morphological augmentations (rotation, shearing, and flipping) and image augmentations (brightness and contrast adjustment, addition of random noise, and randomly inverting the image), these were applied were applied randomly to the image, masks, and skeletons.

To reduce false positive predictions, we also explicitly included images in the training dataset which do not contain any instances. Rather, we chose areas which contained objects visually like mitochondria, such as vacuoles or other organelles, improving the model's ability to discriminate such objects from mitochondria.

Training procedure

Training was distributed over two GPU's using the pytorch DistributedDataParallel model. Each training data image were cropped and augmented in a process outlined above and collated into a batch size of two crops of training data per GPU. We defined a single iteration over our dataset as 32 crops for each fully annotated training image. We trained the model for 10000 epochs with the AdamW optimizer with a weight decay of 1e-6 and a learning rate of 5e-4 which decays on a cosine annealing schedule with a period of 10000. The loss value for each output of the segmentation model, the semantic masks, skeletons, and embeddings, were calculated using the Tversky loss function, as this allowed for fine tuning the network's sensitivity to false positive and false negative predictions.

Rather than directly pre-computing optimal vector fields and computing a loss value against these as in Cellpose and Omnipose, we instead calculated a loss value for each vector based on the distance of the resulting embedding from an optimal skeleton location. This allowed for more flexibility in the model predictions and avoids having to re-compute flow fields after each augmentation, which is incredibly costly in 3D. The general steps were as follows: 1) for each ground truth mitochondria voxel we calculated the closest skeleton, 2) after applying the vectors we calculated a gaussian loss based on the distance and a standard deviation factor. Formally, the embedding loss ϕ at location ijk with embedding E_{ijk} and its associated skeleton S_{ijk} is the following:

$$\phi(\mathrm{E_{ijk}}, \mathrm{S_{ijk}}) = \exp\left(\sum_{\mathrm{l}\in[\mathrm{x,y,z}]} \frac{\left(\mathrm{E^{l}_{ijk}} - \mathrm{S^{l}_{ijk}}\right)^2}{-2\sigma_{\mathrm{l}}^2}\right)$$

When expanded to three dimensions:

$$\phi(E_{ijk}, S_{ijk}) = \exp\left(\frac{\left(E^{x}_{ijk} - S^{x}_{ijk}\right)^2}{-2\sigma_x^2} + \frac{\left(E^{y}_{ijk} - S^{y}_{ijk}\right)^2}{-2\sigma_y^2} + \frac{\left(E^{z}_{zi} - S^{z}_{ijk}\right)^2}{-2\sigma_z^2}\right)$$

By applying this to each voxel we generated a probability map for each voxel, which we then compared to the ground truth semantic segmentation mask via the Tversky loss.

Model Construction

The segmentation model was constructed from a standard UNet model by applying the same steps that would transforms a ResNet model into a ConvNext model[40] then increasing the dimensionality of each operation to 3D. Specifically: 1) we replaced each activation with a gaussian error linear unit[47] (GeLU), increased each kernel size of each convolution from three to seven , 3) replaced each double convolution computational block with 3D ConvNext block (an inverted bottleneck design), and 4) substituted each batch normalization layer with layer normalization[48]. While the original ConvNext used a layer scale initialization factor of 0.1, we elected to use a value of 1 as we observed no increase in final performance yet a considerable increase in the number of epochs to reach a training loss plateau.

Our implementation allowed for dynamic changes in the number of computational blocks, number of dimensions at each block, amount of downscaling, and offers 2D and 3D implementations. Each model used to evaluate any segmentation approach had an identical structure with two downscale steps, two upscale steps, and two computational blocks at each step. Each previously reported architectures used in this study for comparison (UNet, UNet++, Cellpose Net) were re-implemented by convention, but allowed for dynamic adjustment in scaling model size. The source code and implementation details may be found here: https://github.com/indzhykulian/bism.

Segmentation mask inference

SKOOTS applies the following steps for inference on new images. Arbitrarily large images are broken down into crops of identical size as training, which are then evaluated by a pretrained deep neural network model. The model predicts 5-channels of information: one channel for the semantic segmentation mask, one for the unlabeled skeleton, and three for the embedding vectors. Each embedding vector lies between -1 and 1, and therefore must be scaled such that any vector may be sufficiently large to point to a skeleton. Semantic segmentation masks are threshold at 0.5 and mask both the skeleton and embedding vectors, improving segmentation accuracy at the edges of objects. Vectors are scaled and stored, along with unlabeled skeleton masks, in pre-allocated arrays.

Following complete evaluation of the neural-network model, skeletons are assigned labels via flood fill. For speed, this process is also performed in crops, leading skeletons at the edges of a crop assigned multiple ID values, which are then detected and replaced such that each skeleton has one ID. This replacement occurs in place, avoiding copies and lowering the resulting memory footprint. Once each skeleton is assigned an ID, vectors select a voxels ID value by indexing the labeled skeleton mask. Formally, a neural network predicts 3 matrices: an unlabeled 3D semantic mask M, labeled skeleton mask S and predicted embedding vector V which all have shape (X_{in}, Y_{in}, Z_{in}) where (X_{in}, Y_{in}, Z_{in}) is the shape of the input image. Each value of these matrices may be indexed by the indices ijk where i is the X index and lies between 0 and X_{in}, j is the Y index and lies between 0 and Y_{in} and k is the Z index and lies within 0 and Z_{in}. Therefore,

$$\mathrm{i} \in [0, \dots, X_{in}]$$
$$j \in [0, \dots, Y_{in}]$$
$$k \in [0, \dots, Z_{in}]$$

The value at each location of the embedding matrix $\mathrm{V_{ijk}}$ is defined as a vector which points to the closest voxel of a predicted instance skeleton. Therefore,

$$V_{ijk} = [\Delta x, \Delta y, \Delta y]$$

We may use these embedding vectors to assign labels to the unlabeled semantic mask if the probability at pixel location ijk is above some threshold T_{mito} (i.e., it belongs to a mitochondria). Formally,

$$M_{jik} > T_{mito} \rightarrow M_{ijk} = S_{i+\Delta x, j+\Delta y, k+\Delta z} \text{ for } V_{ijk} = [\Delta x, \Delta y, \Delta z]$$

Simply, for each voxel of the input image, if that voxel represents a mitochondrion, there is an embedding vector which points it's associated skeleton. As these skeletons already have a label, the complete instance mask of each mitochondrion can be reconstructed by selecting and ID at each location via the embedding vectors.

Anisotropy

Anisotropic datasets occur when the absolute distance of a voxel a voxel represents is different along spatial dimensions. This commonly occurs when the resolution in Z of a volumetric imaging technique is lower than in X and Y. SKOOTS adapted to anisotropic data in two ways. First in training: Embedding vectors were predicted to lie at the closest skeleton. This was calculated as the minimum Euclidean distance between the location of the origin voxel, and all skeleton voxels. In some cases, this distance may force adjacent vectors to point in extremely different directions,

usually in the Z direction. To combat this, the Euclidean distance from each voxel to the skeleton was scaled based on the anisotropy of the dataset, forcing the model to learn smooth embedding vector fields. Secondly, anisotropy was corrected at evaluation by scaling embedding vectors. Embedding vectors, when predicted, lie between -1 and 1. In a heavily anisotropic dataset, a pixel may need to be pushed in X and Y for tens of voxels to lie within a skeleton, but may be pushed in Z by one or two. With equal scaling of X, Y and Z, embedding vectors in the Z direction tended to be small and poorly predicted by the model, limiting overall performance. Therefore, we scaled embedding vectors such that all embedding vector predictions have similar minimum and maximum. In the case of our mitochondria dataset, we scaled the embedding vectors in X and Y by 60, and Z by 12. Formally, each embedding vector $V_{ijk}^{l} \in [-1, \dots, 1]$ at location ijk and dimension $l \in [X, Y, Z]$ is scaled by a scaling factor α^{l}. Therefore $V_{ijk} = \left[V_{ijk}^{x} * \alpha^{x}, V_{ijk}^{y} * \alpha^{y}, V_{ijk}^{y} * \alpha^{y}, \right]$.

Confocal image analysis

Cochlear hair cell tissue was prepared with a protocol described in Buswinka *et al.* 2023[49], and imaged with a Leica SP8 confocal microscope (Leica Microsystems) using a 63×, 1.3 NA objective. Confocal Z-stacks of 512 x 512 pixel images with an effective pixel size of 144 nm were collected using the tiling functionality of the Leica LASX acquisition software. All experiments were carried out in compliance with ethical regulations and approved by the Animal Care Committee of Massachusetts Eye and Ear. The dataset containing Arabidopsis thaliana lateral root primordia cell walls stained with DAPI was previously reported by Wolny et al. 2020[44] and publicly available for download. Confocal datasets were manually annotated (when necessary) to generate training data and trained in an identical manner as above. As these datasets have differing pixel spacing, and therefore anisotropy, we adjusted the maximum embedding vector scaling accordingly. The brightness and contrast of input images were adjusted prior to segmentation.

Accuracy and morphology metrics

The accuracy of our segmentation approach was validated in two ways: 1) by the calculation of predicted instance intersection over union with associated ground truth segmentation masks, and 2) by the comparison of calculated morphology statistics. The intersection over union for each ground truth instance mask and associated predicted segmentation mask was calculated in pytorch. The population morphology statistics were calculated in the Dragonfly software for both ground truth segmentation masks and predicted segmentation masks and further analyzed in python.

Segmentation deep-learning architecture comparison

To select for the optimal architecture (the connections and operations of a deep-learning model) for our segmentation tasks, we validated our UNet + ConvNext blocks against other published architectures for biomedical segmentation, selecting the model with the highest performance for the least number of floating-point operations (FLOPS). Rather than training each model architecture on each respective segmentation task, we instead trained each on a semantic segmentation task, which is easier for a model to perform, and thereby trains faster. Each segmentation model architecture was implemented in PyTorch. Each model was trained with identical training parameters on our manually annotated mitochondria dataset. To use our data for a semantic task, we treated each uniquely labeled voxel as foreground, and each unlabeled as background. To facilitate reproducibility, we trained with identical random number generator seeding for each. To study the effect of model size, we started with a small number of convolution channels, and following training, doubled the channel count at each stage of the model and retrained. This process continued until a model's total memory requirements eclipsed the total video memory of our graphics cards (2x48 GB). FLOPS were calculated through the facebook research core vision library (fvcore, https://github.com/facebookresearch/fvcore) by passing a 300(x) 300(y) 20(z) voxel volume of random data through an untrained model. Model peak memory usage was calculated through PyTroch's CUDA interface.

Segmentation approach validation

Each segmentation approach may be evaluated on any deep neural network architecture which affects accuracy and performance. Furthermore, training procedure, augmentations, hyperparameters, and other subtle features can affect

overall approach accuracy. Therefore, to directly compare approaches, each were re-implemented to allow for identical training procedure and as similar inference as possible. Each aspect of training was defined by a single configuration file, with no more than necessary differences in training. Data, epochs, learning rate, data augmentations, model size, among other parameters, were kept identical for each approach. Implementation, along with training configuration files of may be found here: https://github.com/indhzkyulianlab/bism.

3D rendering

Two styles of 3D renders are presented in this work. **Figure 1g-l** and **figure 4f-h**were generated in matplotlib[50] with a custom script. Briefly object meshes were approximated from input volumes (either segmentation masks or skeletons) by a marching cubes algorithm from scikit-image[51]. For speed, we used a step size of 2 and appropriate spacing to account for anisotropy. Each ID value was assigned a unique color and rendered in matplotlib. When appropriate, skeleton embedding vectors were rendered with the 3D quiver plot in Matplotlib, assigning a color based on the direction of the vector. For speed and clarity, only every 10th vector was rendered in X and Y. Movies showing the rotations of these were generated by changing the iterating the view angle of the 3D plot from 0 to 360, saving an image at each frame, then creating an animated GIF image in ImageJ[52].

Renderings in **figure 3d, 3e** and **figure 4k-m** were generated by the Dragonfly Software. Multi-stack tifs of the input image, skeletons, and instance masks were imported into Dragonfly. A multi-ROI was generated from each greyscale ID value, and a random color was automatically assigned. The brightness and contrast of the image volume were adjusted. The multi-ROI of the skeletons and instance masks were overlayed on the input stack and rendered with Dragonfly's 3D viewer.

Code and computational environment

SKOOTS is operating system agnostic. All scripts were run on a computer running Ubuntu 20.04.1 LTS, an open-source Linux distribution from Canonical based on Debian. The workstation was equipped with two Nvidia A6000 graphics cards for a total of 98 GB of video memory as well as 256 GB of RAM. Many scripts were custom written in python 3.10 using open source scientific computation libraries including numpy[53], matplotlib, scikit-learn[54]. All deep learning architectures, training logic, and much of the data transformation pipeline was written in pytorch[55], utilizing the torchvision[55] library. The source code may be found here: https://github.com/indzhkyulianlab/skoots.

ACKNOWLEDGEMENTS

We would like to thank Prof. Guy Richardson for providing a tissue sample for FIB-SEM imaging. We also thank Dr. Stephan Kraemer (Harvard University Center for Nanoscale Systems) for their assistance in imaging.
This work was supported by NIH R01DC020190 (NIDCD), R01DC017166 (NIDCD) and R01DC017166-04S1 "Administrative Supplement to Support Collaborations to Improve the AI/ML-Readiness of NIH-Supported Data" (Office of the Director, NIH) to A.A.I. and the Speech and Speech and Hearing Bioscience and Technology Program Training grant T32 DC000038 (NIDCD). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The FIB-SEM imaging was performed at the Harvard University Center for Nanoscale Systems (CNS), a member of the National Nanotechnology Coordinated Infrastructure Network (NNCI), which is supported by the National Science Foundation under NSF award 1541959.

CODE AVAILABILITY

Code outlining SKOOTS segmentation is freely available here: https://github.com/indzhykulianlab/skoots, while training scripts, model construction, and augmentation are found here: https://github.com/indzhykulianlab/bism. SKOOTS documentation is publicly hosted here: https://skoots.readthedocs.io.

References:

1. Chan DC, Detmer SA. Functions and dysfunctions of mitochondrial dynamics. *Nature reviews Molecular cell biology.* 2007;8(11):870-879.
2. Pigino G, Morfini G, Pelsman A, Mattson MP, Brady ST, Busciglio J. Alzheimer's presenilin 1 mutations impair kinesin-based axonal transport. *Journal of Neuroscience.* 2003;23(11):4499-4508.
3. Rui Y, Tiwari P, Xie Z, Zheng JQ. Acute impairment of mitochondrial trafficking by β-amyloid peptides in hippocampal neurons. *Journal of Neuroscience.* 2006;26(41):10480-10487.
4. Trushina E, Dyer RB, Badger JD, et al. Mutant huntingtin impairs axonal trafficking in mammalian neurons in vivo and in vitro. *Molecular and cellular biology.* 2004;24(18):8195-8209.
5. Wang X, Winter D, Ashrafi G, et al. PINK1 and Parkin target Miro for phosphorylation and degradation to arrest mitochondrial motility. *Cell.* 2011;147(4):893-906.
6. Ong S-B, Hausenloy DJ. Mitochondrial morphology and cardiovascular disease. *Cardiovascular research.* 2010;88(1):16-29.
7. Beurg M, Nam JH, Chen Q, Fettiplace R. Calcium balance and mechanotransduction in rat cochlear hair cells. *J Neurophysiol.* 2010;104(1):18-34.
8. Wong H-tC, Zhang Q, Beirl AJ, Petralia RS, Wang Y-X, Kindt K. Synaptic mitochondria regulate hair-cell synapse size and function. *eLife.* 2019;8.
9. Thomas CI, Keine C, Okayama S, et al. Presynaptic mitochondria volume and abundance increase during development of a high-fidelity synapse. *Journal of Neuroscience.* 2019;39(41):7994-8012.
10. Devine MJ, Kittler JT. Mitochondria at the neuronal presynapse in health and disease. *Nature Reviews Neuroscience.* 2018;19(2):63-80.
11. Jeyaraju DV, Cisbani G, Pellegrini L. Calcium regulation of mitochondria motility and morphology. *Biochimica et biophysica acta Bioenergetics.* 2009;1787(11):1363-1373.
12. Chinopoulos C, Adam-Vizi V. Mitochondrial Ca2+ sequestration and precipitation revisited. *The FEBS journal.* 2010;277(18):3637-3651.
13. Esterberg R, Linbo T, Pickett SB, et al. Mitochondrial calcium uptake underlies ROS generation during aminoglycoside-induced hair cell death. *The Journal of clinical investigation.* 2016;126(9):3556-3566.
14. Kokoszka JE, Coskun P, Esposito LA, Wallace DC. Increased mitochondrial oxidative stress in the Sod2 (+/−) mouse results in the age-related decline of mitochondrial function culminating in increased apoptosis. *Proceedings of the National Academy of Sciences.* 2001;98(5):2278-2283.
15. Orrenius S, Zhivotovsky B, Nicotera P. Regulation of cell death: the calcium–apoptosis link. *Nature reviews Molecular cell biology.* 2003;4(7):552-565.
16. Ott M, Gogvadze V, Orrenius S, Zhivotovsky B. Mitochondria, oxidative stress and cell death. *Apoptosis.* 2007;12(5):913-922.
17. Someya S, Prolla TA. Mitochondrial oxidative damage and apoptosis in age-related hearing loss. *Mechanisms of ageing and development.* 2010;131(7-8):480-486.
18. Owens KN, Cunningham DE, Macdonald G, Rubel EW, Raible DW, Pujol R. Ultrastructural analysis of aminoglycoside-induced hair cell death in the zebrafish lateral line reveals an early mitochondrial response. *Journal of Comparative Neurology.* 2007;502(4):522-543.
19. Wei D, Lin Z, Franco-Barranco D, et al. MitoEM dataset: large-scale 3D mitochondria instance segmentation from EM images. Paper presented at: International Conference on Medical Image Computing and Computer-Assisted Intervention2020.
20. Liu W, Qiao Q, Zheng J, et al. An assembly-regulated SNAP-tag fluorogenic probe for long-term super-resolution imaging of mitochondrial dynamics. *Biosensors and Bioelectronics.* 2021;176:112886.
21. Drobne D. 3D imaging of cells and tissues by focused ion beam/scanning electron microscopy (FIB/SEM). In: *Nanoimaging.* Springer; 2013:275-292.

22. Perkins G, Lee JH, Park S, et al. Altered outer hair cell mitochondrial and subsurface cisternae connectomics are candidate mechanisms for hearing loss in mice. *Journal of Neuroscience.* 2020;40(44):8556-8572.
23. Kornfeld J, Denk W. Progress and remaining challenges in high-throughput volume electron microscopy. *Current Opinion in Neurobiology.* 2018;50:261-267.
24. Turaga SC, Murray JF, Jain V, et al. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural computation.* 2010;22(2):511-538.
25. Turaga SC, Briggman KL, Helmstaedter M, Denk W, Seung HS. Maximin affinity learning of image segmentation. *arXiv preprint arXiv:09115372.* 2009.
26. Wang X, Tang Y, Masnou S, Chen L. A global/local affinity graph for image segmentation. *IEEE Transactions on Image Processing.* 2015;24(4):1399-1411.
27. Zeng T, Wu B, Ji S. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. *Bioinformatics.* 2017;33(16):2555-2562.
28. Beier T, Pape C, Rahaman N, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature methods.* 2017;14(2):101-102.
29. Lee K, Zung J, Li P, Jain V, Seung HS. Superhuman accuracy on the SNEMI3D connectomics challenge. *arXiv preprint arXiv:170600120.* 2017.
30. Funke J, Tschopp F, Grisaitis W, et al. Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE transactions on pattern analysis and machine intelligence.* 2018;41(7):1669-1680.
31. Bailoni A, Pape C, Wolf S, Beier T, Kreshuk A, Hamprecht FA. A generalized framework for agglomerative clustering of signed graphs applied to instance segmentation. *arXiv preprint arXiv:190611713.* 2019.
32. Lee K, Lu R, Luther K, Seung HS. Learning and segmenting dense voxel embeddings for 3D neuron reconstruction. *IEEE Transactions on Medical Imaging.* 2021;40(12):3801-3811.
33. Stringer C, Wang T, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods.* 2021;18(1):100-106.
34. Cutler KJ, Stringer C, Lo TW, et al. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature methods.* 2022:1-11.
35. Meyer F, Maragos P. Multiscale morphological segmentations based on watershed, flooding, and eikonal PDE. Paper presented at: International Conference on Scale-Space Theories in Computer Vision1999.
36. Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS).* 2017;42(3):1-21.
37. Gan J, Tao Y. DBSCAN revisited: Mis-claim, un-fixability, and approximation. Paper presented at: Proceedings of the 2015 ACM SIGMOD international conference on management of data2015.
38. Conrad R, Narayan K. Instance segmentation of mitochondria in electron microscopy images with a generalist deep learning model trained on a diverse dataset. *Cell Systems.* 2023;14(1):58-71. e55.
39. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. Paper presented at: International Conference on Medical image computing and computer-assisted intervention2015.
40. Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S. A convnet for the 2020s. Paper presented at: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition2022.
41. Kornilov AS, Safonov IV. An overview of watershed algorithm implementations in open source libraries. *Journal of Imaging.* 2018;4(10):123.
42. Maragos P, Schafer R. Morphological skeleton representation and coding of binary images. *IEEE Transactions on Acoustics, Speech, and Signal Processing.* 1986;34(5):1228-1244.
43. Burtsev S, Kuzmin YP. An efficient flood-filling algorithm. *Computers & graphics.* 1993;17(5):549-561.

44. Adrian W, Lorenzo C, Athul V, et al. Accurate and Versatile 3D Segmentation of Plant Tissues at Cellular Resolution. *bioRxiv.* 2020:2020.2001.2017.910562.
45. Osgood RT. *Cellular mechanisms of damage and repair in sensory hair cells*, University of Sussex; 2020.
46. Russell I, Richardson G. The morphology and physiology of hair cells in organotypic cultures of the mouse cochlea. *Hearing research.* 1987;31(1):9-24.
47. Hendrycks D, Gimpel K. Gaussian error linear units (gelus). *arXiv preprint arXiv:160608415.* 2016.
48. Ba JL, Kiros JR, Hinton GE. Layer normalization. *arXiv preprint arXiv:160706450.* 2016.
49. Buswinka CJ, Osgood RT, Simikyan RG, Rosenberg DB, Indzhykulian AA. The hair cell analysis toolbox is a precise and fully automated pipeline for whole cochlea hair cell quantification. *Plos Biology.* 2023;21(3):e3002041.
50. Hunter JD. Matplotlib: A 2D graphics environment. *IEEE Annals of the History of Computing.* 2007;9(03):90-95.
51. Van der Walt S, Schönberger JL, Nunez-Iglesias J, et al. scikit-image: image processing in Python. *PeerJ.* 2014;2:e453.
52. Rasband WS. ImageJ. In: Bethesda, MD; 1997.
53. Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. *Computing in science & engineering.* 2011;13(2):22-30.
54. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research.* 2011;12:2825-2830.
55. Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:191201703.* 2019.