Instance segmentation of mitochondria in electron microscopy images with a generalist deep learning model.

Ryan Conrad [1,2] and Kedar Narayan [1,2]

[1] Center for Molecular Microscopy, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda 20892, Maryland, USA.

[2] Cancer Research Technology Program, Frederick National Laboratory for Cancer Research, Frederick 21702, Maryland, USA.

Abstract

Mitochondria are extremely pleomorphic in biology. Automatically identifying each one precisely and accurately from any 2D or volume electron microscopy (volume EM) dataset is an unsolved computational challenge. Current deep learning (DL) models are trained within limited contexts, restricting their widescale utility and potential as a universal or generalist solution for mitochondrial segmentation. To address this, we amass a highly heterogeneous ~1.5 x 10^6 unlabeled cellular EM image dataset and a ~22,000, partially crowdsource-labeled, mitochondrial instance segmentation dataset. We release MitoNet, a DL model trained on these data, which performs well on new and challenging volume EM benchmarks. An accompanying Python package and napari plugin, called empanada, can be used for efficient training, inference, and clean-up of instance segmentations on EM images.

Keywords

Volume EM, segmentation, deep learning, electron microscopy, Panoptic, benchmark, image dataset, crowdsourcing

Introduction

Electron microscopy (EM) reveals 2D snapshots of cellular and subcellular ultrastructure at unrivaled resolutions, and more recent volume EM approaches extend these into the third dimension[1,2]. Technological advances in volume EM have dramatically increased the size of volumes that can be interrogated, most notably in connectomics[3–5]. Developing meaningful quantitative insights from EM data often requires the segmentation of features of interest. For these large neuronal datasets, machine learning (ML) or deep learning (DL) based tracing algorithms have helped generate detailed wiring diagrams[6–10]. Similar DL approaches have also provided insights into a variety of sub-cellular structures in other systems[11–15].

As one of the most ubiquitous and morphologically complex organelles in biology[16–18] and a critical player in cellular physiology[19–21] and pathological processes[22–24], mitochondria are frequent targets of analyses. Mitochondria have extraordinarily variable sizes, shapes, distributions and intra-organellar structures, yet they are instantly recognizable by their ultrastructure. This suggests that a universal DL model that accurately and precisely recognizes individual mitochondria in any given EM image should be possible.

Current approaches largely take the following strategy: a region of interest (ROI) from a dataset is chosen for dense manual segmentation, a model is trained on this ROI and then inference is run on the larger dataset, often followed by some manual "polishing"[6,7,11,13,25,26]. This strategy can lead to visually impressive results[11,27]; however, the constrained contexts presented during model training are one reason why such models have failed to generalize. As a result, segmentation quality drops when the models are presented with unseen cell and tissue types, sample preparation and imaging protocols, or even in some cases, regions of a dataset spatially distant from the training ROI[11,13,28]. Poor generalization limits the usefulness of models, as unfamiliar datasets require revisiting the cycle of manual annotation, model training, hyperparameter tuning, and prediction updating. Achieving good generalization is a challenging task. Mitochondria are present in nearly all eukaryotic cells such that the landscape of cellular contexts that a generalist segmentation model must account for is immense. This challenge fundamentally distinguishes mitochondrial segmentation from other tasks such as tracing neurites or synapses that exist in a single tissue context (e.g., brain).

There are several publicly available datasets with segmented mitochondria from a narrow range of tissues: MitoEM,[26] Perez et al.,[29] Lucchi++ and Kasthuri++[25] are from brain tissue, Guay et al.[30] from human platelets, UroCell[13] from urinary bladder and Heinrich et al.[11] from three kinds of *in vitro* cells **(**HeLa, Jurkat and macrophage**)**. Only one of these datasets (MitoEM) includes instance segmentations (i.e., where each mitochondrion is assigned a different label). We hypothesized that sparse 2D instance segmentations from an eclectic set of EM images would more effectively expand the range of contexts as compared to dense volumetric segmentation. Moreover, instance segmentation can unlock methods for the quantification of mitochondrial morphologies[31–34], networks[18,35], and fusion and fission dynamics[36,37], for which semantic segmentation is insufficient.

Here, we have curated a heterogeneous, non-redundant, information-rich and relevant unlabeled EM image dataset (at ~1.5 x 10^6 images, the largest of its kind to our knowledge), called *CEM1.5M,* for use as a database from which to sample images for mitochondrial segmentation. Through the unification of existing labeled datasets and crowdsourced annotations of images from CEM1.5M, we have created a similarly diverse labeled image dataset called *CEM-MitoLab* for training ML/DL models. *MitoNet* is the model resulting from self-supervised pre-training on CEM1.5M followed by supervised training on CEM-MitoLab. We show that MitoNet outperforms other candidate datasets when tested on new and challenging benchmarks (we also share these benchmarks). Finally, in line with our overall mission is to democratize these efforts, we have created a Python package and napari plugin called *empanada*, which allows easy model training and inference as well as prediction clean-ups for MitoNet and any other ML/DL models.

Results

Dataset Collation

The first step in creating this resource was the generation of a database of unlabeled EM images of cells and tissues both for self-supervised model pre-training and as a source for sampling images for mitochondrial annotation. We collected a total of 463 volume EM datasets, of which 352 were produced in-house at the Center for Molecular Microscopy (CMM) and 111 were generated and publicly deposited by external sources. Datasets larger than 5 GB were randomly cropped into 320 cubes of 256x256x256 (equaling 5 GB of unsigned 8-bit voxels) to limit their overrepresentation. Smaller datasets were kept at their original dimensions. In total, this yielded 15,152 ROIs, corresponding to 338 GB. To these we added 5,390 2D EM images acquired at comparable pixel samplings (traditional TEM and STEM images of stained and resin-embedded biological samples), of which 1,738 images were from CMM, 2,261 were from the NCI electron microscopy lab and 1,391 were from external sources. Lastly, 27 video files of volume EM data from online publications were incorporated. Metadata and appropriate attribution were recorded, where possible, for all datasets (**Supplementary File 1).** We then applied the data preparation and curation pipeline previously developed in our work on the CEM500K dataset[38] to generate approximately 1.5 x 10^6 2D image patches of 224x224 pixels. Patches from isotropic voxel datasets were derived from xy, xz and yz planes. The curation pipeline removed near-duplicate patches within each dataset before a neural network classified patches as “informative” or “uninformative”, the latter of which were discarded.

Of the combined 490 volume and video datasets, we selected 478 for annotation. The excluded volumes were those with pixel sizes greater than 40 nm or severe artifacts that likely would have made accurate annotation impossible. 20 of the volumes had 2D or 3D ROIs with mitochondrial instance segmentations (13 externally sourced; seven from CMM). The 3D segmentations were cropped into 2D patches and combined with 3,289 previously generated in-house segmentations of 2D data to form the “legacy” portion of the supervised dataset (see **Materials and Methods**). For all the remaining volumes, which had no corresponding mitochondrial annotations, we sampled patches for later manual segmentation. The overall curation process is outlined in

Figure 1a. Even with the 5 GB cap, the number of patches derived from each dataset within CEM1.5M varied significantly (**Figure 1b**) such that a naïve random sampling would have resulted in drastically oversampling the largest ones. Therefore, when picking annotation targets, we chose a maximum of 15 random patches from every dataset. For large and heterogeneous datasets this very sparse sampling likely misses some unique cellular contexts. However, we decided that for testing our hypothesis and the release of these resources, this level of sampling would adequately represent the data landscape while avoiding the time and labor cost of denser annotation. For the 2D images in CEM1.5M, we circumvented unequal sampling by coarsely grouping sets of 2D images together by known publication or imaging project (see **Materials and Methods**). This resulted in the creation of 83 groups of 2D EM images. To compensate for the possibility that groups may each represent multiple biological contexts, we selected a maximum of up to 25 instead of 15 random patches.

At the end of data collation, we had 1,494,389 image patches for self-supervised pre-training, and for supervised training, nearly 15,897 previously annotated images from legacy data and 5,841 images set aside for crowdsourced annotation. Unlike CEM1.5M and legacy data, which were significantly imbalanced toward a few large datasets, the images set aside for annotation almost evenly represented every volume EM dataset. During model training, weighted sampling from datasets was used to correct for imbalance in the combined legacy and crowdsourced data, see **Figure 1b, Materials and Methods**). The total 21,738 annotated images (of various sizes, **Supplementary Figure 1a**) that constituted CEM-MitoLab contained 134,812 mitochondrial instances and represented myriad pixel resolutions, imaging techniques, sample preparation protocols, cells, tissues, and organisms (**Figure 1c-f, Supplementary Figure 1b**).

Our dataset consists exclusively of 2D images and favors a broad but superficial sampling of very many cellular contexts. **Supplementary Figure 1c** shows a random sample of patches from the Heinrich et al.,[11] and MitoEM[26] datasets, revealing a similar appearance across images. The mitochondria also showed little variation in contrast (**Supplementary Figure 1d**), as may be expected with the standardized sample preparation and imaging techniques used by these groups. By eschewing 3D segmentation strategies, we were able to quickly annotate widely, building a significantly larger and more heterogeneous dataset than the current alternatives. As a resource, the CEM1.5M dataset can be used to build similarly diverse labeled datasets for other downstream segmentation tasks.

Crowdsourced Annotation

We were inspired by the citizen science project Etch-a-cell[15] on the Zooniverse platform**,** in which thousands of volunteers annotated organelles like the nuclear envelope, mitochondria and endoplasmic reticulum (ER) in a single or small number of volume EM reconstructions. To create crowdsourced segmentations for mitochondria in CEM-MitoLab, we created a “private” project on Zooniverse, where 34 students recruited from local high schools were briefly trained to identify and annotate mitochondria in diverse and challenging EM images. **Supplementary Figure 1c** demonstrates the difficulty of the task: The appearance of a mitochondrion depends strongly on cell and tissue type, sample preparation and imaging approach. To give some 3D context and help resolve ambiguous images and edge cases, we presented images extracted from volumetric datasets as short “flipbooks” of five consecutive slices. For this task students were instructed to only annotate the middle slice of the flipbook (screenshots of the user interface shown in **Supplementary Figure 2a**). We also implemented three controls to enhance the quality of annotations. First, a retirement limit of ten was set such that each image was annotated independently by ten students, and not shown again. These ten annotations were combined to form a consensus instance segmentation (see **Materials and Methods, Supplementary Figure 3**). Second, we asked annotators to rate their confidence in their annotation on a scale from 1 to 5 (where 1 was not at all confident and 5 was very confident). Third, all consensus annotations were reviewed by at least two experienced CMM researchers to create the final ground truth segmentation.

For this project, nine sets of 500-1000 images were uploaded to the Zooniverse platform over a period of six months. These sets were organized into three groups with each group containing patches from different 2D and 3D datasets. Group 1 contained a mixture of in-house and external volume EM datasets, Group 2 contained in-house and NCI 2D datasets and Group 3 exclusively contained in-house volume EM datasets acquired since the launch of the project. As a baseline, and to test the annotation interface, the first two sets of images (referred to as Group 1a) were annotated by seven experienced EM researchers, with an image retirement limit of five. The remaining three sets of images in Group 1 (referred to as Group 1b) along with Group 2 and Group 3 were annotated exclusively by students.

We found that the consensus annotations for mitochondria with clearly defined outer membranes and cristae were often high-quality, especially at the pixel-level **(Figure 2a)**. 39.6% of consensus annotations required no corrections upon expert review, 12.5% required cosmetic pixel-level corrections and the remaining 47.9% required some combination of pixel-level, false positive and false negative corrections. Among this last set, the average correction added 3.7 false negatives and removed 0.4 false positives per image. The total number of corrections for each group of images are tabulated in **Supplementary Figure 2b**. Annotations in need of the most corrections were often the result of systematic gaps in the annotators' knowledge. Such errors were manifested as annotations with strong mutual agreements which nevertheless were far from the expert-reviewed ground truth (**Supplementary Figure 2c**). Nevertheless, student annotation performance as a whole improved over the course of the project. The students' first annotations had much lower F1 scores measured at an intersection-over-union (IoU) threshold of 0.5 (F1@50) than experts **(Figure 2b)**. By Group 3, the median student F1@50 scores had risen from 0.57 to 0.72, and impressively, the top 50% of students achieved F1@50 scores that were within the range of experienced annotator scores for Group 1a. This improvement occurred despite the introduction of previously unseen cellular contexts in each group and may be attributed to the heterogeneous data encountered earlier in the project. The improvements were not uniform for all students and this was reflected by a widening range in annotator performance with each new group of images. Seven of the 29 annotators (five dropped out before the last group of images) actually did worse over time **(Figure 2b).**

Despite the variability in annotator quality, consensus annotations were always near or above the 75th percentile of annotators **(Figure 2b)**. This underscores the effectiveness of acquiring multiple independent annotations and applying a robust consensus algorithm. Although there was a slight upward trend with increasing the limit for students, the gains in annotation quality appeared to start leveling off after five or six regardless of the group of images being annotated (**Figure 2c**). The lower limit of five for experienced annotators was sufficient to reach nearly perfect annotations, with consensus scores reaching 0.9. With a retirement limit of ten for the students annotators, we observed that the best consensus annotations on a per-image basis were created by setting a vote threshold of five (i.e., at least five annotators must have agreed on the label of a pixel for that label to be used in the consensus) (**Figure 2d**). This vote threshold also achieved a reasonable balance between true positive, false positive and false negative detections **(Supplementary Figure 2d)**. While this vote threshold was set after evaluating results of Group 1, we subsequently observed that F1@50 over all mitochondrial instances that were annotated by students, instead of over all images, was marginally higher at a threshold of three **(Figure 2d)**. However, even with a vote threshold of five for all consensus annotations, our approach created each instance segmentation as accurately as possible, independent of number of mitochondria per image.

Lastly, we hoped that self-reporting of annotation confidence could be used to filter out low-quality predictions, instead we found this factor to be a poor proxy for annotation quality. While there was an increase in average F1@50 with increasing confidence scores, there was also a wide variance. Thus discarding “low confidence” annotations would have thrown out nearly as many good as bad annotations. **(Figure 2e)**. There was only a weak correlation (r=0.322) between an annotator’s average reported confidence score and that annotator’s average annotation quality (**Figure 2f**), suggesting different levels of baseline confidence within the student cohort. For larger crowdsourced datasets where proofreading every annotation is infeasible, confidence scores may still be useful as a guide for identifying annotations most likely to require correction. Surprisingly, we also observed that there was no correlation (r=-0.096) between the number of annotations a student completed and their average F1@50 score (**Figure 2g**). These findings highlight the difficulty of gathering annotations from the public even with a motivated and trained cohort. Future crowdsource projects may benefit from targeted coaching or exclusion/weighting of lower-performing annotators. Time and resources must be invested to teach the skills necessary for handling such tasks, and still, reliable and high-performing annotators can only be identified after expert review and corrections over several rounds.

Benchmark Datasets

To assess how well models trained on the collected mitochondrial data might generalize to future unseen images, we withheld six instance segmented volumes from the above pipelines. These volumes were chosen to encapsulate a few different cellular contexts, mitochondrial morphologies, and sample preparation protocols. Crucially, they also represented varying levels of difficulty to help identify deficiencies in the training dataset, model architectures and postprocessing algorithms. The six volumes, five of which were not previously annotated, included conventionally fixed, heavy metal stained and resin-embedded samples (different sample preparation protocols, and imaged independently) of fly[39] and mouse brain[25,40] tissue, glycolytic muscle tissue[35], mouse salivary gland, and a HeLa cell. The sixth volume was a high-pressure frozen (HPF), freeze substituted C. elegans embryo specimen. The first three of these were generated externally and the last three were generated in-house. Representative 2D images and 3D renderings of mitochondria from each dataset are shown in **Figure 3a.** The mouse brain volume was the test set from the commonly used Lucchi++ benchmark (we converted the semantically segmented mitochondria to instance segmentation, see **Materials and Methods**). This benchmark was useful to calibrate performance because there were established results for comparison. Importantly, unlike prior research on this dataset, the entire Lucchi++ dataset was excluded from our training data. Aside from the Lucchi++ benchmark, which was imaged at isotropic 5 nm voxel resolutions, we intentionally chose or resampled the other volumes to have intermediate resolutions (10-25 nm) in line with the most common resolutions in the training dataset. In this range, mitochondria are generally still easy to identify on the whole but may be small.

Each of the new benchmark volumes presented unique challenges. The mitochondria in the fly brain volume were morphologically simple but appeared in two distinct variants: lightly stained with poorly defined cristae and darkly stained with well-defined cristae (**orange and blue arrows respectively**). The HeLa cell volume, in addition to being cluttered with subcellular features such as Golgi, ER and a variety of vesicles, had areas of localized heavy metal precipitation from sample preparation (**green arrow**). The C. elegans embryo volume has a mixture of two mitochondrial morphologies that are difficult to segment: small puncta and skinny tubules, with a small median cross-sectional radius of six voxels (**Figure 3b**). Moreover, this volume also contained membranous organelles[41] that might induce false positive detections as they appear similar to mitochondria with swollen cristae at these resolutions (**yellow arrow**). Finally, overall contrast was relatively lower, as expected for high-pressure freezing (HPF) and freeze-substitution used in the sample preparation protocol. The glycolytic muscle volume had a relatively uncluttered background but more complex and elaborate branched morphologies. About a third of the mitochondria had more than one branch and ten had four or more long branches, with the most branched mitochondrion having 11 (**Figure 3b**). The salivary gland volume was by far the most challenging. Besides the unusual tissue context, the mitochondria were weakly stained relative to the cytoplasm, which was packed with ER. Many mitochondria were also flat or bowl-shaped and tightly pressed against and around acini (**red arrow**). Especially difficult for accurate instance segmentation, mitochondria were closely packed together with the boundaries of over 75% being within just five voxels of the next nearest mitochondrion's boundary (**Figure 3b**).

Instance Segmentation Algorithm

For 2D instance segmentation of mitochondria, we adopted the Panoptic-DeepLab[42] (PDL) architecture. PDL is an encoder-decoder architecture, similar to the standard U-Net[43]. As is common for biological images, PDL employs a bottom-up approach to instance segmentation. A key advantage of bottom-up algorithms is that they can segment an arbitrarily large number of objects within a field of view, in contrast to top-down algorithms like Mask R-CNN[44] or Mask2Former[45] that set limits on the number of objects. We trained PDL models to infer mitochondrial semantic segmentations, centers and per-pixel x and y offsets from each center **(Figure 4a)**. To create semantic segmentations at resolutions higher than the input image resolution, we also employed the PointRend[46] module. PointRend iteratively interpolates the segmentation, identifies the most uncertain pixels, and then reevaluates the label for those pixels to refine the segmentation **(Supplementary Figure 4)**. Thus PointRend allows the model to accept images that have been downsampled to lower resolutions – preferably in the 10-20 nm range to match our labeled dataset – and output segmentations with crisp boundaries at the original resolution without the need for a resolution-specific model architecture. Finally, the semantic segmentation, offsets and object centers were merged into an instance segmentation **(Figure 4a)** via a simple postprocessing algorithm (see **Materials and Methods).**

Next, we generalized the 2D instance segmentation algorithm to handle volume EM data. As implemented, PDL did not track objects through an image stack. Instead, we matched objects across consecutive slices by computing IoU scores between all pairs of instances and applying the Hungarian algorithm[47]. We observed that long mitochondrial instances in 2D were sometimes erroneously split into multiple fragments. Since the Hungarian algorithm assumes a 1-to-1 matching, these fragments were left unmatched with an object in the preceding slice. By computing intersection-over-area (IoA) scores we were able to identify fragments that were enclosed by a larger object on the preceding slice. These were then merged to the label of that larger object (**Figure 4b**). Since the matching algorithm relied only on consecutive slices, a false negative detection on even a single slice could break the connectivity of that object in 3D. To correct this, we used median within a short stack of the last 3, 5 or 7 recorded semantic segmentation probabilities (**Figure 4c**) to infer the correct segmentation for the current slice. As a final step, matching was performed in the reverse direction through the stack. This was essential to correctly merge branched mitochondria **(Figure 4d)**.

Optionally, for isotropic-voxel volumes, inference may be performed equivalently on images from the xy, xz and yz planes. Following our previous work[48], we refer to this as ortho-plane inference, although the same idea has been applied elsewhere[15,49]. In this case, the outputs of ortho-plane inference were three independent instance segmentations. We merged these into a single consensus instance segmentation by applying the same algorithm used during crowdsourcing (**Figure 4e**). In all experiments, unless otherwise noted, we applied ortho-plane inference to isotropic-voxel volumes. Other model architectures and postprocessing algorithms like watershed[50] could likely work comparably with the architecture presented here. Our design decisions were heavily influenced by computational efficiency. We used simple run-length encoding compression to represent all 2D and 3D instance segmentations. As a result, forward and backward matching of instances and the consensus algorithm, which must process three complete volumetric instance segmentations, was run on large datasets using minimal compute resources. Such considerations are critical for ensuring that this tool can be adopted widely within the EM community.

Model training and evaluation

Before training any models on the mitochondrial segmentations, we pretrained a ResNet50[51] model on CEM1.5M using the SwAV[52] self-supervised learning algorithm. This pretrained model was employed as the encoder network in our experiments. Our best performing PDL model was trained for 120 epochs on CEM-MitoLab (see **Materials and Methods**). Our results showed that our model, MitoNet, was able to segment a variety of mitochondrial instances accurately in both 2D and 3D (**Figure 5a, b**). Model performance was best on the brain tissue datasets (fly brain and Lucchi++) with both semantic IoU and F1@75 scores of about 0.9 and above **(Figure 5c)**. Detailed performance metrics of MitoNet over all benchmarks are included in Table 1. The high contrast, simplicity of mitochondrial morphologies, and relative homogeneity of tissue ultrastructure in these datasets presented an easier challenge. The model also correctly detected both the lightly and darkly stained variants of mitochondria in the fly brain volume as described above. Strikingly, the F1@75 score of 0.88 achieved on the Lucchi++ test set by our generalist model matched the result from models that were trained exclusively on the Lucchi++ training set[53]. Moreover, we observed that all false negative detections at an IoU threshold of 0.5 were from small instances that were truncated at the edge of the volume (data not shown).

MitoNet performance on the HeLa and glycolytic muscle benchmarks was also strong even though these datasets were more difficult. While semantic IoU scores still held within the 0.8-0.9 range, F1@75 scores were 0.50 and 0.60 respectively. For the HeLa benchmark, the model successfully ignored most of the intracellular clutter that was expected to be challenging (**Figure 5a**). Mitochondria in the glycolytic muscle dataset had clearly defined membranes and cristae that should make them easy to distinguish. That said, model performance was surprisingly good given that no images of glycolytic muscle were present in the training dataset at all and only a handful of images were from muscle tissue (mitochondrial ultrastructure is significantly different depending on the oxidative state of muscle)[54]. The lower F1 scores observed for instance segmentation, despite good semantic segmentation, were evidence that individual mitochondria were harder for our model and postprocessing to distinguish in these datasets. At an IoU threshold of 0.5, we observed that over 75% of the false negatives in the HeLa volume and about 50% in the glycolytic muscle volume occurred for mitochondria within five or fewer voxels of their nearest neighbor **(Supplementary Figure 5). Figure 5b** shows examples of the over merging errors that our method was prone to make. It should be noted, however, that the model still correctly detected 78% of the closely apposed mitochondria in the HeLa volume and 52% in the glycolytic muscle, and performed well with branched mitochondria. As an example, aside from one minor split, the model and postprocessing correctly handled a heavily branched mitochondrion in the glycolytic muscle volume **(Figure 5b,** blue arrow**).**

The C. elegans dataset presented a number of challenges for our model. The mitochondria were low contrast, and small and tightly packed together; this was made worse by the low pixel sampling (24 nm). To get the best performance on this dataset we found that it was necessary to use a lower vote threshold of only one plane for ortho-plane inference. With this lower vote threshold, the IoU score increased to 0.60 from 0.42, though the F1@75 score was lower at 0.35 (still a substantial increase from 0.19)**.** An IoU threshold of 0.5, the model failed to accurately detect any mitochondria at that were smaller than 1,000 voxels, or ~ 0.25 μm^3, in size **(Supplementary Figure 5)**; such small instances accounted for about 25% of all false negatives. The model also performed slightly worse on mitochondria with lower contrast (mean grayscale of correctly identified mitochondria (True Positive), 75.9; wrongly missed mitochondria (False Negative), 70.5, $p=3x10^{-4}$). Supplementary Table 1 lists the mean mitochondrial parameters corresponding to TP and FN detections across all benchmarks. Again, closely apposed instances accounted for over 75% of all false negatives. A cluster of a few tightly packed mitochondria could induce multiple false negatives and false positives – and rapidly decrease F1 scores – even when the voxel-level segmentation appeared quite good (**Figure 5b, black arrow**). The model did manage, however, to mostly avoid erroneously labeling the so-called membranous organelles, but occasionally labeled some vesicles as mitochondria (**Figure 5a**).

On the most challenging of the benchmarks, the salivary gland volume, MitoNet achieved an IoU score of just 0.1 and consequently negligible F1 scores. The unusual appearance and morphologies of the mitochondria and abundance of ER made this benchmark extremely difficult. Therefore, here we tested the ability of the model after minimal fine-tuning, i.e., after a second round of training on a small subset of the volume's data. We extracted a small fraction of ground truth data, 64 random patches of 224x224 pixels from this salivary gland dataset (~0.2%

of the total volume). Even with a very brief finetuning step of 500 iterations (~5 minutes on our system), we achieved dramatically improved semantic segmentation quality with IoU increasing to 0.65 **(Figure 5d).** F1@50 scores also improved, from 0.04 to 0.22. The lower F1 score overall was because, uniquely among these benchmarks, a large fraction of the mitochondria in this dataset were part of a network of closely apposed instances; our model and postprocessing merged nearly all of these mitochondria into a single large instance (**Figure 5a, Supplementary Figure 5**). However, we underscore an important point: Even when the generalist model fails on a given dataset, it is still a strong starting point for training a specialized model, and can succeed with a modest number of examples and compute power. The main challenges that remain for these benchmarks appear to be the handling of small objects and tightly packed instances (Supplementary Table 1). We found that instance branch length, measured in voxels, was not a significant factor that led to false negative detection **(Supplementary Figure 5)**, meaning that MitoNet and our 3D postprocessing algorithms were able to track mitochondrial instances over many slices of volume EM datasets without losing connectivity.

Finally, we directly probed the efficacy of CEM-MitoLab against other training datasets by measuring performance of models across all benchmarks except the salivary gland volume. Since each dataset included a different number of patches, for this experiment we trained all models for approximately 10,000 iterations to control bias toward larger training datasets (the 120 epochs used for MitoNet was the equivalent of about 40,000 iterations). The labeled datasets for training included MitoEM[26] and Heinrich et al.[11], which were the largest and most heterogeneous publicly available datasets, respectively, plus our legacy dataset and the crowdsourced dataset both with and without proofreading. We found that the Heinrich et al. dataset was not useful for our benchmark volumes, likely because of its small size and limited breadth. The model trained on it achieved F1 scores of zero at all IoU thresholds and the highest IoU score was 1 x 10^{-3} on the Lucchi++ benchmark. The model trained on MitoEM achieved respectable – but expected since they were similar neuronal data – F1@75 scores of 0.86 and 0.75 on the fly brain and Lucchi++ volumes, but scores of 0, 0.07, and 0.01 on the C. elegans, HeLa cell and glycolytic muscle volumes. The average performance metrics over all benchmarks for MitoNet versions trained on various annotated datasets is shown in Table 2.

Our legacy dataset, described above, was 5x larger than the crowdsourced dataset and reasonably heterogeneous with data from 17 volumes, over 500 TEM images and over 2,000 small 2D patches from CEM500K. Still, we observed that the model trained on this dataset had slightly lower F1 and IoU scores than the model trained on the expert proofread crowdsourced data (which excluded images in Group 1a). This surprising finding underscores how critical data diversity is to training generalist models. A dataset of nothing but small 2D patches sparsely sampled from numerous datasets was more effective on our benchmarks than any of the much larger but somewhat less heterogeneous datasets. As a corollary to this finding, we observed that the student consensus annotations, despite over 50% requiring some expert correction, yielded a more general model than training on the MitoEM dataset. This suggests that even noisy and inaccurate, but heterogeneous, labeled data contains enough training signal for a model to learn a better representation of mitochondrial ultrastructure than could be achieved from large, high-quality but homogeneous data.

Discussion

In this paper we have reported several resources for the growing volume EM community: a highly heterogeneous ~1.5 x 10^6 unlabeled cellular EM image dataset (CEM1.5M), a well-described, large and heterogeneous dataset with instance labeled mitochondria (CEM-MitoLab), a model trained on this dataset (MitoNet), a set of diverse benchmarks to test this and future models, and a Python package and napari plugin (empanada) for immediate use of MitoNet. Increases in throughput enabled by technological advances and the application of volume EM techniques to a wider range of biological systems have exacerbated the segmentation and data analysis bottleneck in the field. Recent DL-based results have been spectacular in narrow contexts, but researchers risk getting stuck in the cycle of "manually label and train a model on a sub-volume, run inference and polish on the full volume". The models that result from such workflows generalize poorly to unrelated datasets and are usually obsolete as soon as the relevant segmentation is completed. With the ultimate aim of breaking out of this cycle and creating reusable tools that benefit the community, we show that very broad but shallow sampling of cellular contexts is a robust strategy to create general organelle segmentation models. We present these datasets for use as-is or for future expansion or versioning for specialized tasks. To this end, the organization and appropriate description of datasets is crucial. We have created a simple and practical implementation of REMBI protocols[55] in the form of a spreadsheet (**Supplementary File 1)** and have filled in metadata fields at minimum corresponding to image descriptors and basic biological information, so that users can search this release or future iterations of the dataset using metadata terms.

We made the important decision to forego 3D approaches, knowingly trading accuracy for generalization and efficiency. Native 3D models are expected to perform better for instance segmentation because of their expanded spatial context. However, it is exceedingly time-consuming to generate sufficient annotation datasets to adequately train a model. For example, our crowdsourced dataset of ~6,000 2D labeled images is equivalent to roughly 30 3D labeled images of the same size (224 pixels to a side). Choosing to create such a 3D dataset would have left a great variety of contexts unsampled and would have excluded thousands of TEM and STEM images. Here we attempted to thread the needle. We collected annotations in 2D but gave annotators the extra context of a flipbook. We ran model inference in 2D but developed instance matching, median filtering, and ortho-plane techniques to propagate and incorporate 3D information. Moving beyond 2D will likely be necessary to achieve human-level annotation quality on some volumetric datasets. Indeed, one use-case of MitoNet is that it can be incorporated into human-in-the-loop workflows to rapidly produce 3D segmentations needed to train this next-generation of models.

Regardless of dimensionality, our results demonstrate that heterogeneity is a central characteristic for a training dataset. The version of MitoNet trained on noisy and often inaccurate crowdsourced annotations from students was measurably better on our benchmarks than the equivalent model trained on the much larger and expertly labeled MitoEM dataset (**Figure 5d**). Still, proofreading the annotations for accuracy was necessary to achieve our best results. Getting high-quality crowdsourced segmentations without expert intervention is challenging, even with powerful tools like Zooniverse. Accumulating the knowledge necessary to recognize mitochondria across many tissues, organisms and sample preparation protocols takes time and training. Relatively few experienced annotators (3-5) are enough to create strong consensus segmentations, meaning that broader expert participation in shared segmentation efforts would be beneficial to the field.

The value of training data heterogeneity is only apparent when the test data is also heterogeneous. There are few volume EM benchmarks, and those derived from connectomics data are all quite similar to each other; possibly the best known of these, Lucchi++[25,40] has now been mined to the point that it is difficult to track improvements in DL performance[25,53,56], and until this report, only one benchmark, MitoEM, provided instance segmentations. Our benchmark with six instance segmented volumes aims to correct these deficiencies. It is still just a small sampling of the entire cellular landscape, but appears to be the most stringent test available, with a range of mitochondria, contexts, quality and overall difficulty. We suggest that future benchmarks should comprise relatively small ROIs from many volume EM datasets to best test generalization. Training and test datasets must also evolve to accurately measure progress of automated solutions and prevent misleading reports of success.

Finally, at least within the volume EM field, we are acutely aware of the gap between research groups that have ample resources and computational expertise, and smaller labs that do not. Most groups deal with small, discrete, and one-off 3D reconstructions, so it is important to have tools that are relatively easy to use and not compute heavy. We designed our Python package, empanada, and the corresponding napari plugin, with this in mind. Empanada and the MitoNet model are optimized for compute and memory efficiency and our plugin is easy to install and use. Additionally, it can run on consumer-grade laptops without GPUs or on HPC clusters with many. MitoNet can be incorporated into other established image analysis platforms and will ideally give users a better model that can be quickly deployed or finetuned to meet their segmentation needs. We did not focus on trying to get the absolute best architecture possible for MitoNet and we hope and expect others will supersede our results. Rather, we used an efficient architecture and postprocessing scheme that scales well to large images and images with very many objects, as would be expected for mitochondrial instance segmentation in EM.

Volume EM has been suggested to be in the midst of a “quiet revolution” in cell biology[57]. A raft of papers in connectomics and in cell biology have revealed insights into a variety of systems by enabling the high-resolution 3D imaging of cellular and subcellular features. These imaging techniques record ultrastructural information in a largely unselective manner, meaning that the data, once collected, need to be parsed out to extract features of interest for visualization and analysis. This step, segmentation, must be accurate and precise to faithfully represent the underlying biology and it must also be efficient to match the speed of data generation. Recent developments in DL approaches show great promise, yet these tools can be held back by the lack of large-scale and relevant data resources. In this report we directly address this problem as it pertains to universal segmentation of mitochondria in EM images. With an eye towards creating a universal mitochondrial model, we employ a strategy of sparse sampling of widely heterogeneous datasets and show that the resulting model, MitoNet, when trained on these data, yields promising results on challenging tasks. We release the resources to the community with the hope that continued work on this approach will expand the EM segmentation toolkit and further accelerate discoveries in this exciting field.

Materials and Methods

Unlabeled dataset creation

The expansion of CEM500K to create the CEM1.5M dataset followed the data standardization and curation protocols presented in our previous work[58]. External datasets were either downloaded in their entirety or, for large datasets stored online in next generation file formats (n5 or zarr), accessed either with the CloudVolume or fibsem-tools APIs. Any datasets larger than 5 GB were randomly cropped into 320 cubes of 256x256x256 (equivalent to 5 GB). When available, to ensure that the randomly chosen crops included cellular content and not empty image padding or resin, a low-resolution overview image volume of the dataset was downloaded and used to identify the extents of informative ROIs. Lastly, cellular EM images from videos were converted to mrc files after removing all frames that showed non-EM content. Metadata, and proper attribution, for each dataset following REMBI[59] standards is available in **Supplementary file 1.**

In-house 2D EM images were collected from the National Cancer Institute's Electron Microscopy Laboratory database. The database included over 2x10^5 TEM images. A sample of 5,000 images with magnifications between 1000x (~16 nm) and 3000x (~6 nm) that excluded negative stain and immunogold label images was randomly selected. This sample represented the output of collaborations with 76 NIH investigators from over the last 12 years. Metadata for each image was limited. Therefore, images were grouped by investigator before further sampling. Additionally, seven more groups of 2D STEM images were added from previous in-house imaging projects; because metadata was available for these images they were grouped by biological context and not investigator.

Zooniverse Workflow

Our Zooniverse workflow closely approximated the Etch-a-cell project[15].Patches in the CEM1.5M dataset extracted from volume EM images were reconstructed into short "flipbooks" (tif stacks of five consecutive images of 224x224), where each patch from the CEM dataset was the center image (the adjacent patches were typically filtered out by the CEM deduplication pipeline). Before uploading to Zooniverse each flipbook had its 8-bit intensity range rescaled from 25 to 235 and was interpolated to a size of 480x480 for easier viewing. For 2D images, crops of 512x512 were used. Intensity was rescaled to the same range as the flipbooks but no resizing was performed.

The first two sets of images uploaded to Zooniverse (Group 1a) had an annotation retirement limit of five (i.e., every image was annotated independently by five people) and were annotated by a mixture of lab members familiar with cellular EM and inexperienced student annotators. Consensus annotations from these sets only used annotations from experts; the students' feedback on the project helped refine the Zooniverse tutorials and interface. For all remaining sets, which were annotated entirely by an expanded cohort of students, the retirement limit was set to ten. For the first two of these sets, 5-10% of images were annotated by a team of at least three EM experts. These "gold-standard" annotations were used to measure the performance of the student annotators and to provide them with training examples via a Google CoLab notebook. Later, as the students' annotations became more accurate, the non-proofread consensus annotations were shared. In addition to this, feedback via the Zooniverse message channel and a formal instruction session was used to review challenging examples and explain common mistakes.

Consensus annotation algorithm

The input to the algorithm was a set of **N** annotations (the retirement limit) containing **K** mitochondrial detections. An undirected graph was initialized where each node corresponded to one of the **K** detections. Pairwise bounding box IoU scores were calculated for all detections to form a **KxK** matrix. For all pairs of detections with non-zero bounding box intersections, pixel-level mask IoU scores were calculated. Edges were added to the graph to connect detection nodes with mask IoU scores that exceeded a small threshold value of 0.1 (the same algorithm, when applied during ortho-plane inference, used a threshold of 0.01). The resultant connected components in the detection graph were processed independently to determine the consensus object instances.

Nodes in a connected component were organized into cliques where all detections within a clique had IoU scores greater than 0.75. An iterative algorithm was then applied to determine whether cliques should be merged or remain split. First, the clique whose detections shared the most edges with other cliques, i.e. the most connected clique, was selected (node C in Supplementary Figure 3 ii, A in 3 iii). Second, if any of the most connected clique's neighbors contained more detections, then the most connected clique was dissolved and its detections were pushed out to each of its neighbors (Supplementary Figure 3 iv). Otherwise, all the most connected clique's neighbors were dissolved and their detections were pulled in by the most connected clique (between B and D in Supplementary Figure 3 iv, and B and C into A in 3 v). These two steps were repeated until no cliques had outgoing edges. Each clique represented a segmented object instance. The detection masks within a clique were added together to form an image where each pixel had a value from 1 to **N** denoting the number of annotators who labeled it. A vote threshold of 3 when the retirement limit was 5 or 5 when the retirement limit was 10 was applied to create the final binary instance mask. In cases where there was weak instance-level consensus between annotators, the binary instance masks from separate cliques within the same connected component could have non-trivial overlaps with each other. If these overlaps resulted in IoU scores greater than 0.1 between binary instance masks, then those masks were combined into a single mask. In the final consensus instance segmentation, all non-zero binary masks were assigned new labels and merged. A schematic of this algorithm is shown in **Supplementary Figure 3.** In summary, this algorithm assumed that the most connected clique at each step roughly corresponded to a maximally merged instance. When one of this clique's neighbors contained more detections, this implied that more votes were in favor of splitting than maximal merging. Pushing the most connected clique to its neighbors ensured that no detections were deleted before being counted towards the final pixel-level majority vote. Conversely, when the most connected clique had the most detections, this implied that more votes were in favor of maximal merging and therefore all detections in neighboring cliques were pulled into the most connected clique.

Proofreading

Once all images in a set were retired, a consensus strength and instance segmentation were calculated for each. The consensus strength was the average F1@50 score of each individual annotation relative to the consensus instance segmentation. Lower consensus strengths indicated greater disagreement between annotators. Next, all images and instance segmentations were ordered by consensus strength and stacked into a single tif file for later proofreading. If the set used flipbooks, then the images and their consensus instance segmentations were resized from 480x480 back to their original sizes. Padding was added as needed to ensure that all images in the stack had the same dimensions. Typically, the single proofreading stack was split into chunks of 40-50 images/flipbooks such that multiple experts could perform proofreading in parallel. Proofreaders were instructed to emphasize the correction of false positive and false negative detections over fine-grained instance boundary corrections. Annotations in each chunk of images were verified by a second proofreader and disagreements were discussed and resolved by at least three proofreaders. 3D Slicer[60] was used to visualize and correct annotations. The chunks of proofread images and annotations were again combined into a single large stack. Every image, or every third image for flipbooks, and corresponding instance segmentation was stored as a tiff image with any padding cropped out. Importantly, all annotations were passed through a connected components filter to guarantee that there were no disconnected instances.

Training dataset creation

The training dataset included all consensus annotations gathered from Zooniverse as well as the legacy data. The latter included annotations from publicly available mitochondrial segmentation benchmarks (Kasthuri++[25,61], Guay[30], UroCell[62], MitoEM[26], Heinrich et al.[11] and Perez et al.[29]) and previous in-house projects. Of the benchmark datasets, only the MitoEM benchmark had individual instances labeled; instance segmentations for all others were generated manually after applying a connected components filter. In-house annotations for various unrelated projects included 11 volume EM image reconstructions (see **Supplementary File 1**) as well as 529 TEM images and 2,231 image patches from the CEM500K dataset. None of these images were derived from any of the above, or from benchmark datasets. The 11 volume EM images had previously been annotated by a different deep learning model and then manually corrected, while all the 2D images were manually annotated from scratch. Legacy volume EM images and instance segmentations were cropped into patches of 512x512 and passed through the deduplication and filtering pipeline from CEM500K[58] to remove redundant and uninformative patches. Patches from isotropic voxel volumes were taken from xy, xz and yz planes while patches from anisotropic volumes were taken from xy only. For external data comprising the legacy datasets, all images were used as-is, except: prior to cropping patches, the MitoEM-H and MitoEM-R volumes were binned by a factor of two in x and y (from 8 nm to 16 nm pixels), and all Heinrich et al. images were downloaded at 8 nm resolution.

Benchmark dataset creation

Four of the six benchmark volume EM datasets were independently annotated by ariadne.ai (HeLa cell, C. elegans, fly brain, salivary gland). The glycolytic muscle[35] dataset was binned by a factor of 2 in all dimensions (from 9 nm to 18 nm voxels) and then automatically annotated by a deep learning model that was trained solely on patches from the volume, and then manually corrected. The remaining benchmark is derived from the Lucchi++[25,40] benchmark. First, the 2D images and labelmaps were stacked to form volumes. Then, the binary mitochondrial labelmaps were converted to rough instance segmentations by applying a connected components filter and then manually corrected to derive the final instance segmentation.

Not only were all benchmark volumes excluded from both CEM1.5M and crowdsourced annotation, **we also excluded** any images related to them to rigorously test generalization, **Supplementary Figure 6.** OpenOrganelle[39], which was the source of the fly brain benchmark, included multiple volumes of fly brain tissue from different brain areas. We chose the “Drosophila brain: Fan-shaped body” volume and excluded the “Drosophila brain: Entire alpha and alpha' lobes of a mushroom body” and “Drosophila brain: Accessory calyx” volumes. For the C. elegans volume, a related FIB-SEM and STEM dataset were excluded. And lastly, for the Lucchi++[25,40] benchmark, the Lucchi++ training set was excluded.

Benchmark datasets measurements

Mitochondrial volumes were calculated by trivially counting the number of voxels per instance. Minimum distances between neighboring mitochondria were calculated by computing the distance transform of each instance's complement (i.e., set of all voxels not inside the instance). These results were stored in a new volume (called "distance volume") where each voxel was updated to the minimum distance calculated so far from all the measured complements at that location. The minimum distance to a mitochondrion's nearest neighbor was then the minimum value enclosed by that instance in the distance volume. Branch lengths and cross-sectional diameters were calculated by first skeletonizing[63] each mitochondrion and computing its distance transform. Branches shorter than 60 nm were pruned. Lengths were simply the number of voxels in each branch, and the mean cross-sectional diameter was the average of all values in the distance transform that overlapped with the skeleton.

CEM1.5M Pretraining

Unsupervised pretraining followed the SwAV algorithm[64]. A ResNet50[65] model was trained for 200 epochs with a batch size of 256. All other hyperparameters used the default values defined in **https://github.com/facebookresearch/swav.** Image augmentations included 360-degree rotations, randomly resized crops (following SwAV defaults), brightness and contrast jitter, random Gaussian blur and noise, and horizontal and vertical flips. To correct for the imbalance in the number of patches per dataset, weighted random sampling was applied. Weights per dataset were calculated by:

$$w_i = \frac{{n_i}^{-\gamma}}{\sum_{j=1}^{N} w_j}$$

γ was a float from 0 to 1, n_i was the number of patches from the i^{th} dataset, and N was the total number of datasets. Each patch was assigned a sampling weight based on its source dataset. When $\gamma = 0$, patches from all datasets were sampled with the same probability (true random sampling) and when $\gamma = 1$ the sampling was perfectly balanced between datasets. During pretraining we set $\gamma = 0.5$ (i.e., the square root of the number of patches per dataset).

Deep learning model architecture

The model was based on Panoptic-DeepLab[42] (PDL). In brief, PDL is an encoder-decoder style network that uses atrous spatial pyramid pooling (ASPP) to integrate features that occur over multiple scales. A ResNet50 encoder, pretrained as detailed above, was used in all experiments unless otherwise noted. To preserve more spatial information, strides in the last downsampling layer were replaced with dilation such that the output from the encoder was 16x smaller than the input (compared to 32x smaller with strides). The baseline model configuration included two ASPP modules and decoders; one for semantic segmentation prediction and the other for instance center and offsets prediction. Since input patches used during training were just 256x256, the dilation rates in the ASPP modules were set to 2, 4 and 6 with 256 channels per convolution and dropout with probability 0.5 applied to the output. The semantic and instance decoders each had a single level at which the output from the first layer in the encoder (4x smaller than the input image) was fused via depthwise separable convolution with the interpolated output from the ASPP module. Following the standard for PDL, the convolution channels in the instance decoder were half of those in the semantic decoder (16 compared to 32). The semantic segmentation, instance center and offset heads used a single depthwise separable convolution with kernel size of 5.

PointRend: During training, the semantic segmentation logits predicted by PDL were upsampled by a factor of 4 to the original image resolution and then refined for a single step by the PointRend[66] module. For a batch of images, a set of 3,072 points were randomly sampled from anywhere within the dimensions of the images and the segmentation logits were evaluated at these points. From these sampled logits, 1,024 were selected for each image where 25% of the logits were randomly chosen and the other 75% were the logits with the smallest absolute values. The same 1,024 points were also used to sample features from the semantic decoder output. Sampled features and logits were concatenated and fed through a three-layer fully connected network. When evaluating the loss, the 1,024 points were also used to sample the ground truth semantic segmentation. Loss was calculated as the (binary) cross-entropy between the sampled and refined logits and the sampled ground truth. During evaluation, two refinement steps were applied with the semantic segmentation logits being upsampled by a factor of 2 at each step. Points corresponding to the center of all logit pixels were used to sample the logits. From these, points corresponding to the 8,192 logits with the smallest absolute values were selected. The logits and semantic decoder output were sampled at these points and passed through the fully connected network. The refined logits then replaced the unrefined logits in the upsampled segmentation. Since the number of sampled points was fixed, the cost of running PointRend refinement was approximately constant with respect to the size of the input.

Model training parameters

The overall loss function during training was:

$$\mathcal{L} = \mathcal{L}_{sem} + \alpha * \mathcal{L}_{center} + \beta * \mathcal{L}_{offset} + \gamma * \mathcal{L}_{pr}$$

α, β, γ were constants used to weight the relative contributions of each loss and were set to 200, 0.01 and 1. $\mathcal{L}_{sem}$ was the semantic segmentation loss computed as the bootstrapped (binary) cross-entropy where only the top 20% of largest cross-entropy values were averaged across a batch. $\mathcal{L}_{center}$ was the instance center regression loss and $\mathcal{L}_{offset}$ was the center offset loss calculated as mean squared error and absolute error (L1), respectively. $\mathcal{L}_{pr}$ was the PointRend loss calculated as the (binary) cross-entropy.

All models were trained using the One Cycle learning rate policy[67] with AdamW[68]. The max learning rate was set of 0.003 with weight decay of 0.1 and momentum was cycled from 0.85 to 0.95. Learning rate warmup lasted for the first 30% of training epochs. Weighted sampling of patches from datasets was implemented as above ($\gamma = 0.3$ here) to correct for overrepresented datasets in CEM-MitoLab. Image augmentations included large scale jitter[69], where images were zoomed in by a maximum of 2x and zoomed out by a maximum of 10x, random cropping to a 256x256 patch, 360-degree rotations, brightness and contrast adjustments and vertical and horizontal flips. Like CEM1.5M pre-training, images were selected from datasets based on weighted sampling. Unless otherwise noted, $\gamma = 0.3$ was used for all training runs.

Panoptic-DeepLab Postprocessing

First, values in the instance center heatmap that were less than the center confidence threshold (0.1 in all experiments) were zeroed. Non-maximum suppression with a kernel size of 7 was applied to filter out peaks in the heatmap that occurred within the same 7x7 grid. The non-zero coordinates that remained in the heatmap were extracted. Next, the instance center offsets were added to their absolute (x, y) positions within the image such that all pixels belonging to the same object ought to be equal to the coordinate that defines the center of that object. The Euclidean distance between this map of each pixel's corresponding object center and the centers extracted from the heatmap were used to group pixels into instances. The index of the center with the minimum distance at each pixel was adopted as the instance label for that pixel. After applying the sigmoid activation to get probabilities from the semantic segmentation logits and hardening the result over a threshold, the instance labels and semantic segmentation were merged to form the final panoptic segmentation.

Stack instance matching

To match object instances between consecutive slices in a 2D stack, intersection-over-union (IoU) scores were first calculated between instance bounding boxes. Second, for each pair of instances with non-zero bounding box IoU, a mask IoU was calculated and stored in a matrix of size KxN, where K is the number of instances in slice j and N is the number of instances in slice j+1. At this stage, mask intersection-over-area (IoA) scores were also calculated and stored in a separate KxN matrix. The Hungarian algorithm[47] was then applied to the mask IoU matrix to determine the assignment of instances in slice j to instances in slice j+1 that maximized the total IoU. Any unassigned instances in slice j+1, or assigned instances with IoU less than a threshold value (IoU threshold), were considered unmatched. Unmatched instances with a IoA score greater than a threshold value (IoA threshold) were matched to the instance in slice j with which they shared the highest IoA. The remaining unmatched instances were assigned new labels in the stack. After forward matching (i.e., from slice j to slice j+1) was completed for all images in the stack, a backward pass of matching was conducted (i.e., from slice j to slice j-1). No new labels were assigned in the stack during the reverse pass.

Benchmark Set Inference

Ortho-plane inference[48] on xy, xz and yz planes was applied to each volume in the benchmark test set. During inference over each plane, semantic segmentations were generated from the median probabilities within a short queue of previous and subsequent segmentation results. A queue length of 3 was used for the C. elegans and fly brain volumes, 5 for the HeLa and glycolytic muscle volumes and 7 for the Lucchi++ and salivary gland volumes. These queue lengths roughly track with voxel size (in nm) such that smaller voxels corresponded to longer queues. Median probabilities were hardened at a confidence threshold of 0.3. Since mitochondria were never occluded in any of our benchmark volumes, mitochondrial instances on each 2D slice were required to be connected components. Instance matching across slices used IoU and IoA thresholds of 0.25. After a forward and backward matching pass, a simple size and bounding box extent filter were applied to eliminate likely false positives. The minimum object size was set at the 5th percentile of mitochondrial volume for each benchmark while the minimum bounding box extent was fixed at eight voxels for all datasets. Finally, the same algorithm that was used to generate consensus segmentations during crowdsourced annotation was applied to ensemble the three segmentation stacks created by ortho-plane inference **(Supplementary Figure 3)**. A clique IoU threshold of 0.75 was used for all volumes and a vote threshold of 2 out of 3 was used for all benchmarks except the C. elegans and salivary gland volumes. For those a vote threshold of 1 out of 3 was used. This was in response to the relatively low consensus strength between each stack and the final ortho-plane result. Lowering the threshold gave a significant increase in performance on the C. elegans volume and a modest increase for the salivary gland volume.

Salivary gland finetuning

The salivary gland volume was first cropped into patches of 224x224 and then deduplicated and filtered (see *Training dataset creation* section above). 64 patches (~0.2% of the volume) were randomly selected from the filtered subset and set aside as the finetuning training set. The best performing segmentation model was retrained on the finetuning training set for 500 iterations using the same parameters outline in the *Model training parameters* section above.

Benchmark Evaluation

Ground truth instance segmentations were stored in json files where each entry contained an instance id, bounding box and run-length encoded segmentation. Equivalent json files were created for all predictions. Instances between the ground truth and prediction were assigned as matches using the Hungarian algorithm to maximize IoU over all possible pairs of instances. In addition to the F1, IoU, and PQ scores reported in results, average precision (AP) scores for future comparison.

Test Set Evaluation by Training Dataset

To prepare datasets for fair comparison, the MitoEM and Heinrich et al. ground truth ROIs were sliced to 2D images. Since the MitoEM volumes were large and anisotropic, patches of size 512x512 were cropped from xy slices only. The Heinrich et al. volumes being small and isotropic were cropped into patches of 224x224 or smaller from xy, xz and yz planes. As above, mitochondrial instances were relabeled on each 2D slice to guarantee that they were connected components. All models were trained for approximately 10,000 iterations.

Acknowledgements

This project has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institutes of Health, under Contract No. 75N91019D00024. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. This publication uses data generated via the Zooniverse.org platform, development of which is funded by generous support, including a Global Impact Award from Google, and by a grant from the Alfred P. Sloan Foundation. We thank Helen Spiers, Martin Jones and Lucy Collinson for help with Zooniverse project set up. We thank the WHK-SIP program and Batch of 2021 student annotators, with special thanks to student leaders Ella Fitzgerald and Taeeun Kim. Valentina Baena, Adam Harned, Kunio Nagashima and Heather Berensmann helped perform proofreading. Patrick Friday helped curate metadata, and Aayush Bhatawadekar contributed to renderings and plugin development. Finally we appreciate the volume EM community, especially the Data Working Group for general discussions in this space. This work utilized the computational resources of the NIH HPC Biowulf cluster. (http://hpc.nih.gov).

Figure Legends

Figure 1: Creation of a diverse and representative dataset for mitochondrial instance segmentation. a. Schematic of the data curation pipeline. Volume EM reconstructions and 2D EM images were curated[58] to create CEM1.5M, an unlabeled dataset of ~ 1.5 x 10^6 image patches used for self-supervised pretraining. Approximately 6,000 randomly sampled patches from CEM1.5M (green) were uploaded to the Zooniverse platform for crowdsourced annotation. These, and 16,000 randomly selected patches from previously labeled data (legacy annotations, red) were combined to form the supervised mitochondrial training dataset CEM-MitoLab. **b.** Lorenz plots for CEM1.5M (blue, Gini coefficient = 0.802), crowdsourced data (green, 0.209), legacy data (red, 0.840), CEM-MitoLab (dashed gray, 0.686). Black line, perfect equality in distribution (Gini = 0). **c.** Distribution of imaging plane pixel sizes for volume EM images in CEM-MitoLab. The dashed lines denote pixel sizes for 2D EM images in the dataset. **d.** Imaging technique, **e.** Source organism, **f.** Source tissue (vertebrates only, *in vitro* cells grouped under Not Defined) of datasets in CEM-MitoLab (n=478).

Supplementary Figure 1: a. Distribution of longest image side in pixels for patches in the supervised mitochondrial dataset, CEM-MitoLab. **b.** Breakdown of isotropic versus anisotropic data (left) and sample fixation method (right) in source datasets that make up CEM-MitoLab **c.** Random selection of image patches from CEM-MitoLab (left), Mito-EM (middle) and Heinrich et al **d.** Comparison between 2D mitochondrial instance label maps in CEM-MitoLab (blue), MitoEM (purple) and Heinrich et al. (pink) labeled datasets, with respect to (left to right) area in pixels, contrast, eccentricity, perimeter to area ratio, and clustering.

Figure 2: Crowdsourced annotation of CEM-MitoLab. a. Ground truth (top left), consensus annotation (bottom left) and ten independent student annotations of an image showing high degree of consensus. **b.** Instance segmentation quality measured by F1@50 over time. Left, Group 1a (expert annotation), 1b, 2, 3 (students). Blue dots, individual scores, pink, consensus score, box denotes median, 25th and 75th percentile. Right, change in individual annotator performance over time. **c-f** Instance segmentation quality measured by F1@50 plotted against **c.** Increase in retirement limit. Blue, purple, student scores (retirement set at 10); green, expert

score (retirement set at 5), **d.** Vote threshold. Blue, F1 per instance; pink, F1 per image; dashed line, vote threshold of 5 chosen for consensus calculation, **e.** Self-reported confidence score over all instances (connected line, average F1), **f.** Student mean confidence score, **f.** Number of images annotated by each students.

Supplementary Figure 2: a. Screenshots from Zooniverse project, showing the crowdsource annotation user interface **b.** Total count of instance False Positive and False Negative errors made by student annotators. **c.** Ground truth (top left), consensus annotation (bottom left) and ten independent student annotations of an image showing a low degree of consensus; the all-or-nothing pattern suggest differences in individual knowledge or experience. **d.** Trends of true positive (blue), false positive (red), false negative (green) for total instances of mitochondria, plotted against vote threshold. A threshold of 5 was chosen for consensus calculation.

Supplementary Figure 3. Example of consensus algorithm splitting and merging instance votes to create accurate consensus label maps. See Materials and Methods for details.

Figure 3: Challenging and diverse benchmarks for evaluating automatic instance segmentation performance. a. 2D representative images (left) and 3D reconstructions (right) for the benchmark test sets. From top to bottom: C. elegans, Fly brain, HeLa cell, Glycolytic muscle, Salivary gland, Lucchi++. Yellow arrow, membranous organelle, orange and blue arrows, lightly and darkly stained mitochondria, green arrow, heavy metal precipitate, red arrow, mitochondrion and tightly apposed salivary granule in the acinus **b.** Comparison of individual mitochondria and box plots across benchmarks by (top to bottom): volume, branch length, cross-section radius, minimum distance to neighbor (all in voxels) and contrast. Pink, C. elegans n=241; yellow, Fly brain n=91; green, HeLa cell n=68; teal, Glycolytic muscle n=104; blue, Salivary gland n=131; purple, Lucchi++ n=33.

Figure 4: Deep learning model and postprocessing pipeline to create 2D or 3D instance segmentations. a. Schematic of Panoptic-DeepLab. A grayscale image (left) is passed through the model architecture: blue boxes denote outputs from the encoder layers, black boxes output from atrous spatial pyramid pooling (ASPP) layers, gray boxes output from decoder layers.

Outputs of the network are (left to right) semantic segmentation, up-down offsets, left-right offsets and instance centers. Far right, instance segmentation created from the outputs. **b.** Instance matching across adjacent slices with partially overlapping segmentation uses intersection-over-union (IoU) and intersection-over-area (IoA) scores. Top row: predicted segmentation of slice j (left), j+1 (right), bottom row: IoU only merging (left), IoU and IoA merging (right). **c.** Merging of labels with completely missing segmentation uses a median prediction over 3-7 stacks, in direction of black arrow, depending on size of gap **d.** From top to bottom: Stacked 2D segmentations before matching, after forward matching only and after forward and backward matching, to merge falsely split mitochondria. Black arrows denote direction of matching. **e.** An example of 3D instance segmentation of mitochondria after running inference in (left to right) xy, xz, and yz directions, and far right, merged instance segmentation after combining inferences from orthogonal planes.

Supplementary Figure 4. Example image showing difference in semantic segmentation borders of closely apposed mitochondria, after linear interpolation (top) or PointRend (bottom). Blue dots, PointRend sampling locations.

Figure 5. MitoNet results on benchmarks. a. Representative 2D images showing MitoNet segmentation performance. Top to bottom: C. elegans, Fly brain, HeLa cell, Glycolytic muscle, Lucchi++, Salivary gland before model finetuning, Salivary gland after model finetuning. **b.** Representative ground truth and predicted segmentations from MitoNet. Red and green, predicted mitochondrial instances, blue and orange, ground truth instances. Blue arrow, highly branched mitochondrion, black arrow, example of segmentation expected to return a high IoU but low F1 score. **c.** Left, MitoNet F1score on each of the benchmarks as a function of IoU threshold; right, IoU scores **d.** Left, comparison of mean F1 score for models trained on different datasets plotted against IoU threshold, right, comparison of mean IoUs achieved by models trained on various datasets. All benchmarks except the salivary gland are included.

Supplementary Figure 5. Analysis of True Positive and False Negative detections by MitoNet on benchmarks, grouped by different mitochondrial attributes. Top to bottom: mitochondrial volume, minimum distance to nearest neighbor, branch length, branch mean cross-sectional diameter (measurements by voxel) and mitochondrial contrast. Branch length and cross section plots exclude the Lucchi++ benchmark, as the branches for these mitochondria were shorter than the pruning threshold. Fine tuned model results are plotted for salivary gland.

Supplementary Figure 6. Representative images of datasets that were excluded from CEM1.5M and CEM-MitoLab datasets, in order to maintain integrity of the test set. Images to the left of the black line are volumes in the test set (fly brain, C. elegans and Lucchi++). Images to the right show datasets that were excluded because they were considered too similar to the benchmarks.

Supplementary Figure 7. Screenshot of the napari plugin empanada.

Table 1: Performance metrics of MitoNet across benchmarks. AP, Average Precision, PQ, Panoptic Quality

Table 2: Average performance of MitoNet versions pre-trained on CEM1.5M but trained on various annotated datasets

Supplementary Table 1: Average mitochondrial measurements for true positive, false negative MitoNet predictions across various benchmarks.

References

1. Peddie, C. J. & Collinson, L. M. Exploring the third dimension: Volume electron microscopy comes of age. *Micron* **61**, 9–19 (2014).
2. Titze, B. & Genoud, C. Volume scanning electron microscopy for imaging biological ultrastructure. *Biol. Cell* **108**, 307–323 (2016).
3. Scheffer, L. K. *et al.* A connectome and analysis of the adult drosophila central brain. *Elife* **9**, 1–74 (2020).
4. Turner, N. L. *et al.* Reconstruction of neocortex: Organelles, compartments, cells, circuits, and activity. *Cell* **0**, 17 (2022).
5. Yin, W. *et al.* A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nat. Commun. 2020 111* **11**, 1–12 (2020).
6. Januszewski, M. *et al.* High-precision automated reconstruction of neurons with flood-filling networks. *Nat. Methods* **15**, 605–610 (2018).
7. Berning, M., Boergens, K. M. & Helmstaedter, M. SegEM: Efficient Image Analysis for High-Resolution Connectomics. *Neuron* **87**, 1193–1206 (2015).
8. Abdollahzadeh, A., Belevich, I., Jokitalo, E., Sierra, A. & Tohka, J. DeepACSON automated segmentation of white matter in 3D electron microscopy. *Commun. Biol. 2021 41* **4**, 1–14 (2021).
9. Dorkenwald, S. *et al.* Automated synaptic connectivity inference for volume electron microscopy. *Nat. Methods 2017 144* **14**, 435–442 (2017).
10. Funke, J. *et al.* Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome Reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**, 1669–1680 (2019).
11. Heinrich, L. *et al.* Whole-cell organelle segmentation in volume electron microscopy. *Nat. 2021 5997883* **599**, 141–146 (2021).
12. Guay, M., Emam, Z., Anderson, A., Aronova, M. & Leapman, R. Dense cellular segmentation using 2D-3D neural network ensembles for electron microscopy. *bioRxiv 2020.01.05.895003* (2020) doi:10.1101/2020.01.05.895003.

13. Žerovnik Mekuč, M. *et al.* Automatic segmentation of mitochondria and endolysosomes in volumetric electron microscopy data. *Comput. Biol. Med.* **119**, 103693 (2020).

14. Liu, J. *et al.* Automatic Reconstruction of Mitochondria and Endoplasmic Reticulum in Electron Microscopy Volumes by Deep Learning. *Front. Neurosci.* **14**, 599 (2020).

15. Spiers, H. *et al.* Deep learning for automatic segmentation of the nuclear envelope in electron microscopy data, trained with volunteer segmentations. *Traffic* **22**, 240–253 (2021).

16. Glancy, B., Kim, Y., Katti, P. & Willingham, T. B. The Functional Impact of Mitochondrial Structure Across Subcellular Scales. *Frontiers in Physiology* vol. 11 1462 (2020).

17. Vincent, A. E. *et al.* The Spectrum of Mitochondrial Ultrastructural Defects in Mitochondrial Myopathy. *Sci. Rep.* **6**, 1–12 (2016).

18. Vincent, A. E. *et al.* Quantitative 3D Mapping of the Human Skeletal Muscle Mitochondrial Network. *CellReports* **26**, 996-1009.e4 (2019).

19. Pernas, L. & Scorrano, L. Mito-Morphosis: Mitochondrial Fusion, Fission, and Cristae Remodeling as Key Mediators of Cellular Function. *Annu. Rev. Physiol.* **78**, 505–531 (2016).

20. Delgado, T. *et al.* Comparing 3D ultrastructure of presynaptic and postsynaptic mitochondria. *Biol. Open* **8**, (2019).

21. Stoldt, S. *et al.* Spatial orchestration of mitochondrial translation and OXPHOS complex assembly. *Nat. Cell Biol.* **20**, 528–534 (2018).

22. Meyer, J. N., Leuthner, T. C. & Luz, A. L. Mitochondrial fusion, fission, and mitochondrial toxicity. *Toxicology* **391**, 42–53 (2017).

23. Zhang, L. *et al.* Altered brain energetics induces mitochondrial fission arrest in Alzheimer's Disease. *Sci. Rep.* **6**, 1–12 (2016).

24. Siegmund, S. E. *et al.* Three-Dimensional Analysis of Mitochondrial Crista Ultrastructure in a Patient with Leigh Syndrome by In Situ Cryoelectron Tomography. *ISCIENCE* **6**, 83–91 (2018).

25. Casser, V., Kang, K., Pfister, H. & Haehn, D. Fast Mitochondria Segmentation for Connectomics. *arXiv1812.06024 [cs]* (2018).

26. Wei, D. *et al.* MitoEM Dataset: Large-scale 3D Mitochondria Instance Segmentation from

EM Images. *Med. Image Comput. Comput. Assist. Interv.* **12265**, 66 (2020).

27. Müller, A. *et al.* 3D FIB-SEM reconstruction of microtubule–organelle interaction in whole primary mouse β cells. *J. Cell Biol.* **220**, (2021).

28. Buhmann, J. *et al.* Automatic Detection of Synaptic Partners in a Whole-Brain Drosophila EM Dataset. *bioRxiv* 2019.12.12.874172 (2019) doi:10.1101/2019.12.12.874172.

29. Perez, A. J. *et al.* A workflow for the automatic segmentation of organelles in electron microscopy image stacks. *Front. Neuroanat.* **8**, 126 (2014).

30. Guay, M. D. *et al.* Dense cellular segmentation for EM using 2D–3D neural network ensembles. *Sci. Reports 2021 111* **11**, 1–11 (2021).

31. Leonard, A. P. *et al.* Quantitative analysis of mitochondrial morphology and membrane potential in living cells using high-content imaging, machine learning, and morphological binning. *Biochim. Biophys. Acta - Mol. Cell Res.* **1853**, 348–360 (2015).

32. Nikolaisen, J. *et al.* Automated Quantification and Integrative Analysis of 2D and 3D Mitochondrial Shape and Network Properties. *PLoS One* **9**, e101365 (2014).

33. Talwar, A. *et al.* A Topological Nomenclature for 3D Shape Analysis in Connectomics.

34. Miyazono, Y. *et al.* Uncoupled mitochondria quickly shorten along their long axis to form indented spheroids, instead of rings, in a fission-independent manner OPEN. *Sci. REPORtS* | **8**, 350 (2018).

35. Bleck, C. K. E., Kim, Y., Willingham, T. B. & Glancy, B. Subcellular connectomic analyses of energy networks in striated muscle. *Nat. Commun.* **9**, (2018).

36. Abrisch, R. G., Gumbin, S. C., Wisniewski, B. T., Lackner, L. L. & Voeltz, G. K. Fission and fusion machineries converge at ER contact sites to regulate mitochondrial morphology. *J. Cell Biol.* **219**, (2020).

37. Tamada, H. *et al.* Three-dimensional analysis of somatic mitochondrial dynamics in fission-deficient injured motor neurons using FIB/SEM. *J. Comp. Neurol.* **525**, 2535–2548 (2017).

38. Conrad, R. & Narayan, K. CEM500K, a large-scale heterogeneous unlabeled cellular electron microscopy image dataset for deep learning. *Elife* **10**, (2021).

39. Xu, C. S. *et al.* An open-access volume electron microscopy atlas of whole cells and tissues. *Nat. 2021 5997883* **599**, 147–151 (2021).

40. Lucchi, A., Li, Y. & Fua, P. Learning for structured prediction using approximate

subgradient descent with working sets. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 1987–1994 (2013) doi:10.1109/CVPR.2013.259.

41. Riddle, D. L. *et al. C. elegans II. Cold Spring Harbor Monograph Series, Vol. 33* (Cold Spring Harbor Laboratory Press, 1997).

42. Cheng, B. *et al.* Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. *arxiv1911.10194v2 [cs]* (2019).

43. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 9351 234–241 (Springer Verlag, 2015).

44. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 386–397 (2020).

45. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A. & Girdhar, R. Masked-attention Mask Transformer for Universal Image Segmentation. (2021) doi:10.48550/arxiv.2112.01527.

46. Kirillov, A., Wu, Y., He, K. & Girshick, R. PointRend: Image Segmentation As Rendering. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9799–9808 (2020).

47. Kuhn, H. W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955).

48. Conrad, R., Lee, H. & Narayan, K. Enforcing Prediction Consistency Across Orthogonal Planes Significantly Improves Segmentation of FIB-SEM Image Volumes by 2D Neural Networks. *Microsc. Microanal.* 1–4 (2020) doi:10.1017/s143192762002053x.

49. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**, 100–106 (2021).

50. Vincent, L., Vincent, L. & Soille, P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 583–598 (1991).

51. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* vols 2016-Decem 770–778 (IEEE Computer Society, 2016).

52. Caron, M. *et al.* Unsupervised Learning of Visual Features by Contrasting Cluster

Assignments. *Adv. Neural Inf. Process. Syst.* **2020-December**, (2020).

53. Xiao, C. *et al.* Automatic mitochondria segmentation for EM data using a 3D supervised convolutional network. *Front. Neuroanat.* **12**, 92 (2018).

54. Bleck, C. K. E., Kim, Y., Willingham, T. B. & Glancy, B. Subcellular connectomic analyses of energy networks in striated muscle. *Nat. Commun.* **9**, 1–11 (2018).

55. Sarkans, U. *et al.* REMBI: Recommended Metadata for Biological Images—enabling reuse of microscopy data in biology. *Nat. Methods* 1–5 (2021) doi:10.1038/s41592-021-01166-8.

56. Franco-Barranco, D., Muñoz-Barrutia, A. & Arganda-Carreras, I. Stable deep neural network architectures for mitochondria segmentation on electron microscopy volumes. *arXiv:2104.03577 [eess.IV]* (2021).

57. Narayan, K. & Subramaniam, S. Focused ion beams in biology. *Nat. Methods* **12**, 1021 (2015).

58. Conrad, R. & Narayan, K. Cem500k, a large-scale heterogeneous unlabeled cellular electron microscopy image dataset for deep learning. *Elife* **10**, (2021).

59. Sarkans, U. *et al.* REMBI: Recommended Metadata for Biological Images—enabling reuse of microscopy data in biology. *Nat. Methods 2021 1812* **18**, 1418–1422 (2021).

60. Kikinis, R., Pieper, S. D. & Vosburgh, K. G. 3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support. in *Intraoperative Imaging and Image-Guided Therapy* 277–289 (Springer New York, 2014). doi:10.1007/978-1-4614-7657-3_19.

61. Kasthuri, N. *et al.* Saturated Reconstruction of a Volume of Neocortex. *Cell* **162**, 648–661 (2015).

62. Žerovnik Mekuč, M. *et al.* Automatic segmentation of mitochondria and endolysosomes in volumetric electron microscopy data. *Comput. Biol. Med.* **119**, 103693 (2020).

63. Lee, T. C., Kashyap, R. L. & Chu, C. N. Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms. *CVGIP Graph. Model. Image Process.* **56**, 462–478 (1994).

64. Caron, M. *et al.* Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *arXiv* (2020).

65. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition.

Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. **2016-December**, 770–778 (2015).

66. Kirillov, A., Wu, Y., He, K. & Girshick, R. PointRend: Image Segmentation as Rendering. (2019).
67. Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay. *arxiv1803.09820 [cs]* (2018).
68. Loshchilov, I. & Hutter, F. Decoupled Weight Decay Regularization. *7th Int. Conf. Learn. Represent. ICLR 2019* (2017).
69. Ghiasi, G. *et al. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation*. https://cocodataset.org/.

Figure 1

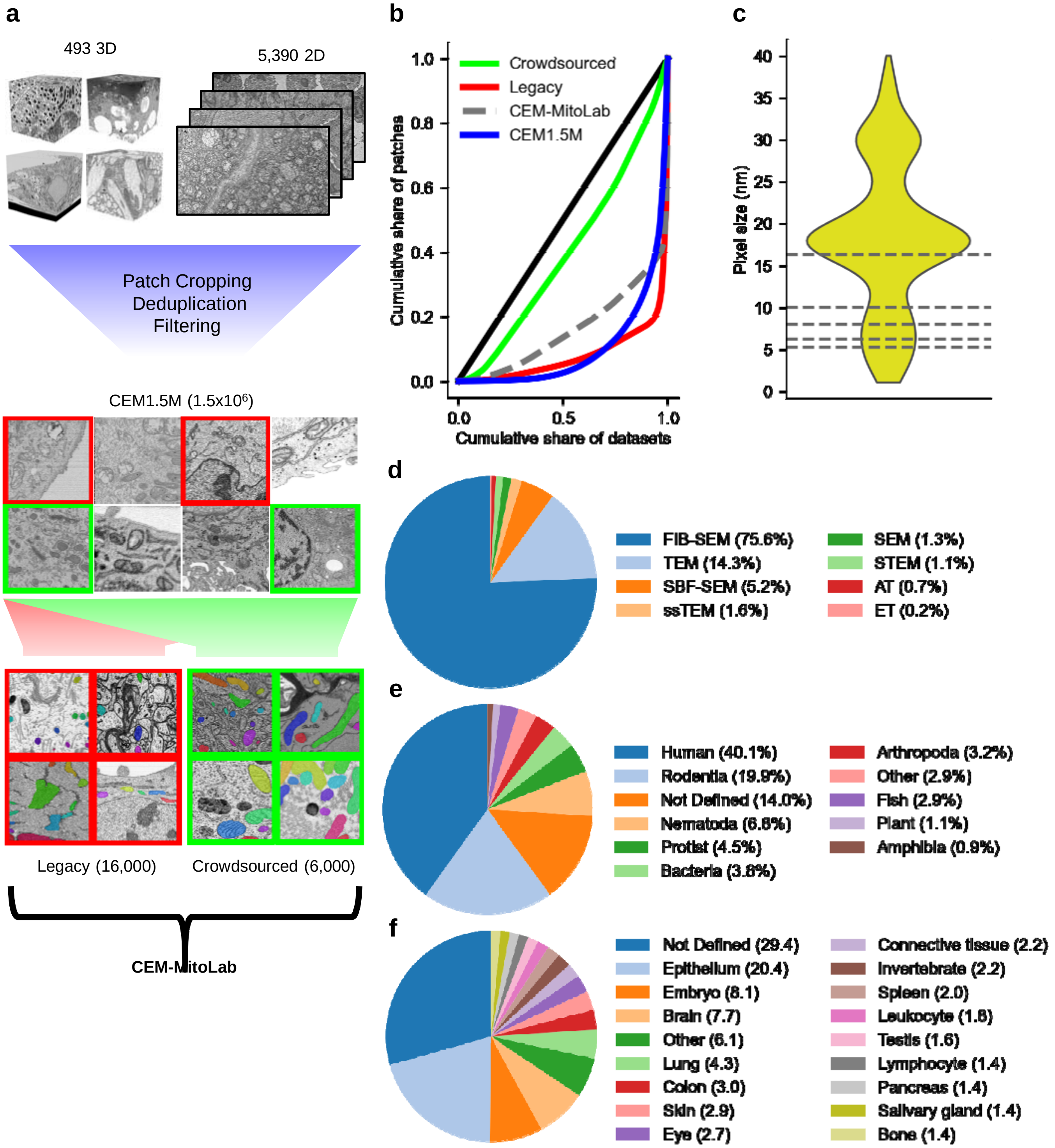

a
493 3D
5,390 2D
Patch Cropping
Deduplication
Filtering
CEM1.5M (1.5x10⁶)
Legacy (16,000)
Crowdsourced (6,000)
CEM-MitoLab
b
Crowdsourced
Legacy
CEM-MitoLab
CEM1.5M
Cumulative share of patches
Cumulative share of datasets
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
c
Pixel size (nm)
0
5
10
15
20
25
30
35
40
d
FIB-SEM (75.6%)
TEM (14.3%)
SBF-SEM (5.2%)
ssTEM (1.6%)
SEM (1.3%)
STEM (1.1%)
AT (0.7%)
ET (0.2%)
e
Human (40.1%)
Rodentia (19.9%)
Not Defined (14.0%)
Nematoda (6.8%)
Protist (4.5%)
Bacteria (3.8%)
Arthropoda (3.2%)
Other (2.9%)
Fish (2.9%)
Plant (1.1%)
Amphibia (0.9%)
f
Not Defined (29.4)
Epithelium (20.4)
Embryo (8.1)
Brain (7.7)
Other (6.1)
Lung (4.3)
Colon (3.0)
Skin (2.9)
Eye (2.7)
Connective tissue (2.2)
Invertebrate (2.2)
Spleen (2.0)
Leukocyte (1.8)
Testis (1.6)
Lymphocyte (1.4)
Pancreas (1.4)
Salivary gland (1.4)
Bone (1.4)

Figure 2

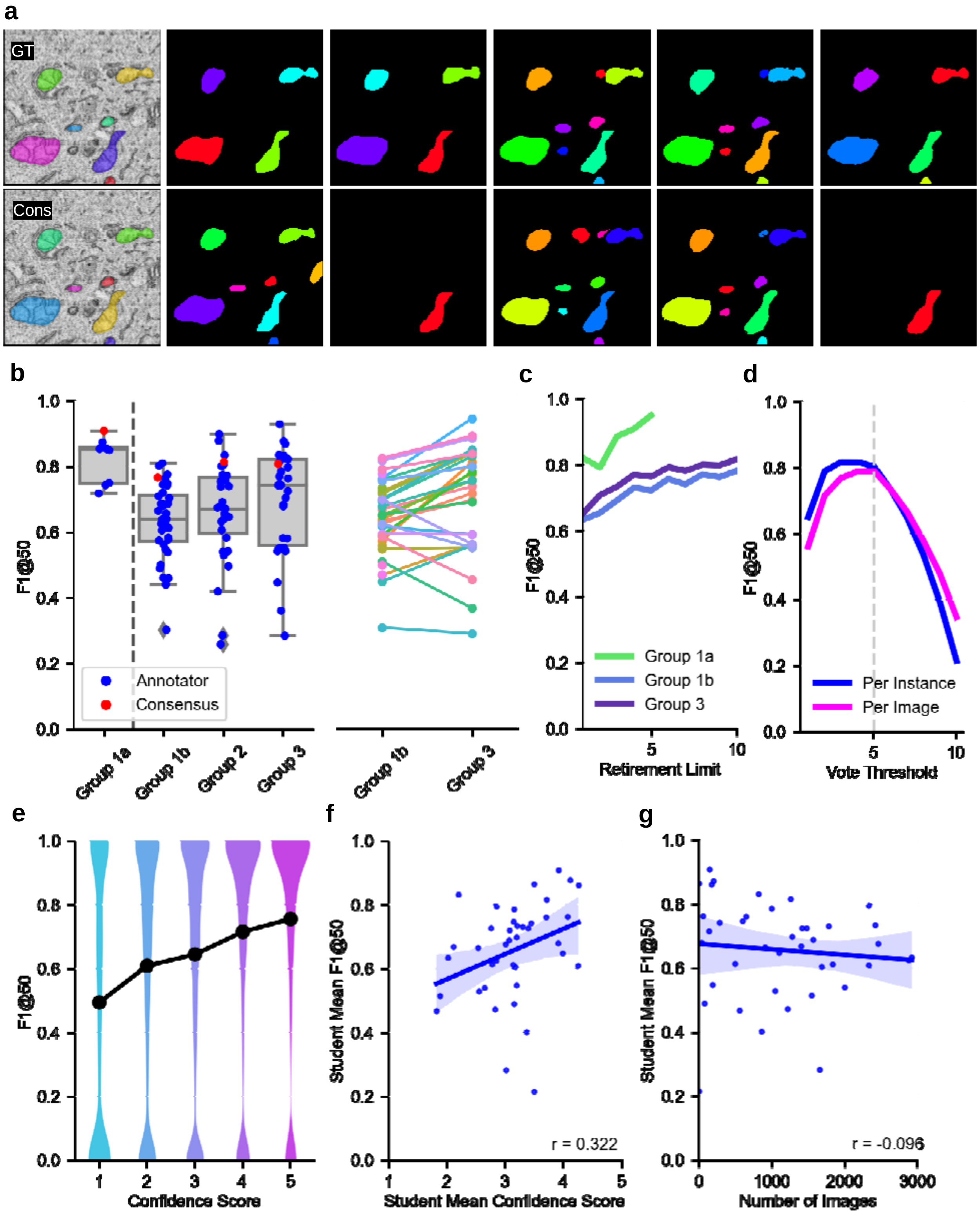

a
GT
Cons
b
1.0
0.8
0.6
0.4
0.2
0.0
F1@50
Annotator
Consensus
Group 1a
Group 1b
Group 2
Group 3
Group 1b
Group 3
c
Group 1a
Group 1b
Group 3
Retirement Limit
5
10
d
Per Instance
Per Image
Vote Threshold
e
Confidence Score
1
2
3
4
5
f
Student Mean F1@50
Student Mean Confidence Score
r = 0.322
g
Number of Images
0
1000
2000
3000
r = -0.096

Figure 3

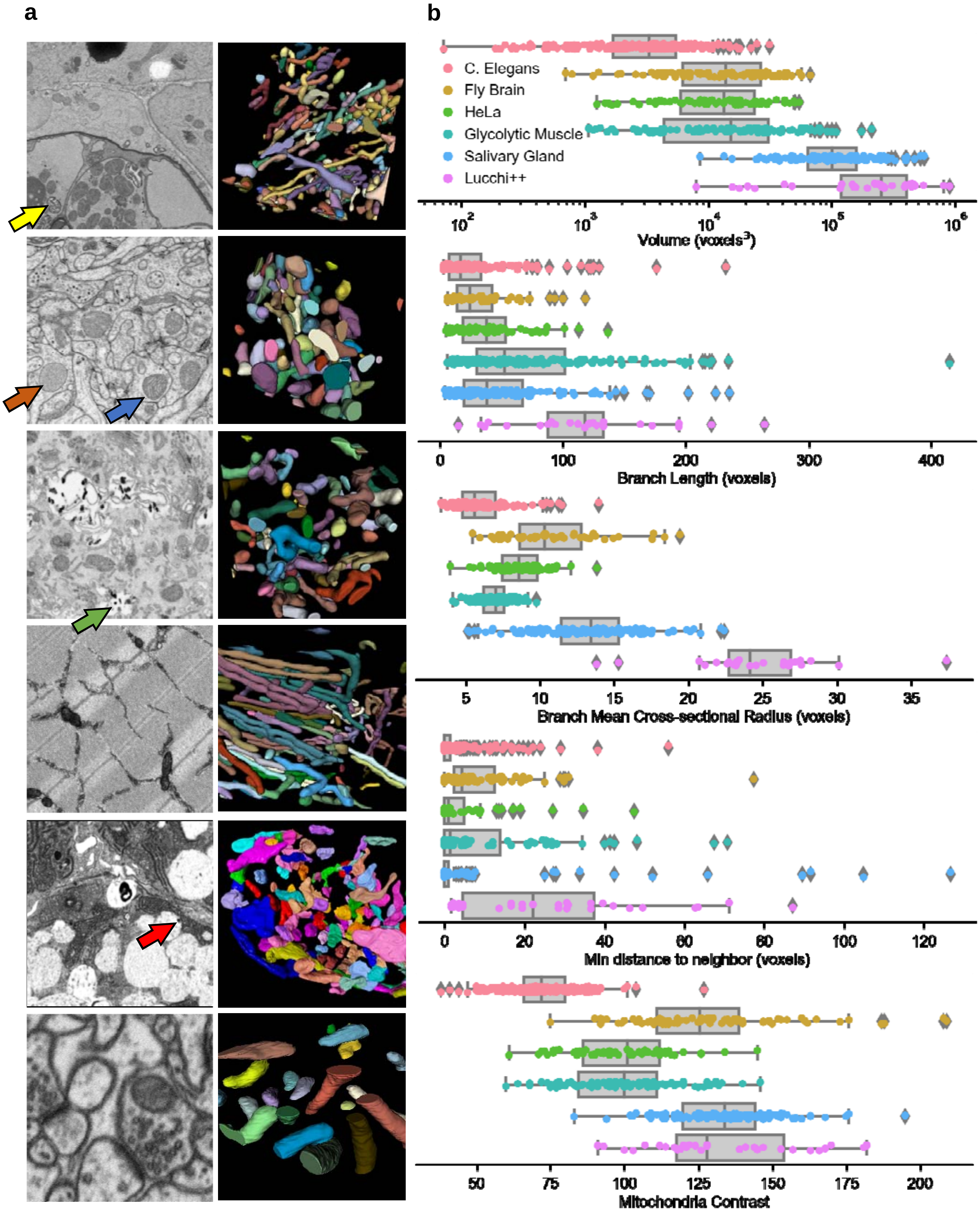

a
b
C. Elegans
Fly Brain
HeLa
Glycolytic Muscle
Salivary Gland
Lucchi++
10^2
10^3
10^4
10^5
10^6
Volume (voxels^3)
0
100
200
300
400
Branch Length (voxels)
5
10
15
20
25
30
35
Branch Mean Cross-sectional Radius (voxels)
0
20
40
60
80
100
120
Min distance to neighbor (voxels)
50
75
100
125
150
175
200
Mitochondria Contrast

Figure 4

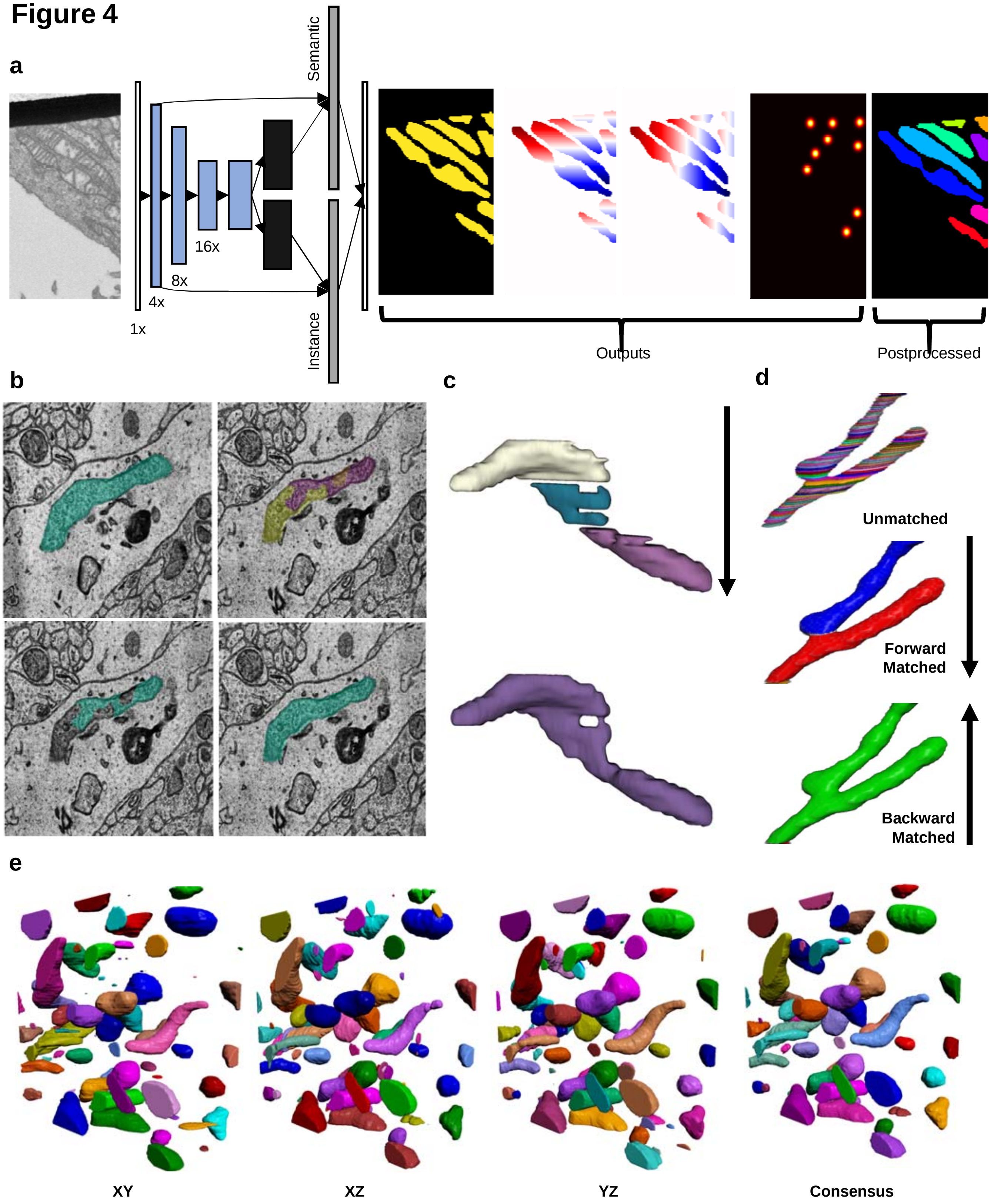
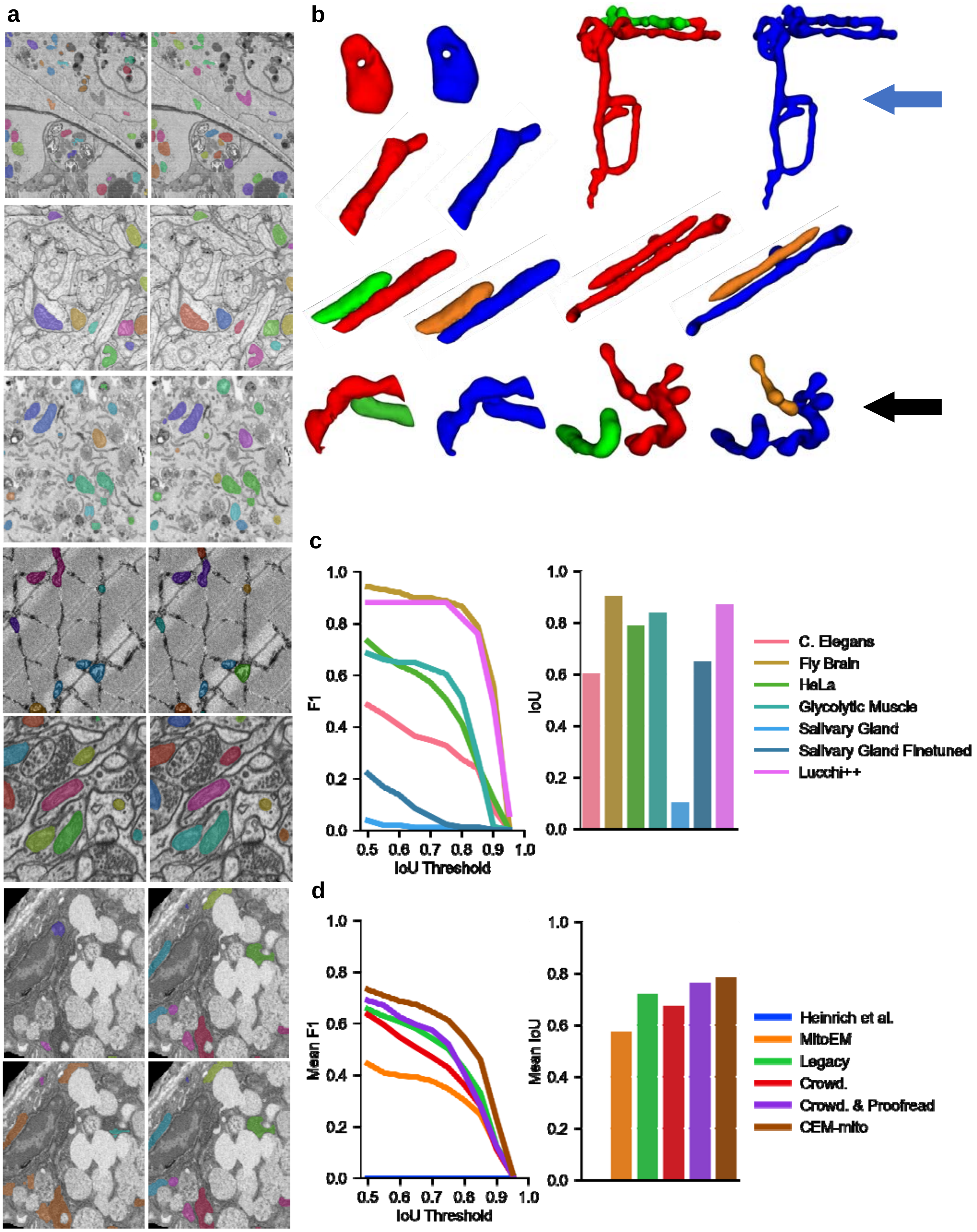

a
Semantic
1x
4x
8x
16x
Instance
Outputs
Postprocessed
b
c
d
Unmatched
Forward
Matched
Backward
Matched
e
XY
XZ
YZ
Consensus

Figure 5


a
b
c
1.0
0.8
0.6
0.4
0.2
0.0
F1
0.5
0.6
0.7
0.8
0.9
1.0
IoU Threshold
IoU
C. Elegans
Fly Brain
HeLa
Glycolytic Muscle
Salivary Gland
Salivary Gland Finetuned
Lucchi++
d
Mean F1
Mean IoU
Heinrich et al.
MitoEM
Legacy
Crowd.
Crowd. & Proofread
CEM-mito