

METHODS ARTICLE OPEN ACCESS

DeepSCEM: A User-Friendly Solution for Deep Learning-Based Image Segmentation in Cellular Electron Microscopy

Cyril Meyer¹  | Victor Hanss²  | Etienne Baudrier³  | Benoît Naegel³  | Patrick Schultz² 
¹IRIMAS, Université de Haute-Alsace, UR 7499, Mulhouse, France | ²IGBMC, Université de Strasbourg, CNRS UMR 7104, Inserm UMR-S 1258, Illkirch, France | ³ICube, Université de Strasbourg, CNRS UMR 7357, Illkirch, France

Correspondence: Cyril Meyer (cyril.meyer@uha.fr)

Received: 4 December 2024 | Revised: 15 July 2025 | Accepted: 12 August 2025

Keywords: cellular imaging | deep learning | electron microscopy | organelles | segmentation | software

ABSTRACT

Deep learning methods using convolutional neural networks are very effective for automatic image segmentation tasks with no exception for cellular electron micrographs. However, the lack of dedicated easy-to-use tools largely reduces the widespread use of these techniques. Here we present DeepSCEM, a straightforward tool for fast and efficient segmentation of cellular electron microscopy images using deep learning with a special focus on efficient and user-friendly generation and training of models for organelle segmentation.

1 | Introduction

The internal organization of the cell into membrane bound compartments defines functional domains named organelles whose number, distribution, shape, and mode of interaction reflect the cellular identity and its pathophysiologic state. The morphological description of organelles benefits from the high spatial resolution of electron microscopes that oversteps Abbe's diffraction limit of standard light microscopes. Early transmission electron microscopy (TEM) observations of cellular structures were essentially two-dimensional (2D) projections of ~100 nm thin resin-embedded cellular slices that lacked volumetric information. Stereological methods were first used to extrapolate 2D measurements to the whole cellular volume (Weibel et al. 1966), but these predictions were limited by the complex shape and uneven distribution of organelles. To reach 3D information on whole cells or tissues, serial sectioning protocols were developed to image successive physical slices of the sample, but this approach is technically very demanding and suffers from section distortions (Williams and Kallman 1955). In electron tomography

the specimen is discretely tilted over a range of angles in the microscope to record multiple views that can be back projected into a high-resolution tomographic reconstruction. This method is limited to 500 nm thick sections, but may be combined with serial sectioning to reach larger volumes.

Advances in scanning electron microscopy (SEM) enabled rapid imaging of resin-embedded thick specimens at nanometric scale. Three-dimensional information was first achieved by sequentially removing the top slice with an ultramicrotome fitted in a SEM chamber and repeatedly imaging the new block face (serial block face, SFB-SEM) (Leighton 1981). The top slices can be removed in finer increments using focused gallium ion beam (FIB) milling thus leading to FIB-SEM tomography (Heymann et al. 2006). FIB-SEM imaging enables recording high-resolution $5 \times 5 \times 5$ nm isotropic images on $40 \times 40 \mu\text{m}$ large image areas and over a depth of several tenths of micrometers (Narayan et al. 2014; Wei et al. 2012) and has become a powerful tool to obtain undistorted volume information (Bosch et al. 2016; Cretoiu et al. 2015).

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2025 The Author(s). *Biology of the Cell* published by Wiley-VCH GmbH on behalf of Société Française des Microscopies and Société de Biologie Cellulaire de France.

The morphological parameters and intracellular distribution of organelles can be studied quantitatively on large portions of cells or on whole tissue. However, image interpretation requires a segmentation step to annotate the cellular compartments and organelles in order to extract quantitative information such as size, distribution, interactions and morphology from these delimited regions. These quantitative morphological values can be used to compare the cell state in normal or diseased states and, in case of known mutations, may provide molecular clues to the observed phenotypic variations. A human expert can identify and segment these cellular domains interactively, but this task is very tedious and may take several days for each cell. In a context of massive datasets, the size and complexity of these image stacks prevents manual segmentation, therefore biologists need computer-aided methods to identify and segment cellular organelles.

In the field of image analysis, the recent rise of deep learning approaches led to the development of models based on the widespread U-Net neuron network architecture showing very good performance in supervised segmentation tasks (Ronneberger et al. 2015). These methods are now the standard to solve complex analyses of medical and biological images, in particular in the field of cellular electron microscopy (Heinrich et al. 2018; Ronneberger et al. 2015; Xiao et al. 2018). Deep learning-based segmentation of cellular organelles for EM images has been applied to mitochondria, endoplasmic reticulum and other organelles on images with possible anisotropy (Meyer et al. 2021, 2023). Recently, results were shared on the OpenOrganelle web repository on FIB-SEM images at 4 nm spatial resolution for over 35 organelles and macromolecular structures (Heinrich et al. 2021).

However, the use of supervised methods based on deep learning in the field of FIB-SEM imaging faces high image variability arising from the large diversity of cell lines, the various preparation protocols including different fixatives or cryo-preservation conditions and a large panel of contrasting agents and resins. Variations in the appearance, contrast, and texture of organelles can further arise from FIB-SEM image acquisition conditions. Altogether, variations in organelle aspect affect the efficiency of their automated detection and analysis. Our tests showed that the currently available U-net architectures are very powerful to predict organelles when the training and the prediction sets are imaged in the same laboratory on samples prepared from a single protocol, while they generally fail when trained on a given FIB-SEM dataset and applied on images from another laboratory or protocols. These observations led us to conclude that semi-automated segmentation is performed most efficiently when training and prediction are performed on a similar dataset which reduces image variability.

The implementation of a semi-automated segmentation solution through the training of a dedicated U-Net architecture applied to FIB-SEM images is not accessible to every user as it often requires coding skills. Some U-Net based segmentation solutions for FIB-SEM images exists and are provided within python jupyter notebooks or inside python-based anaconda environment. Both of them requires the user to have basic knowledge and ease of use in python coding and python environment (Hirabayashi et al. 2024; Gallusser et al. 2023). Here, we developed an easy

to install, ready to use interactive software that can use any volumetric FIB-SEM dataset to train a dedicated model. Existing tools, such as ilastik (Berg et al. 2019) or DeepMIB (Belevich and Jokitalo 2021) already provide a ready-to-install segmentation solution encapsulated in a Graphical User Interface (GUI), but our program differs in several aspects. Firstly, DeepSCEM was designed to be very user-friendly especially for the evaluation of its performances by a non-expert user. It can be operated through a few clicks thanks to dedicated distributions of the software binaries for both Windows and Linux environments. This means that DeepSCEM can be evaluated for a given problem in just a few hours by a non-computer-expert. The second property of DeepSCEM is its modularity. Being written in Python and using the widely-used TensorFlow library, new models and parameters can be easily added, increasing the possibilities for more advanced users while providing a basic tool with pre-set conditions that provide satisfactory results. For example, any TensorFlow model which has valid input and output shapes can be loaded, trained and used for prediction in DeepSCEM. The third property of the software lies in its performance. As our software is dedicated to the particular task of organelle segmentation, it has been optimized to perform this task. This tool accepts as input a training dataset composed of a representative set of 2D slices (stack) of a volumetric dataset along with the interactively segmented organelles or regions of interest. DeepSCEM will train a model to automatically segment the complete input volumetric datasets based on the provided manual segmented region. An evaluation set composed of a few manually segmented sections can be used to evaluate the accuracy of the predictions qualitatively and quantitatively. Thanks to its flexible software architecture, DeepSCEM can be used either through a graphical user interface (GUI) or a command line interface (CLI). The CLI mode is particularly useful for integrating the software into automated data processing pipelines.

2 | Results

In this section, a description of the DeepSCEM application and its usage is provided. We will first present the general workflow followed by examples of how the program is used, illustrated on a representative FIB-SEM image stack. More information about the imaging conditions and the tools used for initial image segmentation are available in the methods section.

2.1 | Workflow

In this subsection, we describe the general usage of DeepSCEM, for automated segmentation of cellular electron microscopy images. The process can be divided into five main steps (Figure 1).

- i. A training image dataset is selected from the original image stack and the corresponding annotations need to be created. Annotations correspond to a manual segmentation of the organelles in which image features that allow the recognition of a particular cellular ultrastructure are associated with the corresponding organelle class or instance. When analyzing a large 3D stack of several hundred images, the training set typically represents a few tens of images that

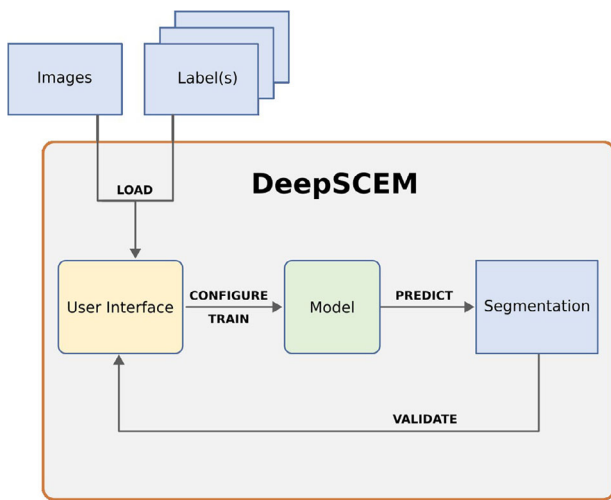


Figure 1 | General workflow of the DeepSCEM application. Images along with one or multiple labels (binary mask files) are loaded into DeepSCEM through its user interface. Each combination of image and labels composes a sample. Existing compatible model can be directly trained. New models can be created and configured by the user for training. The model is used to generate a segmentation prediction that can be evaluated or validated through the user interface.

should be representative of the different types of structures of interest, and should be large enough to train a robust neural network model. In order for the deep learning process to make a good prediction, the object of interest must be present several times on several images. The accuracy of the prediction increases with the number of annotations, and poor or unsatisfactory predictions can be improved by increasing the number of the annotated instances. Annotations can be provided using software products like MIB (Belevich et al. 2016), ilastik (Berg et al. 2019) or any other tool that allows manual segmentation of the images and generates binary images of the region of interest in TIFF format.

- ii. A neural network model for segmentation is then created. DeepSCEM provides a model generator which, in the current version, can create 2D and 3D U-Net based models with various parameters including number of filters per layer, model depth, and use of residual connections. By default, the program provides a standard neural network model, but users can also create their own custom models using TensorFlow and load them directly into DeepSCEM.
- iii. The model is then trained on the annotated dataset to recognize the distinctive features of the object of interest. Briefly, small patches of images and their corresponding masks are extracted from the segmented area to be used as training examples for deep learning. The application allows the user to select the patch size, batch size, loss function, and the number of steps to train the model as described in the example section.
- iv. The performance of the training model can be assessed quantitatively by using a separate set of manually segmented images. This validation dataset consists of a few images extracted from the 3D image stack. The aim of this step

is to compare the segmentation performed by an expert, also called ground truth, with the predicted segmentation. Quantitative values using F1-score and intersection-over-union metrics can be used to prevent overfitting and to select the best-performing training model.

- v. The trained model can now be used to segment cellular images coming either from the remaining images of the same stack or from different image stacks.

DeepSCEM provides tools to visualize the results of the segmentation, allowing the user to make a quick qualitative evaluation.

2.2 | Examples

In this section, we present two case studies to illustrate the use of DeepSCEM to segment organelles in cellular electron microscopy images. All the data used for these examples and the exact results presented in this section are available for public research (Meyer 2024a, 2024b; Meyer and Schultz 2023). We present qualitative and quantitative results for different classes (organelles) to illustrate the accuracy of the implemented method.

2.2.1 | User Case 1: Single Mask

The aim of this first example is to train a model to automatically segment a single class of organelle, the mitochondria, in a 3D stack of 600 consecutive FIB/SEM images of a HeLa cell (see Material and Methods). The training set consists of 40 images of the 3D stack that may be consecutive or any selected series. Mitochondria were manually segmented in this training set by contouring the external mitochondrial membrane using the MIB program and filling its inner part (Belevich et al. 2016). The result of this segmentation is saved as the annotation set corresponding to a binary mask in TIFF format for each image where the inside of the mitochondria is marked with 1 and the outside with 0. This mask is used by DeepSCEM to extract image patches 256 by 256 pixels in size that are representative of the Region Of Interest (ROI) to be used for the training process. Alternatively, the mask may delineate only the contour of the ROI, or characteristic membrane-less textures. To start the training process, the training set and its corresponding annotation set are first loaded using the DeepSCEM interface (Figure 2). In order to improve the prediction score, the number of slices annotated for training may be adjusted according to the complexity of the object of interest, the noise level, or the abundance of the feature of interest.

The next step is to generate a starting model. For this experiment, we used the U-Net provided by default within DeepSCEM user interface which is a 2D U-Net with a depth of five down sampling/up sampling blocks, 32 filters in the first block that are doubled after each down sampling and residual connections for each block (He et al. 2016a). We use the Dice loss function (Milletari et al. 2016) for training during 64 epochs of 256 batches with a batch size of 8 (Figure 3). In most cases these default values will produce a useful trained model, but these default values such as batch size, loss function or number of training steps can be modified to optimize the training model.

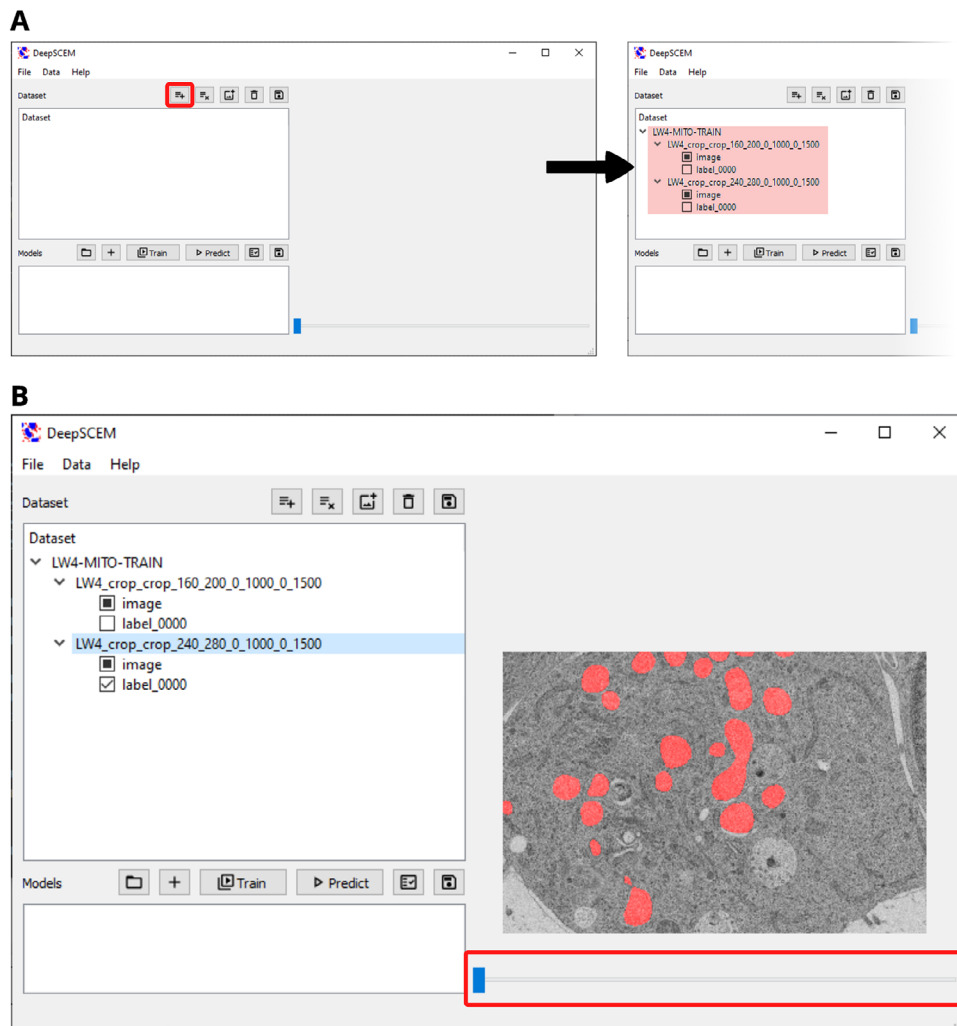


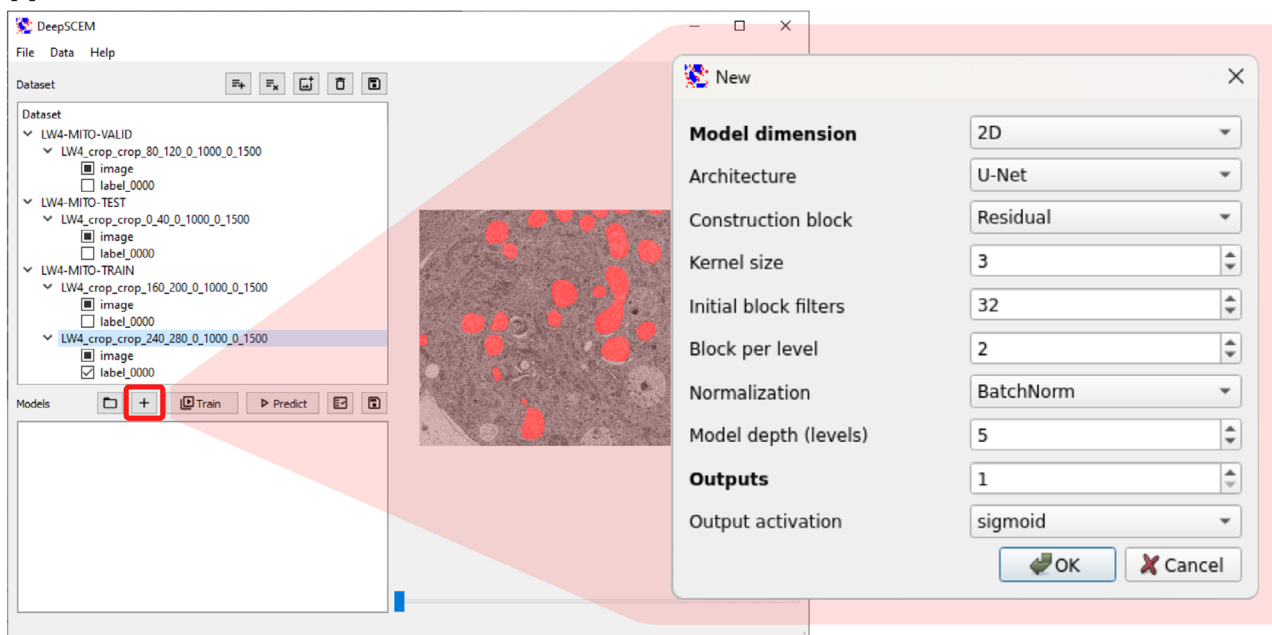
Figure 2 | Loading and browsing datasets within the DeepSCEM graphical user interface (GUI). (A) hdf5 based dataset can be uploaded using the icon highlighted in red and is visible in the “dataset” section of the GUI (Red arrow). Here, a complete dataset is highlighted in red. Datasets in DeepSCEM are structured as follows, the dataset name is first displayed. Each dataset can contain multiple samples representing different regions of interest (ROI) of a volumetric image stack. Here, the dataset entitled “LW4-MITO-TRAIN” has two samples named “LW4_crop_crop_160_200_0_1000_0_1500” and “LW4_crop_crop_240_280_0_1000_0_1500”, respectively. Each sample is composed of at least a stack of EM volumetric images. Presently, the sample “LW4_crop_crop_160_200_0_1000_0_1500” is composed of a stack of EM images named “image” and a set of binary masks of segmented mitochondria named “label_0000”. Datasets can be visualized and browsed by moving the blue thumb of the scrolling bar framed in black at the bottom of the screen. (B) The sample “LW4_crop_crop_240_280_0_1000_0_1500” labels are shown: the “image” label shows an EM micrograph and the “image + label_0000” shows the superimposition of the “image” and “label_0000” labels highlighting the segmentation of mitochondria. Images have been annotated, M stands for mitochondria, E, endosomes, ER, endoplasmic reticulum.

Then the starting model can be trained on the annotated dataset (Figure 4). The program also provides opportunities for data augmentation; modifications made to the original data during the training phase. In our case, no data augmentation was used except for image rotations and flipping. Within the annotated “mitochondria” region, representative and non-overlapping image patches that describe the texture and intrinsic features of this class of organelles are extracted and used in the learning process. The patch size needs to be adapted to the dimensions of object of interest and the image magnification. The patch should be large enough to include as many characteristic features of the object as possible as well as some information about the context of the organelle, but not too large to avoid including surrounding noise that has no predictive power.

To evaluate the performance of the trained model, a validation dataset has to be prepared either by using the crop function available in the software GUI or by splitting the validation and the training sets by employing a generic software such as ImageJ. In our case we selected and annotated the next 20 slices of the image stack to assess whether the model correctly predicts the presence of mitochondria. The validation set must contain the feature of interest and can be a selection of non-consecutive image patches representing the mitochondria in different conditions. Annotations must be provided by the experts as segmented mitochondria, and are loaded into DeepSCEM along with the validation set and the trained model (Figure 5).

As a measure of predictive performance, we use the F1-score which is calculated at the pixel level from the precision and

A



B

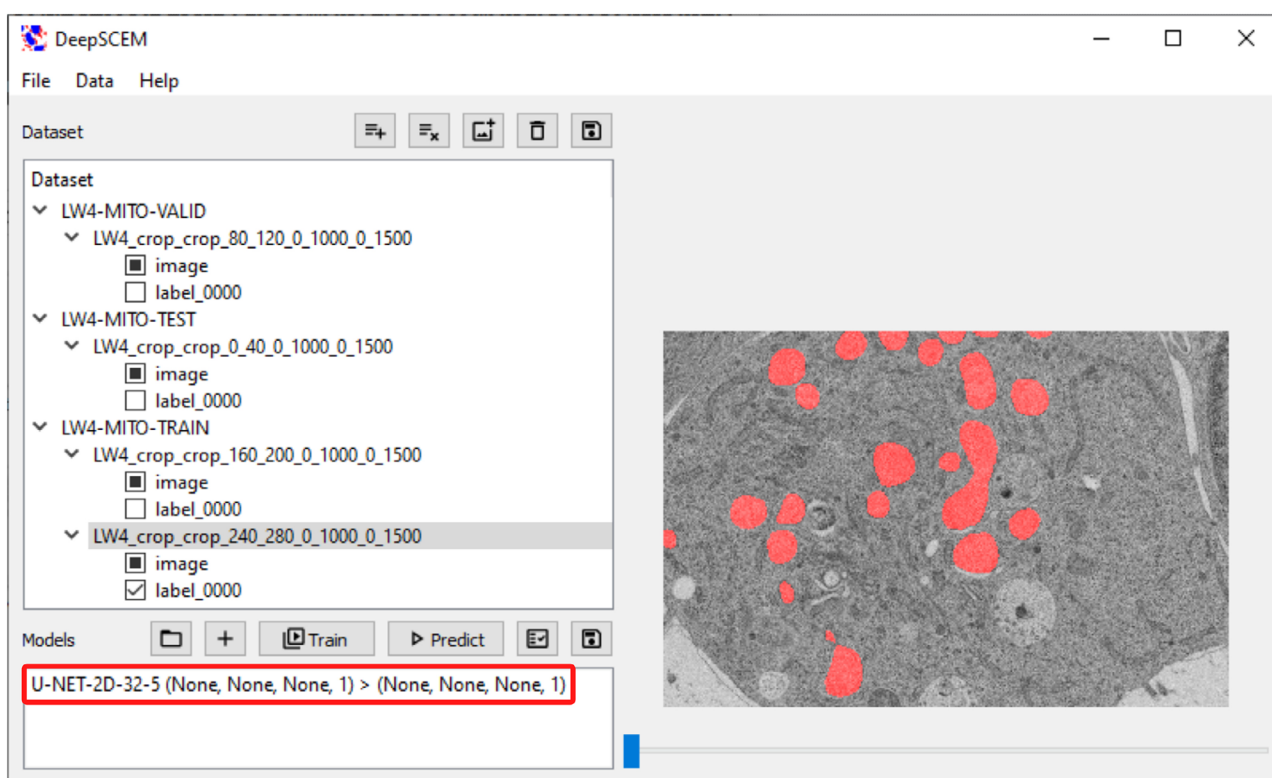


Figure 3 | Creation of a starting model. (A) Upon selection of the red framed “+” icon, a starting model can be set up and loaded into the DeepSCEM GUI. The zoomed frame shows the deep learning network parameters used to create a starting model. The values shown here are displayed by default as they gave good results when used for both user cases. (B) The newly created model is listed in the “Models” section of the DeepSCEM GUI (red frame). The model dimension, the number of initial block filters as well as the model depth are reported in the model name. As it stands, this initial model would give poor segmentation prediction because it has not been trained.

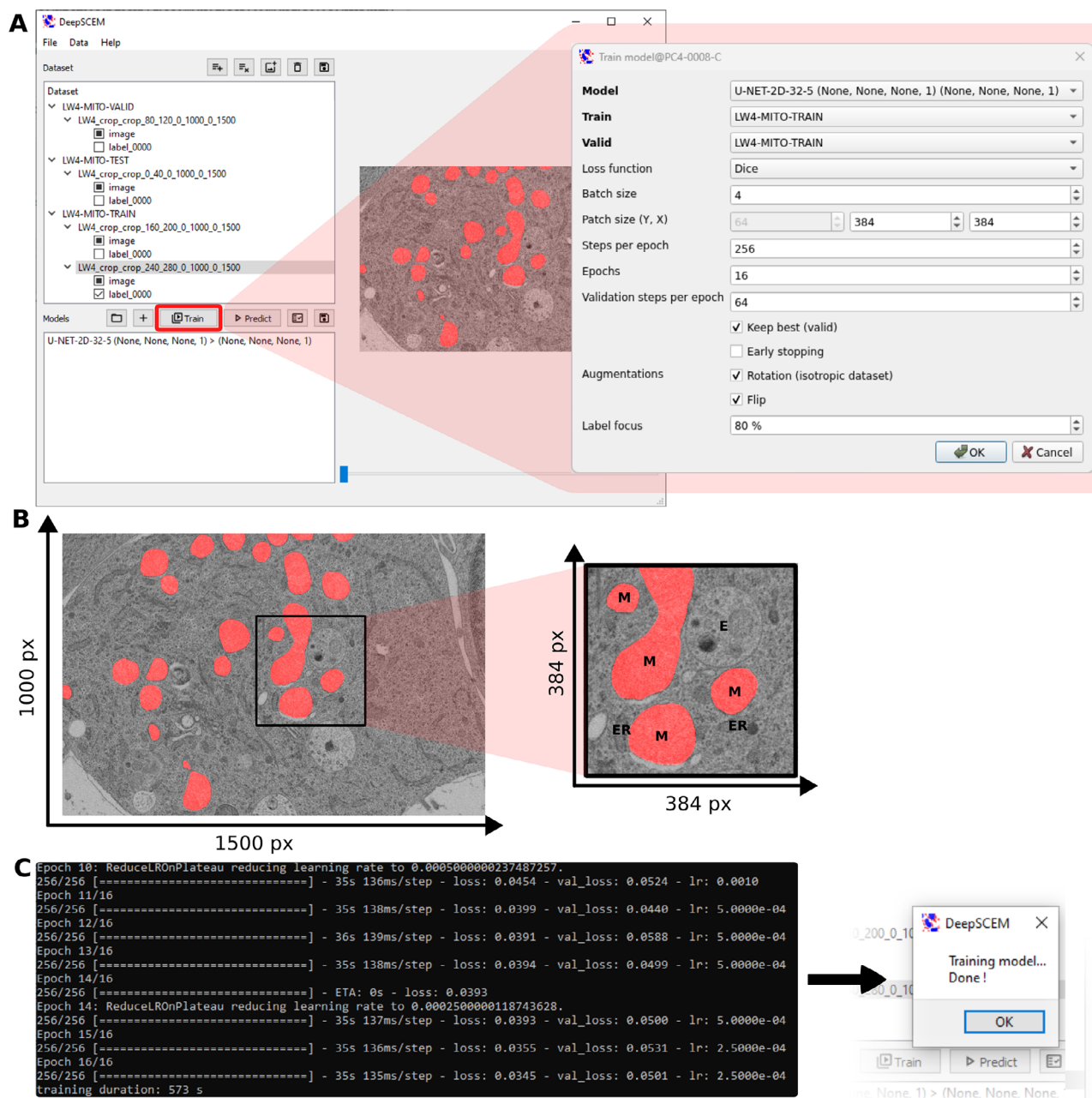


Figure 4 | Prediction model training. (A) Newly created models or compatible ones uploaded to the DeepSCEM GUI can be trained using the red framed “train” icon. This will open a new window (see the zoomed frame) giving access to the training configuration. There, the training “Model” with the “Train” and “Valid” datasets have to be selected accordingly. “Batch size” and “Patch size” have to be balanced depending on the image pixel size, the image features size (such as organelles) and the computer memory available. The number of “epochs” correspond to the number of times the complete dataset was used by DeepSCEM learning algorithm during training. The “steps per epochs” are the number of batches used to train on the complete dataset and the number of “validation per epoch” are the batches used to compute the validation (val_loss) at the end of each epoch. The “keep best” option saves the version of the model with the lowest val_loss. The “early stopping” option stops the training of a model that doesn’t improve based on the losses. The “rotation” and “flip” are data augmentation options that increase the variability of the training dataset. The “label focus” option biases the randomness of the patches extraction resulting in a training dataset composed of a desired proportion of segmented patches, here 80%. (B) Illustration of a patch size of $384 \times 384 \text{ pixel}^2$ (xy) extracted from a section of a 3D stack of $1500 \times 1000 \times 600 \text{ pixel}^3$ (xyz). The magnified image on the right panel shows a 2D patch of $384 \times 384 \text{ pixel}^2$ with a pixel size of 7.5 nm, sufficient to extract multiple instances of mitochondria (M), endosomes (E), or endoplasmic reticulum (ER). (C) left panel: Training losses as well as the learning rate (displayed as “lr”) are shown within the terminal as soon as the training starts and are updated after each completed epoch. The progression of the training can be appreciated by comparing the losses of older epochs to new ones. B right panel: The user is informed of the end of the training by a pop-up window on the main GUI.

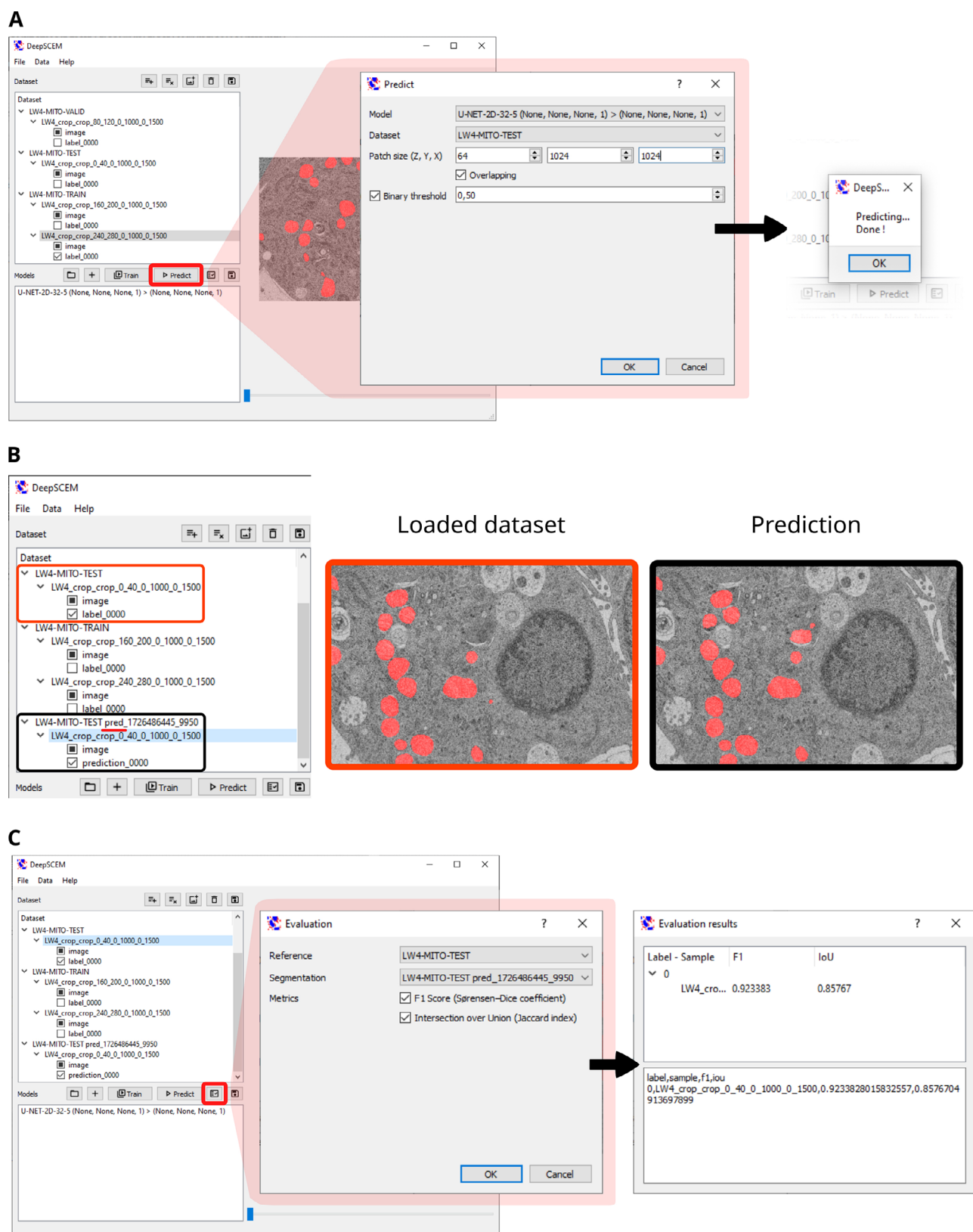


Figure 5 | Segmentation prediction and evaluation. (A) The red framed “predict” icon on the main GUI gives access to the prediction parameters window. “Model” and “Dataset” entries have to be selected accordingly and the patch size has to be set up depending on the size of the stack to be segmented and the computer memory. The prediction progression can be followed in the terminal. The user is informed that the prediction ended by a pop-up on the main GUI. (B) The dataset loaded for segmentation and its prediction can be both visualized within DeepSCeM GUI upon selection. The title of the predicted segmentation sample contains an added “pred” mention (see the red stroke). (C) The button with the mark symbol framed in red gives access to the evaluation windows. “Reference” loaded dataset and “Segmentation” predicted datasets have to be selected accordingly as well as the metrics wanted. F1 and/or IoU scores are then calculated and displayed.

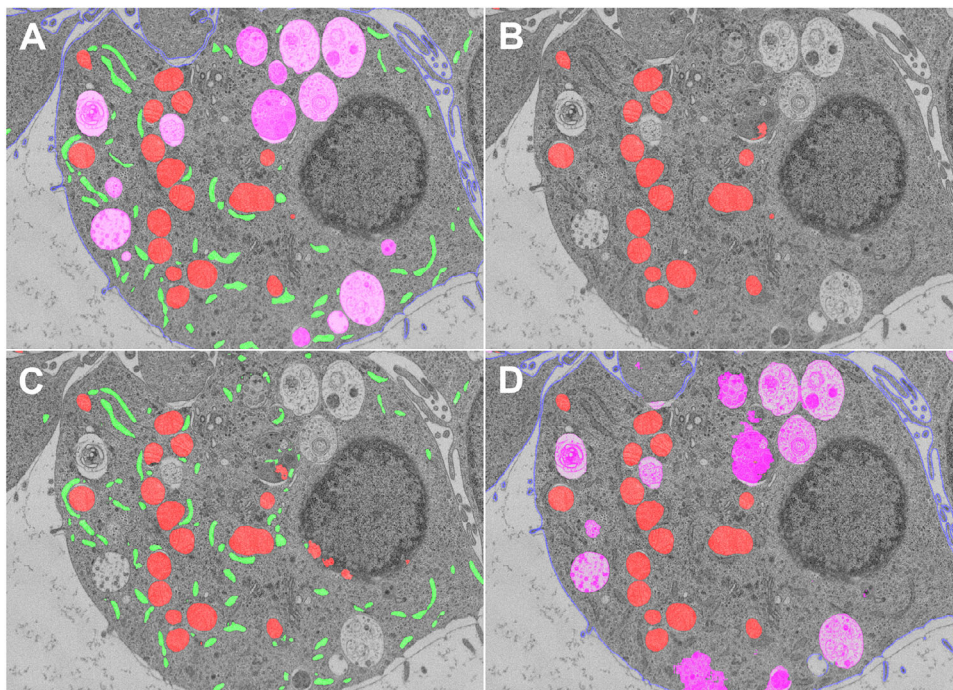


Figure 6 | Qualitative segmentation predictions. (A) Reference segmentation of mitochondria (red), endoplasmic reticulum (green), endosomes (pink), and plasma membrane (blue). (B) Mitochondria segmentation prediction using a binary model. (C) Mitochondria and endoplasmic reticulum segmentation predictions using a model trained on two classes. (D) Mitochondria, endosomes, and plasma membrane segmentation predictions using a model trained on three classes.

recall of the test, where a pixel has a negative value when the predicted class does not correspond to the segmented class, and a positive value when the prediction matches the annotation. The precision is the number of true positives divided by the number of predicted positives and the recall is the number of predicted positives divided by the number of all positives. The F1-score corresponds to the harmonic mean of precision and recall. The intersection-over-union (IOU) metrics (Everingham et al. 2010) can also be used to measure how well pixels predicted to be mitochondria align with the ground truth. Both metrics can be retrieved from the software GUI. In the case of mitochondria, the predictions are of high quality (F1 score of 0.950).

The last step is to apply the trained model to the remaining 560 images of the stack. One has to enter the prediction mode of DeepSCEM, select the trained model to be used and load the stack of images on which a segmentation prediction is to be performed. If such a prediction is made on images that have already been annotated, such as a validation set, a test score can be computed to compare the manual segmentation with the prediction. To do this, the evaluation mode has to be entered to select the images to be compared which then allows to compute the F1-score.

2.2.2 | User Case 2: Multiple Masks

The aim of this second example is to train a model to automatically segment multiple organelles in the same 3D stack. The training set consisted of 40 images where each organelle was segmented separately resulting in a different mask file for each organelle and for each annotated image (Figure 6A). Mitochondria were segmented as described above, and endoplasmic

reticulum (ER) as well as endo-lysosomes were annotated in the training set. In addition, we also annotated the external cell membrane (plasma membrane) which separates the cell boundary from the external matrix. In this case only the cell contour appeared in the mask as a continuous line whose width was three times the size of the membrane.

In order to train a deep neuronal network model for multiple organelles the number of labels has to be set accordingly. In the DeepSCEM environment, each label corresponds to binary masks of the segmented objects of interest. All mask files as well as the raw images are loaded separately, listed in the DeepSCEM “Dataset” section and organized as a sample (Figure 2). A new model is generated as for user case 1.

When the model is trained with two classes, mitochondria and ER, using default parameters and a patch size of 256×256 , the prediction score for mitochondria remains similarly high (F1-score of 0.951) while ER is less accurately predicted (F1-score of 0.709, Figure 6B,C). This reflects the lower detection limit of the organelle boundaries even for a trained expert. Indeed, two independent experts perform only slightly better and their compared annotation yields a F1-score of 0.751. Changing the patch size to 512×512 slightly degrades the F1 score for both organelles (0.947 for mitochondria and 0.691 for ER) possibly because including more contextual information adds more noise than signal. We also trained the model with three classes, namely, mitochondria, endosomes, and plasma membrane which yielded F1 values of 0.966, 0.836, and 0.740, respectively. The relatively poor score for endosomes may be related to their pleiotropic shapes and internal organization. Indeed, the endo-lysosomal compartment includes early and late endosomes that have a

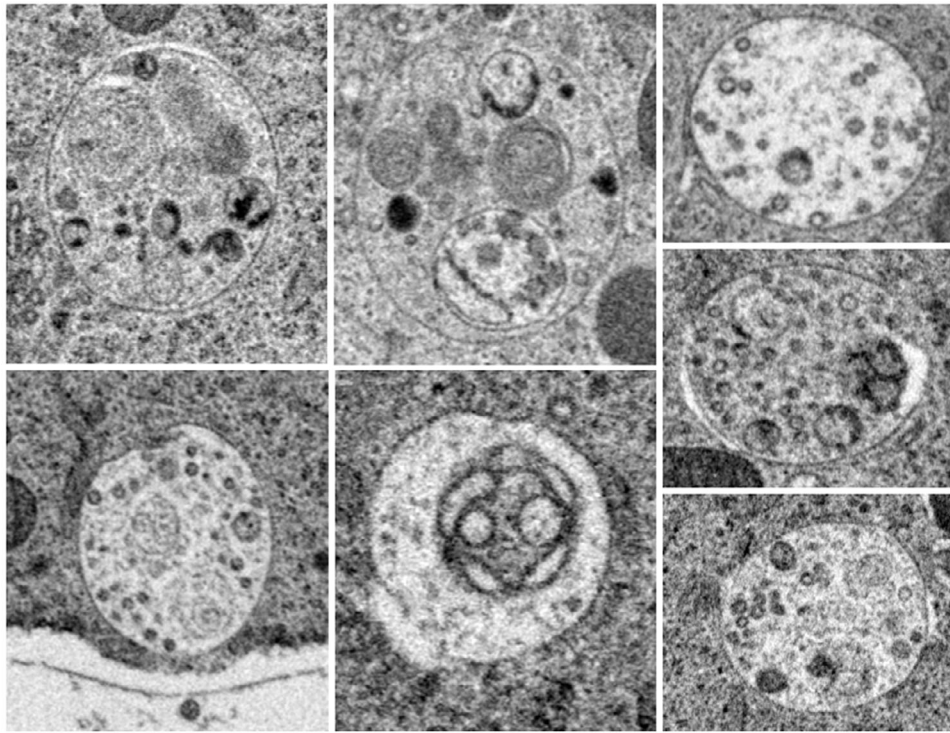


Figure 7 | Gallery of endosomes and lysosomes that form a very heterogeneous class of organelles that have a different appearance according to their origin and maturation stage.

different texture, contrast and general appearance (Figure 7). A larger training set representing the whole variability in shape and textures is likely to provide better predictions.

3 | Conclusion

Here we describe DeepSCEM, a new tool to train a deep learning model for automatic segmentation of volumetric electron microscopy images. This tool delivers predicted segmentations of cellular features that was illustrated by FIB/SEM images for HeLa cells but can be adapted to any kind of cellular or tissue system as well as for other volumetric images modalities such as block phase SEM images or serial sectioning. The trained model can be further used to predict segmentations for image stacks of similar cell types and preparation conditions. More expert transfer learning approaches will be developed in the future to overcome the prediction failure when the cells are very different or are prepared with different staining or fixation conditions. Considering the limited time required to train a model, it may be possible to overcome this limitation by providing a dedicated pre-trained model for each preparation condition.

DeepSCEM is intended to be evolutive and to be extended to comply with state-of-the-art deep learning architectures, computing of losses and additional options to optimize the deep learning models, but also to adapt to the diverse needs of future users. The automatization of hyperparameter tuning using nnU-Net, the implementation of sparse segmentation (Isensee et al. 2021) (https://github.com/MIC-DKFZ/nnUNet/blob/master/documentation/ignore_label.md) as well as the

availability of different loss functions such as cross entropy or mean square error will be considered in future versions of the program to help users to fine tune parameters and extend the operability of the segmentation tool. For this purpose, users are widely encouraged to contact the authors using the GitHub issue page in order to propose improvements to the software.

4 | Materials and Methods

4.1 | Sample Preparation

The sample consists of HeLa cells stably expressing the protein STARD3 (Wilhelm et al. 2017) and processed after high pressure freezing and freeze substitution. HeLa cells were grown on carbon-coated sapphire disks in DMEM containing 10 percent Foetal Calf Serum before being high pressure frozen (HPM 10 Abra Fluid AG) in liquid nitrogen. The samples are then freeze-substituted (AFS, Leica) and embedded in Lowicryl HM20 as previously described (Spiegelhalter et al. 2010). The samples were dehydrated at -90°C for 9 h in dry acetone containing 0.25 percent uranyl acetate and 0.1 percent glutaraldehyde. Temperature was raised to -45°C over 18 h with a slope of 2.5°C/h . The cells were then rinsed with acetone and infiltrated in 10 percent and 25 percent resin (Lowicryl HM20) for 2 h each bath. When the resin concentration reached 25 percent, the temperature was raised to -25°C and the resin concentration raised to 100 percent in three successive steps. The samples were then placed in three consecutive baths of pure resin lasting 8 h each before UV polymerization at -25°C for 48 h and warming to 20°C during 9 h.

4.2 | Cellular Electron Microscopy

Focused Ion Beam (FIB) milling and SEM observations were performed using the Auriga 60 instrument (Carl Zeiss Microscopy GmbH, Oberkochen). For 3D reconstruction, 15 nm thick slices of the resin embedded sample were removed by FIB milling and the freshly exposed cross-section was imaged in a serial manner with a lateral pixel size of 7.5 nm. The analyzed image stack is 2048 by 1536 pixels in size corresponding to 15.3 by 11.5 μm and consists of 600 slices corresponding to a depth of 9 μm . For FIB milling the probe current was set to 2 nA at 30 kV acceleration potential. For noise reduction line averaging with line averaging count number $N = 11$ and scan speed 4 was used. The resin block was glued on an aluminum stub using silver paint (Silver dag 1415, Plano GmbH, Wetzlar). First all side walls beside the block face were covered by silver paint and then the block face was sputter-coated with a few nanometer thick metal film to avoid resin charging. The SEM acceleration voltage was set to 1.5 kV, the SEM aperture was 60 μm and high current mode was turned on. For SEM imaging the Energy selective Back-scattered electron (EsB) detector was used with a retarding EsB grid voltage of 1155 V. The gray level scale was reversed in order to obtain a TEM like image contrast.

4.3 | Annotations

Manual segmentations or annotations have been generated using MIB (Belevich et al. 2016). Similar annotations can also be created using software tools such as ilastik (Berg et al. 2019) or any other tool which allows annotations of the images and generates binary images of the region of interest in TIFF format. Closed regions of interest such as mitochondria, nuclei, endosomes, endoplasmic reticulum are segmented to encompass all pixels that are part of the structure. External cell membranes or nuclear membranes are segmented as continuous lines with a suitable line thickness. For the validation set, each instance has been interactively segmented by two experts in order to estimate the inter-expert decisional variation.

4.4 | Deep Learning

4.4.1 | Hardware Configuration

The analysis shown here has been performed on a workstation equipped with 64 Go of RAM, a NVIDIA RTX 2080 Ti GPU, and an Intel Xeon W-2135 CPU. Using this configuration, the average training time was 59 min, ranging between 2 h for a large model containing 64 filters, instead of the standard 32, and less than 30 min for models containing 16 filters. The prediction time on a test set of 40 slices was 40s, but the prediction time for a stack of 600 images 2048 by 1536 pixels in size was 20 min. A total processing time of 1 h is therefore to be expected, not counting the annotation time. DeepSCEM can also run on a less powerful hardware configuration but, in order to load large datasets and train bigger models, the amount of memory should be proportional to the dataset size and model parameters. Although not required, the GPU card significantly reduces the computation time, especially for the training process.

4.4.2 | Architecture

In the biomedical field, the most common architectures include U-Net (Ronneberger et al. 2015), DeepLab and its evolutions, DeepLabV3+ (Chen, Papandreou, et al. 2018; Chen, Zhu, et al. 2018). The U-Net architecture has quickly become popular as the model is easy to implement, and extend. The original article presents a high-performing network with a small amount of training data, thanks to data augmentation techniques. Also, as the model can be implemented as a fully convolutional neural network, its input and output shapes can be adapted to user prerequisites.

The U-Net uses the principle of an encoder-decoder architecture with long skip connections between the convolutional blocks of the same level. Many variants of the U-Net model have been proposed that modify the processing blocks (He et al. 2016a, 2016b) or the types of connections (Chaurasia and Culurciello 2017). However, regardless of the variations proposed, the overall architecture remains the same: an encoder-decoder model with connections between the encoder and the decoder.

To give a simplified idea of how the elements of the model work together, it can be summarized as follows. The encoder extracts features from the input image. It consists of a series of convolutional blocks, each of which is composed of convolutional layers and various optional layers such as batch normalization (Ioffe and Szegedy 2015). The number of filters in the convolutional layers is increased as the image is down sampled, allowing the network to capture more complex features. The decoder is responsible for generating the segmentation mask from the features extracted by the encoder. It is in essence the symmetry of the encoder, where down sampling is replaced by up sampling to generate a segmentation mask with the same resolution as the input image. The long skip connections are concatenations between the convolutional blocks of the same level in the encoder and in the decoder. These connections allow the network to preserve the spatial information and fine details of the input image to generate a more accurate segmentation mask.

The U-Net architecture can be easily parametrized to fit specific tasks. You can for example change the number of filters per layer, the depth (number of down sampling/up sampling steps in the encoder/decoder) or the processing blocks composition by adding residual connections or batch normalization layers. Those changes can allow us to create a very coarse model for a specific easy task, but also to create a very large and efficient model for a more complex task. The architecture can be used with either 2D or 3D processing blocks.

4.5 | Training

Once we have a segmentation model, it is classically initialized with random weights and the model needs to be trained. In order to train a model, a dataset with images and their corresponding annotations are required in TIFF format. During the training phase, the model is presented with batches of images (patches) and their corresponding annotations. The set of patches are generated randomly in the image by avoiding overlapping and each patch may contain between 0 and all classes under

consideration. No sampling correction is applied to compensate for class imbalances. The loss function measures the difference between the predicted segmentation mask and the reference segmentation. The weights of the model are updated based on this difference using an optimization algorithm based on stochastic gradient descent, like Adam (Kingma and Ba 2015). To improve the performance of the model, various techniques can be used during training, like data augmentation, which involves applying transformations to the images and annotations. Overfitting is a common issue in deep learning that can be simply explained as the model starting to memorize the training data instead of learning the underlying patterns. This can lead to poor performance on unseen data. To prevent overfitting, we use a validation set during training. The validation set is a subset of data that is not used for updating the weights of the model, but to monitor the performance of the model. The validation can be used to stop a training which is getting worse due to overfitting or divergence, but more importantly to select the best overall model on this validation set. The performance on the validation set can also be used to determine the best hyperparameters for training, such as learning rate, number of layers, batch size, and so forth.

After the model is trained, it can be used for automated segmentation of the images. We may further use a final image subset, the testing set which should not have been used for training or validation, to determine the performance of the model. If the model is only evaluated on the training and validation set, it may appear to perform well, but may not generalize well.

Author Contributions

E.B., B.N., and P.S. have designed the study and supervised the work, C.M. created the application, V.H. tested the application, P.S. and C.M. wrote the manuscript with inputs from all the authors.

Acknowledgements

We are grateful to Véronique Mallouh and Danièle Spehner for helpful discussions and manual segmentations, and to Fabien Alpy for giving access to standardized resin-embedded samples. We acknowledge support from the Institut National de la Santé et de la Recherche Médicale (INSERM), the Centre National pour la Recherche Scientifique (CNRS), the Ligue contre le Cancer and to IGBMC (ANR-10-LABX-0030-INRT). This work of the Interdisciplinary Thematic Institute IMCBio+, as part of the ITI 2021–2028 program of the University of Strasbourg, CNRS and Inserm, was supported by IdEx Unistra (ANR-10-IDEX-0002), and by SFRI-STRAT^{US} project (ANR-20-SFRI-0012) and EUR IMCBio (ANR-17-EURE-0023) under the framework of the France 2030 Program. We acknowledge the use of resources of the French Infrastructure for Integrated Structural Biology FRISBI (ANR-10-INBS-0005) and of Instruct-ERIC.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Data Availability Statement

The complete data (images and annotations: <https://zenodo.org/records/12168941> and results: <https://zenodo.org/records/12518270>) are accessible on Zenodo. The data are available for public research and non-commercial use, to access the data, users must request access to confirm those requirements. The software, its source code, tutorials and documentation are available on GitHub (<https://github.com/Cyril-Meyer/DeepSCEM>)

and shared under MIT license. If you use our tool to perform automatic segmentation for your projects, please cite this paper as a reference of our work.

References

- Belevich, I., M. Joensuu, D. Kumar, H. Vihinen, and E. Jokitalo. 2016. "Microscopy Image Browser: A Platform for Segmentation and Analysis of Multidimensional Datasets." *PLOS Biology* 14, no. 1: e1002340. <https://doi.org/10.1371/journal.pbio.1002340>.
- Belevich, I., and E. Jokitalo. 2021. "DeepMIB: User-Friendly and Open-Source Software for Training of Deep Learning Network for Biological Image Segmentation." *PLOS Computational Biology* 17, no. 3: e1008374. <https://doi.org/10.1371/journal.pcbi.1008374>.
- Berg, S., D. Kutra, T. Kroeger, et al. 2019. "ilastik: Interactive Machine Learning for (Bio)Image Analysis." *Nature Methods* 16, no. 12: 1226–1232. <https://doi.org/10.1038/s41592-019-0582-9>.
- Bosch, C., A. Martínez, N. Masachs, et al. 2016. "Erratum: FIB/SEM Technology and High-Throughput 3D Reconstruction of Dendritic Spines and Synapses in GFP-Labeled Adult-Generated Neurons." *Frontiers in Neuroanatomy* 10: 100. <https://doi.org/10.3389/fnana.2016.00100>.
- Chaurasia, A., and E. Culurciello. 2017. "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation." 2017 *IEEE Visual Communications and Image Processing (VCIP)*, 1–4. <https://doi.org/10.1109/VCIP.2017.8305148>.
- Chen, L.-C., G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2018. "DeepLab: Semantic Image Segmentation With Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, no. 4: 834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>.
- Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. 2018. "Encoder-Decoder With Atrous Separable Convolution for Semantic Image Segmentation." In *Computer Vision—ECCV 2018*, edited by V. Ferrari M. Hebert C. Sminchisescu, and Y. Weis, 833–851. Springer International Publishing. https://doi.org/10.1007/978-3-030-01234-2_49.
- Cretoi, D., M. Gherghiceanu, E. Hummel, H. Zimmermann, O. Simionescu, and L. M. Popescu. 2015. "FIB-SEM Tomography of Human Skin Telocytes and Their Extracellular Vesicles." *Journal of Cellular and Molecular Medicine* 19, no. 4: 714–722. <https://doi.org/10.1111/jcmm.12578>.
- Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. "The Pascal Visual Object Classes (VOC) Challenge." *International Journal of Computer Vision* 88: 303–338. <https://doi.org/10.1007/s11263-009-0275-4>.
- Gallusser, B., G. Maltese, G. Di Caprio, et al. 2023. "Deep Neural Network Automated Segmentation of Cellular Structures in Volume Electron Microscopy." *Journal of Cell Biology*. 222, no. 2: e202208005. <https://doi.org/10.1083/jcb.202208005>.
- He, K., X. Zhang, S. Ren, and J. Sun. 2016a. "Deep Residual Learning for Image Recognition." 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- He, K., X. Zhang, S. Ren, and J. Sun. 2016b. "Identity Mappings in Deep Residual Networks." In *Computer Vision—ECCV 2016*, edited by B. Leibe, J. Matas, N. Sebe, and M. Welling (Eds.), (630–645). Springer International Publishing. https://doi.org/10.1007/978-3-319-46493-0_38.
- Heinrich, L., D. Bennett, D. Ackerman, et al. 2021. "Whole-Cell Organelle Segmentation in Volume Electron Microscopy." *Nature* 599: 1–6. <https://doi.org/10.1038/s41586-021-03977-3>.
- Heinrich, L., J. Funke, C. Pape, J. Nunez-Iglesias, and S. Saalfeld. 2018. "Synaptic Cleft Segmentation in Non-Isotropic Volume Electron Microscopy of the Complete Drosophila Brain." In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2018*, edited by A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, 317–325. Springer International Publishing. https://doi.org/10.1007/978-3-030-00934-2_36.

- Heymann, J. A. W., M. Hayles, I. Gestmann, L. A. Giannuzzi, B. Lich, and S. Subramaniam. 2006. "Site-Specific 3D Imaging of Cells and Tissues With a Dual Beam Microscope." *Journal of Structural Biology* 155, no. 1: 63–73. <https://doi.org/10.1016/j.jsb.2006.03.006>.
- Hirabayashi, Y., H. Iga, H. Ogawa et al. 2024. "Deep Learning for Three-Dimensional Segmentation of Electron Microscopy Images of Complex Ceramic Materials." *NPJ Computational Materials* 10: 46. <https://doi.org/10.1038/s41524-024-01226-5>.
- Ioffe, S., and C. Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *Proceedings of the 32nd International Conference on Machine Learning*, 448–456. <https://proceedings.mlr.press/v37/loffe15.html>.
- Isensee, F., P. F. Jaeger, S. A. A. Kohl et al. 2021. "nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation." *Nature Methods* 18: 203–211. <https://doi.org/10.1038/s41592-020-01008-z>.
- Kingma, D. P., and J. Ba. 2015. "Adam: A Method for Stochastic Optimization." In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, edited by Y. Bengio and Y. LeCun. <http://arxiv.org/abs/1412.6980>.
- Leighton, S. B. 1981. "SEM Images of Block Faces, Cut by a Miniature Microtome Within the SEM—A Technical Note." *Scanning Electron Microscopy Pt 2*: 73–76.
- Liu, J., L. Li, Y. Yang, et al. 2020. "Automatic Reconstruction of Mitochondria and Endoplasmic Reticulum in Electron Microscopy Volumes by Deep Learning." *Frontiers in Neuroscience* 14: 599. <https://doi.org/10.3389/fnins.2020.00599>.
- Meyer, C. 2024a. *DeepSCEM: Examples Datasets*. Zenodo. <https://doi.org/10.5281/zenodo.12168941>.
- Meyer, C. 2024b. *DeepSCEM: Examples Results*. Zenodo. <https://doi.org/10.5281/zenodo.12518270>.
- Meyer, C.,  . Baudrier, P. Schultz, and B. Naegel. 2023. "Combining Max-Tree and CNN for Segmentation of Cellular FIB-SEM Images." In *Reproducible Research in Pattern Recognition*, edited by B. Kerautret, M. Colom, A. Kr henb hl, D. Lopresti, P. Monasse, and B. Perret, 77–90. Springer. https://doi.org/10.1007/978-3-031-40773-4_7.
- Meyer, C., V. Mallouh, D. Spehner,  . Baudrier, P. Schultz, and B. Naegel. 2021. "Automatic Multi Class Organelle Segmentation For Cellular Fib-Sem Images." *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 668–672. <https://doi.org/10.1109/ISBI48211.2021.9434075>.
- Meyer, C., and P. Schultz. 2023. *IGBMC ICube LW4-1: Image and Annotations*. Zenodo. <https://doi.org/10.5281/zenodo.8344292>.
- Milletari, F., N. Navab, and S.-A. Ahmadi. 2016. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation." *2016 Fourth International Conference on 3D Vision (3DV)*, Stanford, CA, USA, 565–571. <https://doi.org/10.1109/3DV.2016.79>.
- Narayan, K., C. M. Danielson, K. Lagarec, et al. 2014. "Multi-Resolution Correlative Focused Ion Beam Scanning Electron Microscopy: Applications to Cell Biology." *Journal of Structural Biology* 185, no. 3: 278–284. <https://doi.org/10.1016/j.jsb.2013.11.008>.
- Ronneberger, O., P. Fischer, and T. Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, 234–241. Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28.
- Spiegelhalter, C., V. Tosch, D. Hentsch, et al. 2010. "From Dynamic Live Cell Imaging to 3D Ultrastructure: Novel Integrated Methods for High Pressure Freezing and Correlative Light-Electron Microscopy." *PLOS ONE* 5, no. 2:e9014. <https://doi.org/10.1371/journal.pone.0009014>.
- Wei, D., S. Jacobs, S. Modla, et al. 2012. "High-Resolution Three-Dimensional Reconstruction of a Whole Yeast Cell Using Focused-Ion Beam Scanning Electron Microscopy." *BioTechniques* 53, no. 1: 41–48. <https://doi.org/10.2144/000113850>.
- Weibel, E. R., G. S. Kistler, and W. F. Scherle. 1966. "Practical Stereological Methods for Morphometric Cytology." *Journal of Cell Biology* 30, no. 1: 23–38. <https://doi.org/10.1083/jcb.30.1.23>.
- Wilhelm, L. P., C. Wendling, B. V die, et al. 2017. "STARD3 Mediates Endoplasmic Reticulum-to-Endosome Cholesterol Transport at Membrane Contact Sites." *Embo Journal* 36, no. 10: 1412–1433. <https://doi.org/10.15252/embj.201695917>.
- Williams, R. C., and F. Kallman. 1955. "Interpretations of Electron Micrographs of Single and Serial Sections." *Journal of Biophysical and Biochemical Cytology* 1, no. 4: 301–314. <https://doi.org/10.1083/jcb.1.4.301>.
- Xiao, C., X. Chen, W. Li, et al. 2018. "Automatic Mitochondria Segmentation for EM Data Using a 3D Supervised Convolutional Network." *Frontiers in Neuroanatomy* 12: 92. <https://doi.org/10.3389/fnana.2018.00092>.