

# Generation and study of the synthetic brain electron microscopy dataset for segmentation purpose

N.A. Sokolov<sup>1</sup>, E.P. Vasiliev<sup>1</sup>, A.A. Getmanskaya<sup>1</sup>

<sup>1</sup> Department of Mathematical Software and Supercomputing Technologies, Lobachevsky University, 603950, Nizhny Novgorod, Russia, Gagarina st. 23

## Abstract

Advanced microscopy technologies such as electron microscopy have opened up a new field of vision for biomedical researchers. The use of artificial intelligence methods for processing EM data is largely difficult due to the small amount of annotated data at the training stage. Therefore, we add synthetic images to an annotated real EM dataset or use a fully synthetic training dataset. In this work, we present an algorithm for the synthesis of 6 types of organelles. Based on the EPFL dataset, a training set of 1161 real fragments 256×256 (ORG) and 2000 synthetic ones (SYN), as well as their combination (MIX), were generated. The experiment of training models for 6, 5-classes and binary segmentation showed that, despite the imperfections of synthetics, training on a mixed (MIX) dataset gave a significant increase (about 0.1) in the Dice metric for 6 and 5 and same results at binary. The synthetic data strategy gives annotations for free, but shifts the effort to producing sufficiently realistic images.

**Keywords:** multi-class segmentation, electron microscopy, neural network, image segmentation, machine learning.

**Citation:** Sokolov NA, Vasiliev EP, Getmanskaya AA. Generation and study of the synthetic brain electron microscopy dataset for segmentation purpose. *Computer Optics* 2023; 47(5): 778-787. DOI: 10.18287/6179-CO-1273.

**Acknowledgements:** The work was carried out with the support of the Priority 2030 program.

## Introduction

Microscopy plays an indispensable role in biomedical research. Advanced technologies of microscopy like electron microscopy have opened up new eyesight for biomedical researchers. The image resolution is very high, which is why a cubic millimeter of brain tissue can take up more than 1000 terabytes. The resulting images are analyzed to identify individual cells. Segmentation is usually done by biologists manually. Processing one experiment takes up to six months of manual work.

One of the first papers which use the serial block scanning electron microscopy as a source of high-resolution three-dimensional nanohistology for cells and tissues was the [1] (2010). A subsequent series of works was aimed at creating datasets for training deep learning networks and DNN methods and models for EM data segmentation designed for binary segmentation of brain cell organelles – neural membranes [2] and supervoxel segmentation of mitochondria [3]. Simultaneously, the problem of 3D reconstruction of the brain neural network and the problem of brain connectomics on the basis of neuron organelles and connections between neurons (synapses) is stated [4]. In this problem, of particular importance is the segmentation of such organelles as postsynaptic densities (PSD), vesicles, and axons.

The invention of U-Net in 2015 [5] opened a series of novel models and adaptations for segmenting brain EM data. The source of U-Net success is in involving the contextual information of the input image at all levels of processing. Almost immediately, the publication [6]

experimentally confirmed that the skip connection of the U-Net architecture is effective for solving segmentation problems in biomedicine.

U-Net also provided a basis for creating numerous models: [7, 8], 3D U-Net [9] (2016), V-Net [10] (2016), DeepMedic [11] (2017), HighRes3DNet [12] (2017), Inception Unet [13] (2020), R2U++ [14] (2022).

The application of artificial intelligence methods for EM data processing is largely hampered by a small amount of labeled data for training and testing DNNs. Open EM data as a whole are represented by only a few labeled datasets, both due to the laboriousness of preparing samples for an electron microscope, and due to the lack of specialists for manual labeling. We found four open EM datasets the earliest and most popular of which are labeled only for one class (mitochondria or membranes). In the two other datasets, several classes are distinguished. As a result, the majority of neural networks used in EM processing are trained only to perform binary segmentation.

In our previous article [15], we presented the first results of creating of an algorithm that synthesizes EM-data and markup. We used synthetic data to supplement the original dataset with a synthetic axon. It should be noted that attempts to synthesize electron microscopy data of highly porous structures well described in an article published in 2022 [16]. Examples of synthetic scanning electron microscopy image data from Fend et al. shown in Fig. 1.

In connection with the above, the main aim of this work is to (1) to improve algorithms for automatic

generation of a dataset of synthetic objects; (2) to develop a parameterization for an algorithm that synthesizes data; (3) to study the capabilities of multiclass segmentation of

U-Net-like architectures, starting with U-Net (in this work), using EM-dataset, synthetic dataset and mixed dataset.



Fig. 1. Examples of synthetic scanning electron microscopy image data from Fend et al [16]

### 1. Data and methods

In this section, we describe publicly available datasets. The most popular datasets for assessing the segmentation of mitochondria were collected by Lucchi et al. in [3].

It is seen that in three of the four labeled open datasets, only one class is labeled. Only one dataset contains more than one labeled class. For this reason, the vast majority of neural networks in EM are trained to classify only two classes (object and background).

Tab. 1. Open labeled electron microscopy datasets

No	Name	Amount of data	Amount of labeled data	Labeled classes	Resolution nm/voxel
1	AC4, ISBI 2013 [17]	4096×4096×1850	1024×1024×40	membranes	6×6×30
2	EPFL [18], Lucchi [3]	1065×2048×1536	2 datasets 1024×768×165	mitochondria	5×5×5
3	Kasthuri et al [19]	1463×1613×85	1334×1553×75	mitochondria	3×3×30
4	UroCell [20, 21]	1366×1180×1056	5 datasets 256×256×256	mitochondria, endolysosomes, fusiform vesicles	16×16×15

We used the dataset EPFL or the data set of mitochondria segmentation Lucchi available at <https://www.epfl.ch/labs/cvlab/data/data-em/>. Initially, these data contain masks only for mitochondria. For this reason, to assess multiclass segmentation algorithms, we manually labeled 27 layers in the training sample (1024×768) and 5 test layers for the following classes: (1) mitochondria, including their boundaries; (2) boundaries of mitochondria; (3) cell membranes; (4) postsynaptic densities (PSD); (5) axon sheaths; and (6) vesicles. Accurate manual labeling of one layer takes ~8 hours. Our labeling of the dataset EPFL is available at <https://github.com/GraphLabEMproj/unet>. We plan to continue the work on labeling and do this for both volumes. It just so happens that the axon sheath in the

training dataset is present only in the first 36 layers and looks completely different from the axon sheath in the test dataset Fig. 2. In the test dataset, the axon is represented in the first 70 layers, changes its shape from elongated to more rounded, and also has a darker interior and inner ring.

For the synthesized dataset, we generated 2000 images of size 256×256 pixels containing the least represented classes—postsynaptic densities and axon sheaths. An example of one labeled sample is shown in Fig. 3, an example of several samples is shown in Fig. 4. The shape, size, and gray levels of compartments are chosen to be similar to the shape, size, and gray levels of the test EPFL dataset. The advantage of a synthetic set is that you can get any number of images you need along with their labeling automatically.

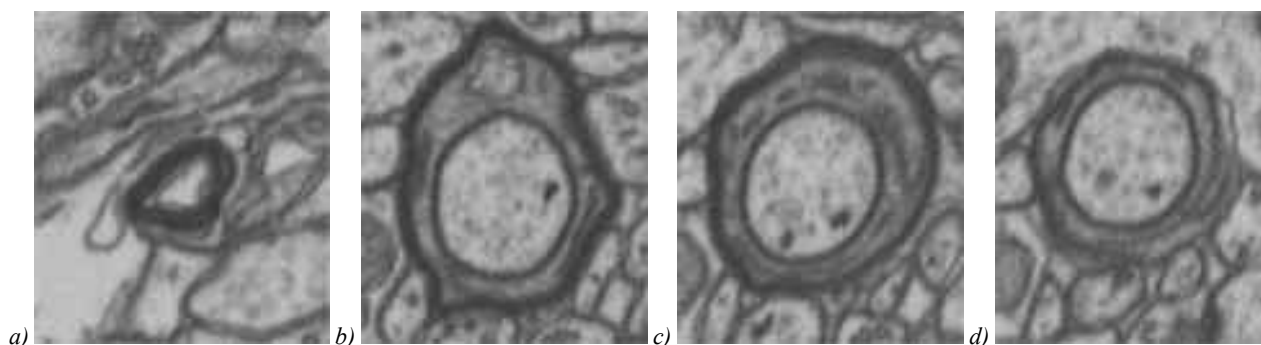


Fig. 2. Axon sheaths in the training and test EPFL datasets: (a) axon sheath in the training set; (b) axon sheath in the test set, first layer; (c) axon sheath in the test set, 35th layer; (d) axon sheath in the test set, 70th layer

The studies presented in this article are the development of our work, in this study, we expand the topic of synthetic image generation by modifying some organelles generation

and adding the ability to parameterize synthetic image generation algorithms to achieve high similarity with different real datasets using a single universal code.



Fig. 3. Example of synthesized data (only nonzero masks are shown): (a) layer, (b) mask of axon sheaths, (c) mask of vesicles, (d) PSD mask

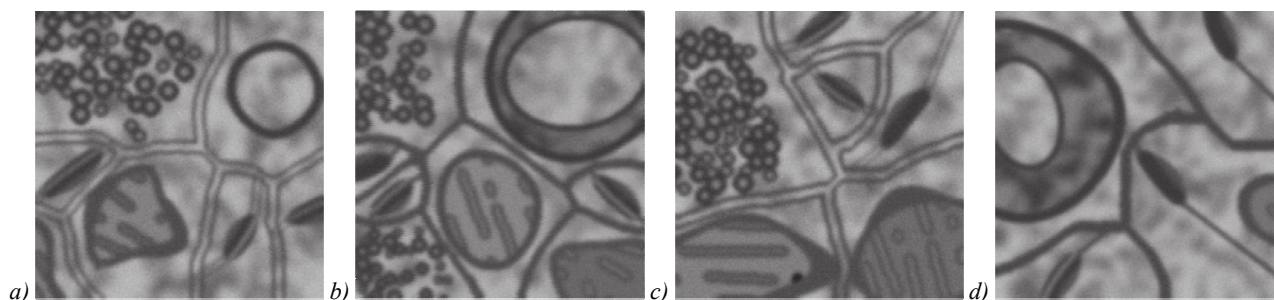


Fig. 4. Example of synthesized data

### 1.1. Algorithm of synthetic data generation

You can view and download the implementation of our algorithms for synthesizing by scanning electron microscopy here: <https://github.com/GraphLabEMproj/Synthetics/>

Next, we provide a brief description of the synthesis algorithms. The generation of synthetic images is the following sequence: 1) generation of the organelles (axon, mitochondria, PSD, vesicles); 2) placing organelles to the image; 3) generation of the membranes; 4) image blurring and noising.

**Organelle location.** The position of an organelle is determined by its central point. All other organelle points (like shell) are calculated relative to the center point. A random position of the center point is chosen in the range of  $[32: image\_size - 32]$  pixels from the edge for the PSD and  $[5: image\_size - 5]$  pixels for the rest of the organelles. The organelle is rotated at a random angle around its center. After generating and rotating the points, we check for intersections with organelles already existing in the layer, and if there is an intersection, then the procedure for setting the location and rotation of the organelle is repeated. If in several iterations (we use 300 iterations) it is not possible to add a new organelle to the layer without intersecting with the existing ones, then it is skipped from the generation.

**Number of organelles** The number of organelles of one or another class is set by parameters. Empirically, we decided to add one axon, 3 mitochondria, 3 PSDs, and 3 vesicle regions in a single  $256 \times 256$  patch. Organelles are added randomly one after the other. However, if it is

physically impossible to arrange a given number of organelles without intersections, then new organelles will not be added. Since for training networks it is important that the images with the class for which training takes place are in sufficient volume. Therefore, for high-quality training, it is necessary that both the number of images and the area of the required class on the layer be sufficient for training (for example, if the class is too small in area, then the optimizer can consider it noise and ignore it, and if there are few examples with the class, then there is a chance that the grid will have time to forget this class). Therefore, we do not focus on the statistics of the initial data, but, on the contrary, add more organelles to the synthetic set, either poorly represented in the initial data (axon) or occupying a small area (PSD).

**Axons.** We generate two types of axons: filled inside and empty (only shell). After that, from 7 to 14 shell points are created around the center of the organelle. The distance from the center to the point is set randomly in some given range (from 15 to 82 pixels for the shell and from 40 to 139 pixels for the axon with internal filling). To draw the shell, a closed spline is used along the points of the shell. To drawing the thickening of the shell, we take a subset of successive shell points and use an open spline with a wider line thickness. In the case of generating an axon with an internal part, the internal part is filled via darkened and blurred background texture. The inner shell is drawn as an ellipse.

**Mitochondria** are generated in the form of an elongated smoothed, slightly asymmetrical figure with a shell of varying thickness. We use from 4 to 10 shell

points created around the center line of the mitochondrion for drawing the spline. To generate an oblique cut of the mitochondrion, an open spline is used, consisting of points located near the most distant parts of the mitochondrion. For the interior of the mitochondrion, a texture is generated, consisting of non-intersecting line segments of different lengths.

**Vesicles.** The generation of vesicles begins with the selection of the area in which the vesicles will be located. After that, a certain number of single vesicles are added to this area - circles of a certain radius, which for the most part are added without intersections.

**PSD** is generated as a curve segment, and additional areas in front of and behind the curve mimic the darkening from the PSD.

**Masks.** All masks of all organelles (axons, mitochondria, vesicles, PSD, membranes) are drawn by changing the color for the drawing algorithm to white. We use organelle's masks to place its to the images without intersection. We make several attempts to add a mask to image to an empty space.

**Membranes.** We use organelle's masks as starting points of the region growth algorithm. Regional boundaries become our membranes. We also add lines from PSD to membranes to connect them. On some tiles we make double thin borders instead of single thick borders to be more like the original dataset.

**Blur and Noise.** We use Gauss filtering and Poisson noise to simulate image blurring and noise from the registration device. All images were blurred with a Gaussian filter with a kernel of radius 7 pixels. According to the Vulović et al. [22] the appearance of Poisson noise is due to the statistical nature of electromagnetic waves such as x-rays, visible light and gamma rays. X-ray sources emit a certain number of photons per unit time. Such sources have random fluctuations in the number of emitted photons. As a result, the resulting image has spatial and temporal randomness corresponding to the Poisson distribution

$$P(\lambda, k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad \lambda > 0, \quad k \in \mathbb{N}.$$

We add Poisson noise with  $\lambda=1$ . It makes the generated images more similar to real-life images.

### 1.2. Personification of generation algorithm for a specific real dataset

Despite the fact that different datasets are obtained from similar biological materials, their stored digital images can vary greatly depending on the equipment used, the scale selected, the quality of the matrix, etc. As a result, datasets of different developers have different statistical characteristics. In order for synthetic datasets to better match real ones, generation algorithms must be parameterized to be able to generate data for a specific type of dataset. In Fig. 6 presented examples images from

datasets EPFL (a) and AC4 dataset (b). It can be seen from their images that the images differ greatly in the average brightness of organelles, the thickness of the lines, and in the blurring of the contours.

In order to make it possible to adjust the systematic set to real data, the characteristics of the generated objects were parameterized. The parameterization touched the average brightness of all objects and the deviation from the average brightness, thickness and color of the borders.

To select the parameters, we used the histograms of the classes marked on the zero (layer0000) layer of EPFL dataset. We compared the histograms of the synthetic dataset with the histograms of the entire labeled dataset (Fig. 5).

Of course, by exactly repeating the distribution as in a histogram, one cannot guarantee the similarity of synthetic data to the original ones, and more, one cannot guarantee a good AI model on them. But this is a fairly simple method for checking the correctness of the algorithm.

As can be seen from the histogram of axons, we did not focus on the histogram of the training dataset and testing dataset, but rather proceeded from the generalized representation of the axon in the electron microscopy data, because axon is the least represented class with the least variability. Nevertheless, this approach gave good results when training an AI model.

Thanks to parameterization, the generator can generate images similar to EPFL (Fig. 6c) and AC4 dataset (Fig. 6d), adjusting to a specific dataset.

### 1.3. Network architecture

U-Net is considered to be a standard convolutional network architecture for image segmentation tasks. This architecture consists of a contracting path for capturing the global context and a symmetric expanding path that enables accurate localization. The basis of this network is the project U-Net <https://github.com/zhixuhao/unet>. In the original project, U-Net was used for the binary classification of membranes. In this work, we use U-Net for multiclass segmentation. We copied the original repository and made modifications in it, which are available at <https://github.com/GraphLabEMproj/unet> together with our labeling of the Lucci data. Following the author of the code at <https://github.com/zhixuhao/unet>, the implementation of U-Net has some differences from the classical U-Net network [5]:

- The network input is an image of size  $256 \times 256 \times 1$ .
- The network output is  $256 \times 256 \times N$ , where  $N$  is the number of classes.
- The sigmoid activation function guarantees that the mask is in the range  $[0, 1]$ .

In addition, we added batch normalization after each ReLU activation layers.



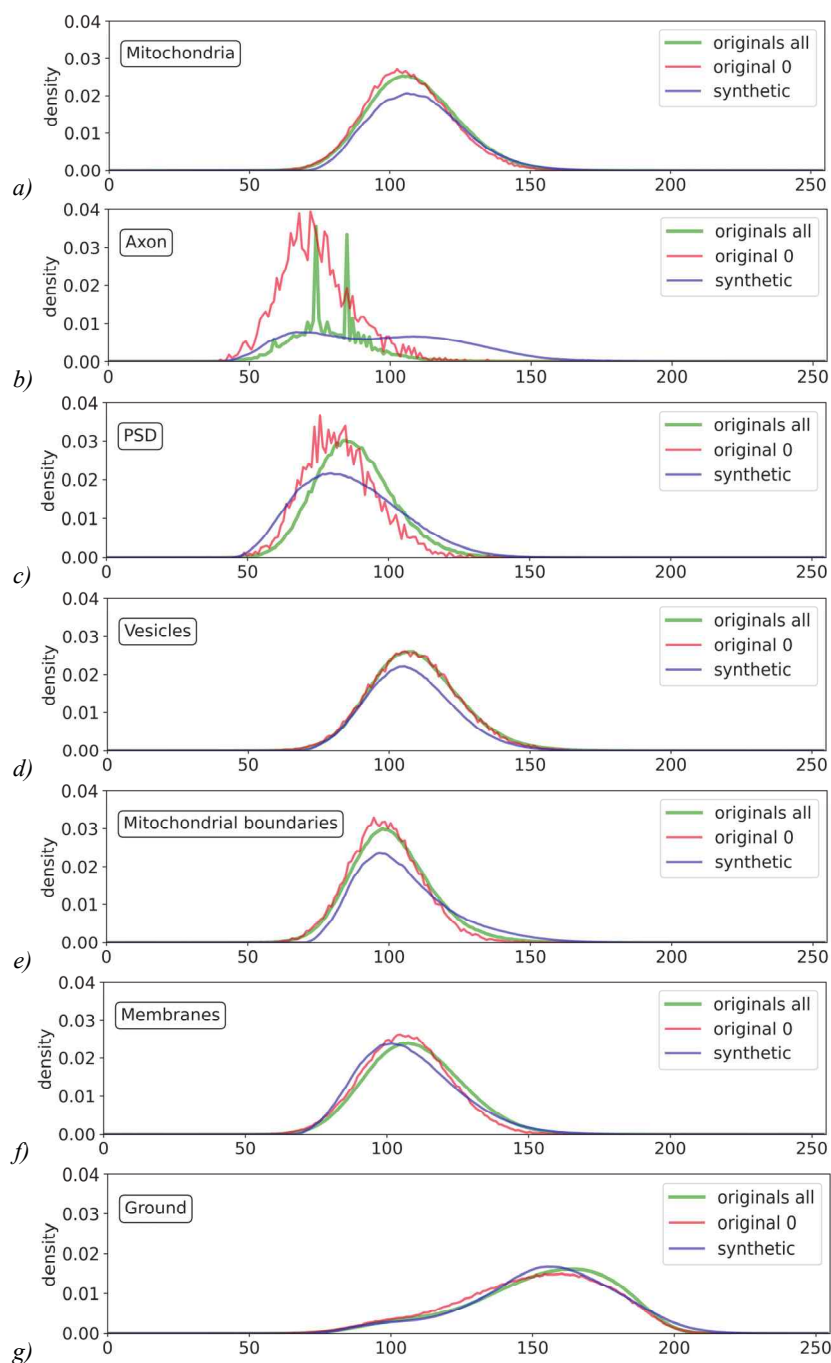


Fig. 5. Class histograms for zero layer of EPFL dataset (original 0), for all labeled layers of EPFL dataset (original all) and for synthetic dataset: (a) mitochondria; (b) axon; (c) PSD; (d) vesicles; (e) mitochondrial boundaries; (f) membranes; (g) ground

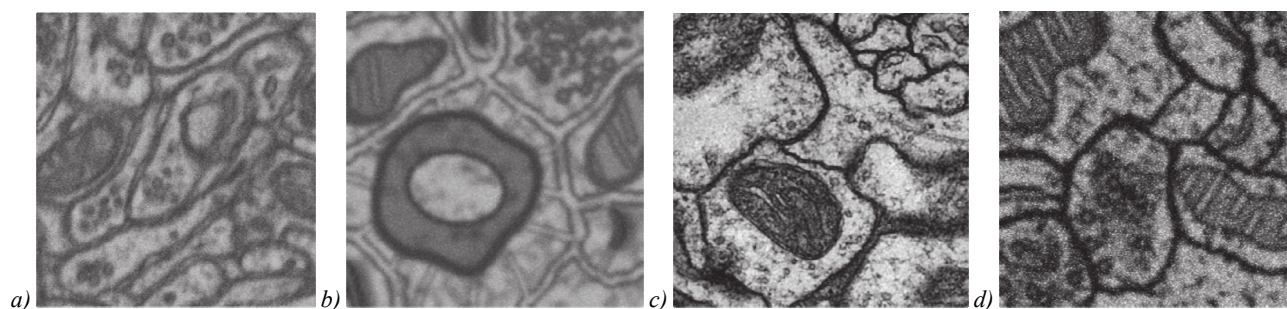


Fig. 6. Parameterization of the generator for various datasets: (a) original EPFL; (b) synthetic EPFL; (c) original AC4 dataset; (d) synthetic AC4 dataset

In the process of improving the model, we constructed more compact modifications of the U-Net model, as a result of which we present tiny-unet-v2 model, which has the following differences from the previous architecture:

Number of channels in original U-Net convolutional blocks:  $64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 1024$ , number of channels in our architecture:  $32 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256$ .

The resulting model contains 15.7 times fewer parameters than the original model and takes up 15.5 times less memory (24 MB instead of 364 MB).

## 2. Experimental results

### 2.1. Assessment criteria

We use the Dice–Sørensen coefficient (DSC), which are usually used for segmenting biomedical images. The values of the DSC vary from zero to one. Define the number of correctly classified pixels as belonging to the target class (true positive) TP, the number of correctly classified background pixels (true negative) TN, the number of erroneously classified pixels as belonging to the target class (false positive) FP, and the number of erroneously classified background pixels (false negative) FN. Then, define the metrics as follows:

$$DSC = \frac{2TP}{2TP + FP + FN}.$$

Since we consider multiclass segmentation in this work, we are interested in multiclass metrics. Since the Dice metrics compare two sets, in the case of multiclass classification the result will be a vector of Dice metrics for each class. For training a neural network, a scalar error function is used. Therefore, for multiclass segmentation, we should convolve the metrics vector. To convolve a vector into a scalar, we use the linear convolution

$$W_{scalar} = \sum_{i=1}^N \lambda_i W_i, \lambda_i \geq 0, \sum_{i=1}^N \lambda_i = 1,$$

where  $\lambda_i$  is a weighting coefficient and  $W_i$  is the value of the DSC for the  $i$ -th class.  $W_{scalar}$  is a scalar value or convolution of a metrics vector, and  $N$  is the number of classes. In this work, we use the linear convolution of DSC with the weighting coefficients  $\lambda_i$  equal to  $1/N$ .

### 2.2. Experiments

We manually labeled 27 EPFL slices ( $1024 \times 768$ ) for the training purposes and 5 slices ( $1024 \times 768$ ) for testing purposes from the EPFL dataset.

**ORG** is a training dataset included only EPFL data. Twenty seven high-resolution images of the original training sample were cut into  $256 \times 256$ ,  $512 \times 512$  and  $768 \times 768$  fragments, with an overlap of a half of the fragment size. In total, ORG initial training part includes 1161 fragments.

**SYN** is a training dataset included only synthetic data. To obtain a synthetic (SYN) training dataset, we generated 2000 synthesized fragments of size  $256 \times 256$ .

**MIX** is a mixed training dataset. It includes 1161 fragments of EPFL data and 2000 synthesized fragments; thus, we have 3161 fragments in total.

To additionally increase the training datasets, we made random rotations of images, random shifts, and random scale changes in a small range (5%). We selected 20 % of images from the training sample into a validation sample. The batch size equal to seven.

All three models were tested on five layers (175 fragments  $256 \times 256$  with an overlap of a half of the fragment size). We used the Adam's optimizer with dynamic learning rate from  $1 \times 10^{-4}$  to  $1 \times 10^{-6}$ . The change in the learning rate starts from the 100th epoch and then decreases every 25 epochs by 5 times.

**Experiment 1.** Five segmentation classes – mitochondria with their boundary, membranes, PSD, axon sheaths, and vesicles. The number of epochs is 200.

**Experiment 2.** Six segmentation classes – mitochondria with their boundary, boundaries of mitochondria, membranes, PSD, axon, and vesicles. The number of epochs is 200. One more class of mitochondria boundaries is added.

**Experiment 3** One segmentation class – mitochondria with their boundary. The number of epochs is 200.

It is seen from Tab. 2 that the quality of multiclass segmentation is only slightly better or slightly lower than binary segmentation in various experiments. The class mitochondria boundaries is a subclass of the class mitochondria with their boundaries, and the additional edge enhancement improves the segmentation results of the unifying class. The network was trained on unbalanced classes, since the sizes of compartments and their occurrence differ dozens of times.

In commercial applications based on deep learning, in addition to quality metrics, the performance characteristics of algorithms also play a large role. Based on the values of the quality metric Dice given in Tab. 2 and 3, we see that with a tenfold decrease in the number of model weights (and hence the execution time), the results of the quality of work remain comparable.

## 3. Discussion

We test our models on the full EPFL test volume and full Lucci++ test volume and use these values instead of the results of the Tab. 2. We cannot directly compare the results from the Tab. 4 because our models were trained on a significantly reduced version of the EPFL dataset (27 layers instead of 165) and we use our own markup.

The Lucchi++ dataset is based on the EPFL Hippocampus dataset, as published in Structured Image Segmentation using Kernelized Features by Lucchi et al [3]. The experts re-annotated the two EPFL Hippocampus stacks. The goal was to achieve consistency for all mitochondria membrane annotations and to correct any

misclassifications in the ground truth labelings. The markup was manually corrected by one senior biologist and additionally validated by two neuroscientists. In

cases of disagreement the biologist corrected the annotations until consensus was reached. The dataset can be loaded there <https://casser.io/connectomics/>.

Tab. 2. Dice coefficient of electron microscopy data segmentation for U-Net model for the original dataset (ORG), the dataset enriched with synthesized images (MIX) and full synthesized dataset (SYN)

Class	Number of classes								
	binary			5			6		
	ORG	MIX	SYN	ORG	MIX	SYN	ORG	MIX	SYN
Mitochondria with the boundaries	<b>0.930</b>	<b>0.930</b>	0.888	0.934	<b>0.946</b>	0.893	<b>0.936</b>	<b>0.938</b>	0.873
Boundaries of mitochondria	-	-	-	-	-	-	0.789	<b>0.800</b>	0.706
Membranes	-	-	-	0.849	<b>0.857</b>	0.774	0.848	<b>0.865</b>	0.763
Postsynaptic densities (PSD)	-	-	-	<b>0.827</b>	<b>0.825</b>	0.669	<b>0.844</b>	0.820	0.653
Vesicles	-	-	-	0.702	<b>0.707</b>	0.562	0.677	<b>0.701</b>	0.542
Axon sheaths	-	-	-	0.426	<b>0.938</b>	0.818	0.353	0.872	<b>0.920</b>
<b>Average</b>	-	-	-	<b>0.747</b>	<b>0.855</b>	<b>0.743</b>	<b>0.741</b>	<b>0.833</b>	<b>0.743</b>

Tab. 3. Dice coefficient of electron microscopy data segmentation for tiny-unet-v2 model for the original dataset (ORG), the dataset enriched with synthesized images (MIX) and full synthesized dataset (SYN)

Class	Number of classes								
	binary			5			6		
	ORG	MIX	SYN	ORG	MIX	SYN	ORG	MIX	SYN
Mitochondria with the boundaries	0.931	<b>0.938</b>	0.890	<b>0.938</b>	0.928	0.887	<b>0.944</b>	<b>0.945</b>	0.843
Boundaries of mitochondria	-	-	-	-	-	-	0.788	<b>0.801</b>	0.611
Membranes	-	-	-	0.847	<b>0.855</b>	0.776	0.850	<b>0.854</b>	0.754
Postsynaptic densities (PSD)	-	-	-	0.812	<b>0.823</b>	0.617	0.789	<b>0.808</b>	0.684
Vesicles	-	-	-	0.691	<b>0.709</b>	0.574	0.690	<b>0.705</b>	0.563
Axon sheaths	-	-	-	0.473	0.856	<b>0.870</b>	0.510	<b>0.915</b>	0.823
<b>Average</b>	-	-	-	<b>0.752</b>	<b>0.834</b>	<b>0.745</b>	<b>0.761</b>	<b>0.838</b>	<b>0.713</b>

Tab. 4. Comparison of mitochondrial segmentation methods

Method	Number of classes	Article	Training dataset labels	Test labels	Dice
HIVE-net[23]	1	[23]	EPFL	EPFL	0.948
3D Casser et al. [24]	1	[23]	EPFL	EPFL	0.942
Cheng et al. (3D) [25]	1	[23]	EPFL	EPFL	0.941
U-Net	6	ours	ours+syn	Lucchi++	0.939
U-Net	1	ours	ours	Lucchi++	0.937
tiny-unet-v2	1	ours	ours+syn	Lucchi++	0.935
3D U-Net [9]	1	[23]	EPFL	EPFL	0.935
tiny-unet-v2	1	ours	ours	Lucchi++	0.934
tiny-unet-v2	5	ours	ours+syn	Lucchi++	0.931
Cheng et al. (2D) [25]	1	[23]	EPFL	EPFL	0.928
U-Net [5]	1	[23]	EPFL	EPFL	0.915
Peng et al. [26]	1	[23]	EPFL	EPFL	0.909
3D Xiao et al. [8]	1	[23]	EPFL	EPFL	0.9
Cetina et al. [27]	1	[23]	EPFL	EPFL	0.864
Lucchi et al. [3]	1	[23]	Lucchi++	Lucchi++	0.86
U-Net	6	ours	ours+syn	EPFL	0.86
U-Net	1	ours	ours	EPFL	0.859
tiny-unet-v2	1	ours	ours	EPFL	0.857
tiny-unet-v2	6	ours	ours+syn	EPFL	0.854

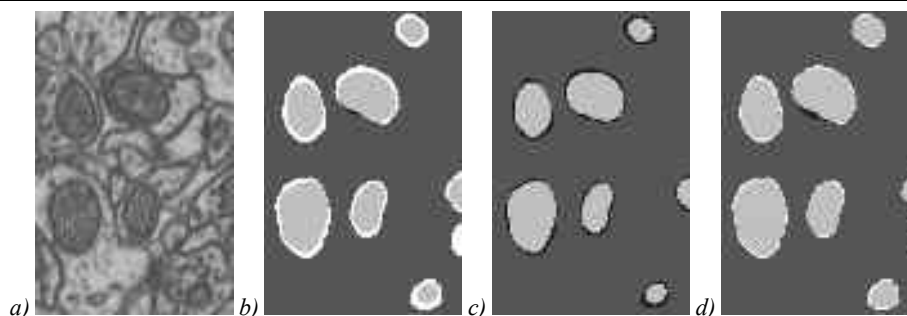


Fig. 7. Markup differences (background – dark gray, intersection areas-light gray): (a) original EPFL slice; (b) EPFL(black)/ours(white) markup; (c) Lucchi++ (black)/EPFL(white) markup; (d) Lucchi++(black)/ours (white) markup

Thus, the table uses three markups of the same dataset, the difference between the markups is shown in the fig. 7.

We calculate differences between markups using the formula

$$\text{difference} = \frac{\text{count\_different\_areas\_in\_pixels}}{\text{count\_intersection\_in\_pixels}}.$$

The difference result by 27 layers is: Lucchi++ vs our 0.09, EPFL vs ours 0.21, Lucchi++ vs EPFL 0.19. This is significant difference. This explains the fact that the our test results are better for Lucchi++ markup. At the same time, it can be noted that Lucchi++, due to enhanced markup checking compared to epfl, like our dataset, more correctly solves the problem as a whole. This can explain the results in Table. 4, in which the model trained on our markup gives Dice metric 0.93+ for Lucchi++, and 0.85+ for EPFL.

We test U-Net (1,5,6 classes) and tiny-unet-v2 (1,5,6 classes) trained via original dataset and dataset using synthetic data. For table. 4 we choose best result for original dataset and for original+synthetic dataset for each model. For example we have tiny-unet-v2 Dice results on Lucchi++ dataset trained via original + synthetic data for 1 class 0.935, for 5 classes 0.931, for 6 classes 0.93. In the table we put only the best result 0.935.

The results with synthetic additive is better. The best results in the table. 4 belong to 3D models. This leads us to the idea of building a 3D synthetic dataset. According to the comparison Tab. 4, our approach using the generation of synthetic datasets gives the segmentation quality comparable to the current results from other researchers in the world on the same class.

No one deep learning model architecture can solve the problem of limited training real-data datasets, and the generation of synthetic data will help to obtain high quality segmentation even on medium and small models, which we have demonstrated using tiny-unet-v2 as an example.

The use of synthetic data for training deep models is becoming more widespread in various fields of science, and in particular in computer vision, and we follow this trend. We developed software for generating synthetic EM brain datasets; main purposes of our work were (1) to improve algorithms for automatic generation of a dataset of synthetic objects; (2) to develop a parameterization for an algorithm that synthesizes data; (3) to study the capabilities of multiclass segmentation of U-Net-like architectures, starting with U-Net (in this work), using EM-dataset, synthetic dataset and mixed dataset.

To synthesize electron microscopy images classic computer graphics methods were used. A parametric model of a slice EM-data was constructed. The gray level parameters were chosen in accordance with the histograms of one labeled layer of the base dataset set. The parameters put into the geometric model were based

on the shape of the organelles in the layers of the simulated data volume. For example, to select the size of the organelle, the largest organelle and the smallest were selected, and the size of the organelles was uniformly selected from this range.

In order to verify the correct statistical distribution of intensities in synthetic images, the gray levels of the resulting synthetic dataset were checked using histograms of the entire available labeled data volume.

We have to further develop the algorithm in collecting statistics of the geometric parameters of organelles and study the influence of parameter variability on the quality of the trained model.

The approach we have developed can be used in two different directions. The first, narrower direction may be the rapid creation of synthetic datasets with the characteristics of individual tissues, and this is demonstrated in this article. Another, broader goal is to create a more versatile synthetic dataset that can be used to segment various types of data with high quality. This option is much more difficult to develop and validate, however, the potential benefit of such a solution due to its universality could be much bigger.

### Conclusion

The algorithm for the automatic generation of a synthetic electron microscopy dataset was developed. The proposed approach allows generating synthetic datasets of any size, as well as quickly changing generation parameters to simulate data from various image registration devices.

As can be seen from the Table. 2 synthetic axon generation gives good results in both purely synthetic and mixed tests. The test results depend on how similar structures were presented in the training and test data. Well-represented mitochondria are spoiled by imperfect synthetics. The using of synthetics for a rarely found axon takes segmentation to a new level of quality.

A fully synthetic dataset that was parameterized according to the real EPFL dataset allows training the neural network with Dice metric results on test dataset from 0.55 to 0.92 for different classes, while training on real data gets results from 0.35 to 0.94, and training on a mix dataset gets results from 0.70 to 0.94. Synthetic data needs to be improved to achieve micro-realistic quality. It is especially necessary to add the "imperfections" of the EM technology, such as rough noise, blurred boundaries, smeared vesicles, and smearing of the mitochondrial interior areas. Further efforts should be made to increase the realism of the images.

### References

- [1] Deerinck T, Bushong E, Lev-Ram V, Shu X, Tsien R, Ellisman M. Enhancing serial block-face scanning electron microscopy to enable high resolution 3-D nanohistology of cells and tissues. *Microsc Microanal* 2010; 16: 1138-1139.
- [2] Ciresan D, Gambardella L, Giusti A, Schmidhuber J. Deep neural networks segment neuronal membranes in electron

- microscopy images. NIPS'12: Proc 25th Int Conf on Neural Information Processing Systems 2012; 2: 2852-2860.
- [3] Lucchi A, Smith K, Achanta R, Knott G, Fua P. Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features. *IEEE Trans Med Imaging* 2012; 31: 474-486.
  - [4] Helmstaedter M, Mitra P. Computational methods and challenges for large-scale circuit mapping. *Curr Opin Neurobiol* 2012; 22: 162-169.
  - [5] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)* 2015: 234-241.
  - [6] Drozdal M, Vorontsov E, Chartrand G, Kadoury S, Pal C. The importance of skip connections in biomedical image segmentation. In Book: Carneiro G, Mateus D, Peter L, Bradley A, Tavares JMRS, Belagiannis V, Papa JP, Nascimento JC, Loog M, Lu Z, Cardoso JS, Cornebise J, eds. *Deep learning and data labeling for medical applications*. Cham: Springer International Publishing AG; 2016: 179-187.
  - [7] Fakhry A, Zeng T, Ji S. Residual deconvolutional networks for brain electron microscopy image segmentation. *IEEE Trans Med Imaging* 2017; 36: 447-456.
  - [8] Xiao C, Liu J, Chen X, Han H, Shu C, Xie Q. Deep contextual residual network for electron microscopy image segmentation in connectomics. *2018 IEEE 15th Int Symp on Biomedical Imaging (ISBI)* 2018: 378-381.
  - [9] Çiçek Ö, Abdulkadir A, Lienkamp S, Brox T, Ronneberger O. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In Book: Ourselin S, Joskowicz L, Sabuncu MR, Unal G, Wells W, eds. *Medical image computing and computer-assisted intervention – MICCAI 2016. Pt II*. Cham: Springer International Publishing AG 2016: 424-432.
  - [10] Milletari F, Navab N, Ahmadi S. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth Int Conf on 3D Vision (3DV)* 2016: 565-571.
  - [11] Kamnitsas K, Ledig C, Newcombe V, Simpson J, Kane A, Menon D, Rueckert D, Glocker B. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med Image Anal* 2017; 36: 61-78.
  - [12] Li W, Wang G, Fidon L, Ourselin S, Cardoso M, Vercauteren T. On the compactness, efficiency, and representation of 3D convolutional networks: Brain parcellation as a pretext task. In Book: Niethammer M, Styner M, Aylward S, Zhu H, Oguz I, Yap P-T, Shen D, eds. *Information processing in medical imaging*. Cham: Springer International Publishing AG; 2017: 348-360.
  - [13] Zhang Z, Wu C, Coleman S, Kerr D. DENSE-INception U-net for medical image segmentation. *Comput Methods Programs Biomed* 2020; 192: 105395.
  - [14] Mubashar M, Ali H, Grönlund C, Azmat S. R2U++: A multiscale recurrent residual U-Net with dense skip connections for medical image segmentation. *Neural Comput Appl* 2022; 6: 17723-17739.
  - [15] Getmanskaya AA, Sokolov NA, Turlapov VE. Multiclass U-Net segmentation of brain electron microscopy data using original and semi-synthetic training datasets. *Program Comput Softw* 2022; 48: 164-171.
  - [16] Fend C, Moghiseh A, Redenbach C, Schladitz K. Reconstruction of highly porous structures from FIB-SEM using a deep neural network trained on synthetic images. *J Microsc* 2021; 281(1): 16-27.
  - [17] Arganda-Carreras I, Turaga SC, Berger DR, Cireşan D, Giusti A, Gambardella LM, Schmidhuber J, Laptev D, Dwivedi S, Buhmann JM, Liu T, Seyedhosseini M, Tasdizen T, Kamensky L, Burget R, Uher V, Tan X, Sun C, Pham TD, Bas E, Uzunbas MG, Cardona A, Schindelin J, Seung HS. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Front Neuroanat* 2015; 9: 142.
  - [18] Lucchi A, Smith K, Achanta R, Knott G, Fua P. Electron microscopy dataset. 2023. Source: <https://www.epfl.ch/labs/cvlab/data/data-em/>.
  - [19] Kasthuri N, Hayworth KJ, Berger DR, Schalek RL, Conchello JA, Knowles-Barley S, Lee D, Vázquez-Reina A, Kaynig V, Jones TR, Roberts M, Morgan JL, Tapia JC, Seung HS, Roncal WG, Vogelstein JT, Burns R, Sussman DL, Priebe CE, Pfister H, Lichtman JW. Saturated reconstruction of a volume of neocortex. *Cell* 2015; 162(3): 648-661.
  - [20] Mekuč MŽ, Bohak C, Hudoklin S, Kim B, Romih R, Kim M, Marolt M. Automatic segmentation of mitochondria and endolysosomes in volumetric electron microscopy data. *Comput Biol Med* 2020; 119: 103693.
  - [21] MancaZerovnikMekuc / UroCell. The UroCell dataset. 2023. Source: <https://github.com/MancaZerovnikMekuc/UroCell>.
  - [22] Vulović M, Ravelli R, Van Vliet L, Koster A, Lazić I, Lücken U, Rullgård H, Öktem O, Rieger B. Image formation modeling in cryo-electron microscopy. *J Struct Biol* 2013; 183: 19-32.
  - [23] Yuan Z, Ma X, Yi J, Luo Z, Peng J. HIVE-Net: Centerline-aware hierarchical view-ensemble convolutional network for mitochondria segmentation in EM images. *Comput Methods Programs Biomed* 2021; 200: 105925.
  - [24] Casser V, Kang K, Pfister H, Haehn D. Fast mitochondria detection for connectomics. *Proc Mach Learn Res* 2020; 121: 111-120.
  - [25] Cheng H, Varshney A. Volume segmentation using convolutional neural networks with limited training data. *2017 IEEE Int Conf on Image Processing (ICIP)* 2017: 590-594.
  - [26] Peng J, Yuan Z. Mitochondria segmentation from EM images via hierarchical structured contextual forest. *IEEE J Biomed Health Inform* 2020; 24: 2251-2259.
  - [27] Cetina K, Buenaposada J, Baumela L. Multi-class segmentation of neuronal structures in electron microscopy images. *BMC Bioinformatics* 2018; 19: 298.

### Authors' information

**Nikolai Alexandrovich Sokolov**, (b. 1998) graduated from Institute of Information Technology, Mathematics and Mechanics of Lobachevsky State University of Nizhny Novgorod. Works as a junior researcher in Lobachevsky State University of Nizhny Novgorod. Research interests: medical data segmentation, image processing, deep learning. E-mail: [nikolai.sokolov@itmm.unn.ru](mailto:nikolai.sokolov@itmm.unn.ru).

**Evgenii Pavlovich Vasiliev**, (b. 1993), graduated from Institute of Information Technology, Mathematics and Mechanics of Lobachevsky State University of Nizhny Novgorod. Works as a junior researcher in Lobachevsky State



---

University of Nizhny Novgorod. Research interests: medical data segmentation, image processing, deep learning, artificial intelligence. E-mail: [evgeny.vasiliev@itmm.unn.ru](mailto:evgeny.vasiliev@itmm.unn.ru).

**Alexandra Alexandrovna Getmanskaya**, (b. 1986) graduated from Computational Mathematics and Cybernetics faculty of Lobachevsky State University of Nizhny Novgorod. Works as a junior researcher in Lobachevsky State University of Nizhny Novgorod. Research interests: medical data visualization, image processing, deep learning. E-mail: [alexandra.getmanskaya@itmm.unn.ru](mailto:alexandra.getmanskaya@itmm.unn.ru).

---

*Received January 9, 2023. The final version – May 23, 2023.*

---