

FEATURE EXTRACTION OF ORGANELLE NETWORKS

Anamika Agrawal^{*}, Parikshit Jain[†], and Ishan Mehta⁺

^{*}Department of Physics, University of California San Diego

[†]Department of Mechanical and Aerospace Engineering, University of California San Diego

⁺Department of Electrical and Computer Engineering, University of California San Diego

ABSTRACT

Cellular organelles in extended cells sometimes exist in a networked state, to fasten transport of synthesis material to different domains of the cell. Popular examples include mitochondrial networks and Endoplasmic Reticulum (ER) networks. These network structures, when perturbed, often lead to declining cell health. Thus it is important to study the physical properties of these networks and their effect on transport in the cell. Neural networks are now increasingly used for image segmentation and feature extraction in biomedical fields. In our paper, we describe a workflow to denoise, segment and extract the network skeleton from fluorescent images of ER networks.

Keywords— U-net, Biology, Image Processing, Deep Learning, Segmentation

1. INTRODUCTION

1.1. Background

Biological networks are observed at many scales, and often serve important transport functions. Examples of such networks include dynamic organelle networks of Mitochondria and Endoplasmic Reticulum (ER) that exist in various extended cells. Such organelle networks lead to better transport of cargo such as protein throughout a large eukaryotic cell. Several studies point to the importance of the topological properties of such networks (such as the number of nodes, loops and branches) in maintaining optimum cellular health and function [1, 2, 3].

In order to study the role of topological properties of a network and its contribution to cargo transport within a cell, we need to visualize the network clearly. The most common way of imaging an ER network itself is by fluorescent tagging of membrane proteins that decorate the ER surface [4]. However, not only is fluorescent tagging a time and labor intensive process, it often leads to the introduction of a lot of background noise in the images. Moreover, fluorescent protein tagged images still need human intervention to separate out parts of the network from the rest of the image. Hence,

it is important to investigate noise-reduction and automation techniques for extraction of properties of the network. In this project our input to the algorithm is an images of Mitochondria network with the fluorescent noise on it. We will first perform preprocessing as per section 2.2 to remove noise and then use neural network to form a deep learning model as mentioned in the section 2.3 to predict the network. Finally, we will form skeleton of the image as per section 2.4.

1.2. Previous Work

There has been some success previously in utilizing ML techniques for automation of biomedical segmentation. One of the most successful neural network is the U-net implementation [5]. In the context of organelle segmentation, it has been leveraged successfully for label-free determination of organelle structure from grayscale images [6] as well as for segmentation of organelles from 3D images [7]. Apart from U-net segmentation of organelle networks, interactive softwares like ImageJ/FIJI [8] and Ilastik [9] exist which provide a reasonably good interface for beginners to experiment with machine learning based algorithms like RFC and WEKA segmentation and train using their own custom data. We stick to the more popular U-net algorithm for our problem, coupling it iteratively with preprocessing and binary masking using FIJI/ImageJ to improve our training accuracy and fine-tune it for the kind of fluorescence data we have. In addition, it circumvents manual interference of having to train data by user and thus makes the entire process for ER-network dataset very fast.

2. METHODS

2.1. Description of Dataset

Our training dataset comprises of ER network dataset provided by the Avezov group at Cambridge. The images are that of an illuminated network within COS7 cells, which are relatively flat and make the network 2D and easy to study. The dataset is multi-channel fluorescence data, with illuminated network and photoactivated proteins spreading over time. The photoactivated proteins offer valuable information

on the transport dynamics of an ER network; however for the purposes of network segmentation they create a lot of background noise as well as increase the intensity threshold making the rest of the network dimly lit in comparison. By using all time frames of the .tif files, we trained on 504 samples and validated on 126 samples. Normalization of images and masks was done appropriately (only the masks needed to be normalized). Description on training data labels is further elaborated in the U-net implementation section.

2.2. Preprocessing of Images

Every image acquired by a microscope exhibits inhomogeneous illumination mainly because a nonuniform light source or optical path often yields shading around edges as well as bright spots of fluorescence light to track the movement of the proteins within the organelle network. These intensities usually vary by 10–30%, thus corrupting accurate segmentation and intensity measurements. An example of the image with illumination of fluorescence has been shown in the Figure 1. In addition, fluorescent proteins emit their own light, and particles residing above and below the desired plane of focus often radiate light causing blurring of the image details. Also, the fluorescence light presents an amount of static and dynamic errors in particle tracking as presented in [10]. Other methods that have been presented on the deconvolution methods for 3-D fluorescence microscopy images has been presented in [11]. One approach of reduce the fluorescence light present

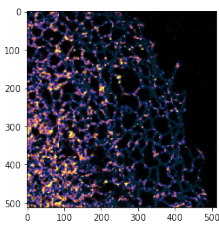


Fig. 1. An image of Organelle Network with the fluorescence light. The presence of fluorescence light corrupts the accurate image segmentation

on the image is to visualize the RGB image in one particular channel. It can be observed from the Figure 1 that visualizing the image in the Green channel reduces the amount of fluorescence signal on the image significantly.

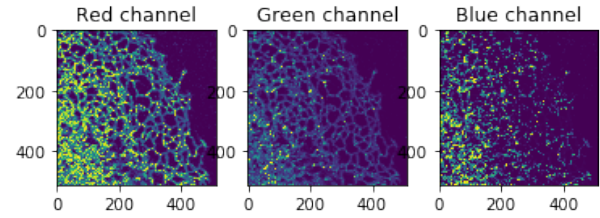


Fig. 2. Visualizing the image of Figure 1 in different color channels of color. Green channel reduces the fluorescence slightly.

As it can be seen in the Figure 3 that by visualizing the image on the the Green channel produced a slightly better version of the Organelle network. The result further improved after using the algorithm 2. The representation of the network gets more clear and the amount of the fluorescent light has been reduced significantly.

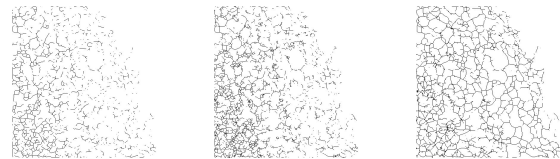


Fig. 3. Left: Network generation from original Image of the Figure 1; Center: Network generation from the Green channel Image from the Figure 2; Right: Network generation after preprocessing from Algorithm 2

2.3. U-net implementation

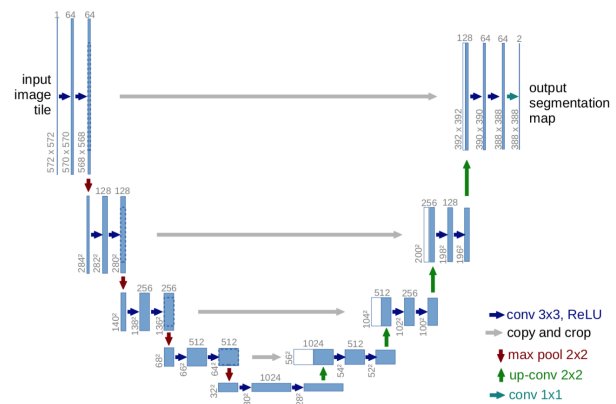


Fig. 4. U-net architecture

A U-net is a fully convolutional network, with a U-shaped architecture as shown in Figure 4. It performs the task of net

Algorithm 1 Preprocessing of the images for the U-net implementation

Require: Image data with the florescence light ($Image_{Original}$)

Ensure: Image with reduced amount of florescence for forming better Skeleton.

```

1:  $image1 = cv2.imread(Image_{Original})$ 
2:  $RGB_{img} = cv2.cvtColor(image1, BGR2RGB)$ 
3:  $b1 = RGB_{img}.copy()$ 
4: for  $i$  in range(0, size( $RGB_{img}$ ), 1) : do
5:   for  $j$  in range(0, 512, 1) : do
6:     if  $RGB_{img}[:, :, 0][i][j] > 180$  : then
7:       if  $RGB_{img}[:, :, 1][i][j] > 100$  : then
8:         if  $RGB_{img}[:, :, 2][i][j] < 255$  : then
9:            $b1[:, :, 0][i][j] = 4$ 
10:           $b1[:, :, 1][i][j] = 33$ 
11:           $b1[:, :, 2][i][j] = 48$ 
12:         end if
13:       end if
14:     end if
15:   end for
16: end for
17:  $Output_{Image} = b1[:, :, 1]$ 

```

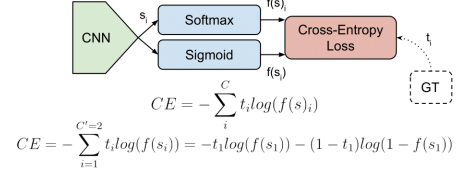
only categorizing each pixel into a class (either belonging to a structure or belonging to the background, in our case), but also retains the spatial location of each pixel. For this purpose, the first part (the contracting part) consists of downsampling layers, and the second part (the expansive part), symmetric to the first part, consists of upsampling layers, so that the input and output layers have the same pixel size. In the contracting path of the U-net, the size of the image is halved 3 times consecutively, but simultaneously the depth of the image increases. Since the image size is reduced, the location information is lost. It is recovered again in the expansion layers through transposed convolution.

A major strength of using U-net for segmentation is that it circumvents the requirement of extensive amounts of training data. In our case, we used preliminary image and mask pairs obtained from ImageJ modules, since we did not already have labels for our data. We required 3 unique time-series .tif files, with each frame counted as an individual image (since the network is dynamic) with an attached mask for each frame. This did not require a lot of preprocessing; however, results can be improved by improving the mask labels attached and by including more unique images for training.

For our U-net implementation, we used the Adam optimizer with a learning rate $1e-4$. While each convolution layer had exponential linear unit ('elu') as activation, the final output layer has 'sigmoid' activation given by the following equation.

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (1)$$

The loss function was binary cross-entropy, given by



2.4. Post-Processing and Skeletonization

The skeletonization process we use is very important for future attribute extraction. We input a black and white mask image (i.e. a binary mask) and then we use a cascade of denoising, erosion and convolution filters to erode the thick lines of the binary mask into skeletonized representation. Python lends itself to the thinning skeletonization algorithm. Thinning can be described by the hit and miss transform such that: $thin(i, j) = I - H(I, J)$ where H is the hit or miss transform used. For each element we assume that the center/origin of the element is the center of a 3x3 cross. Once we apply this for corner finding, we OR the image set of the corners together to get our final image of right angles. Our algorithm combines this idea with popular erosion functions to lead to the final algorithm shown below:

Algorithm 2 Post-processing of segmented binary mask to create skeleton

Require: Binary Mask of image to be skeletonized

Ensure: Final Skeletonized image in binary form

```

1:  $skel = Initialize\ skeleton\ of\ size\ image\ as\ zeros$ 
2:  $Cross = Morphological\ cross\ of\ 3 \times 3\ shape$ 
3: while True: do
4:    $open = Eroded\ then\ dilated\ image\ using\ cross\ as\ kernel$ 
5:    $sub = subtract(img, open)$ 
6:    $eroded = erode(img, kernel = cross)$ 
7:    $skel = img\ OR\ sub$ 
8:    $img = eroded$ 
9:   if a then all pixels of  $img = 0$ 
10:    break
11:  end if
12: end while
13:  $Output_{Image} = skel$ 

```

A mathematical explanation of the erosion and dilation parts of the algorithm can be seen below. Erosion takes the form:

$$A \ominus B = \{x \in E^N | x + b \in A \text{ for every } b \in B\} \quad (2)$$

Whereas dilation takes the form:

$$A \oplus B = \{c \in E^N | c = a + b \text{ for some } a \in A \text{ and } b \in B\} \quad (3)$$

Here B is the kernel or structuring element i.e. the 3x3 cross for our specific case, while A is the image we are looking at. As discussed the open operation is an erosion followed by a dilation. When we put these together we get the following skeletonized image results for a given binary mask:

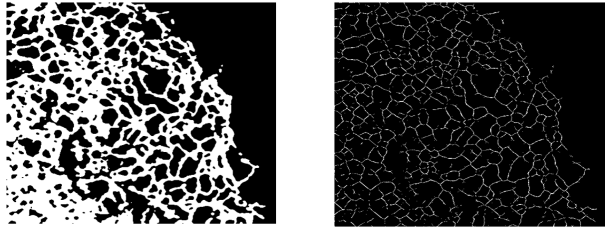


Fig. 5. The left image shows the binary mask, and the right is the skeletonized image.

3. RESULTS

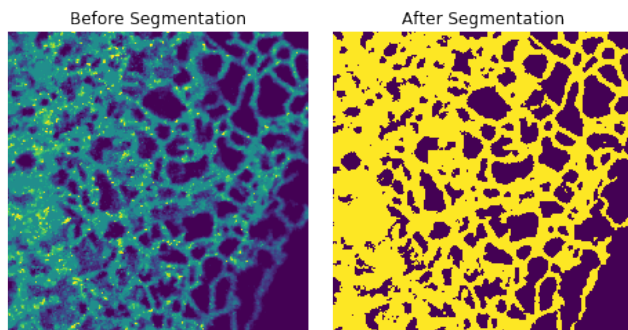


Fig. 6. U-net segmentation : before and after

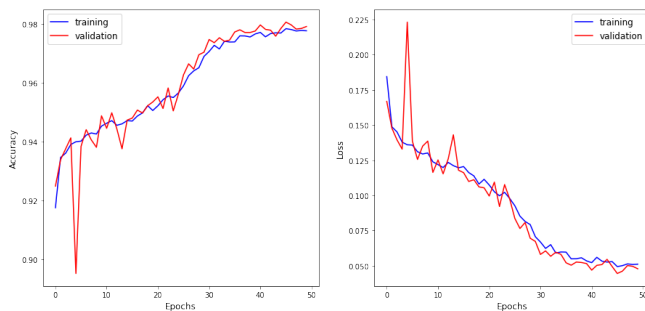


Fig. 7. Loss and Accuracy curves

The results of applying the U-net segmentation to a noisy fluorescent image (without pre-processing filters applied) can be seen in Figure 6. We see that despite the input to the U-net having high-intensity fluorescent protein signal, the

segmentation results are reasonably good. In addition, a great strength of this method of segmentation, which is not seen in simpler classic algorithms, is the ability to segment peripheral regions of the network with low illumination.

Our non-quantitative results from preprocessing and post-processing are elucidated in the methods section 2. We include our results on loss and accuracy here.

Training the U-net model for 20 epochs yields considerably good accuracy, but we trained the model for 50 epochs. The results are in Figure 7. Our model accuracy and loss start to saturate after 30 epochs. We are able to achieve a final converging accuracy of 0.97 for both training and validation data. There are some sudden spikes in both curves, which can potentially be resolved by lowering the learning rates further. However, they are not seen in later epochs.

4. CONCLUSION AND FUTURE DIRECTIONS

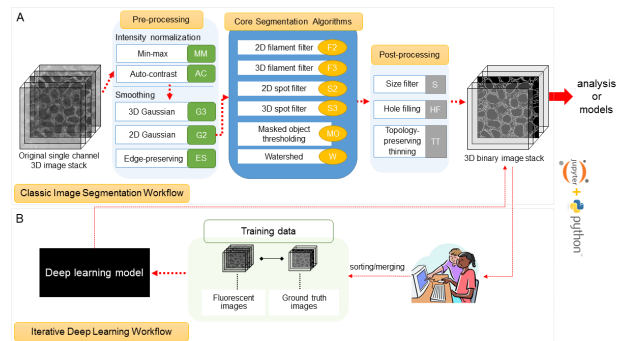


Fig. 8. Iterative workflow for organelle segmentation (courtesy: Allen Institute)

Our work presents a preliminary workflow to segment ER-networks from fluorescence data. There are certainly many improvements that can be done. We can iteratively make the training data better (Fig 8), by using classic image processing and segmentation algorithms and our U-net algorithm. This should make our loss and accuracy curves look even better, and make the segmentation more robust. Other things on our list are:

- Currently we are able to get a skeleton from the network. We want to apply graph-traversal algorithms which would enable us to extract average node density, edge length and loop numbers from the network. These features would give quantitative metrics to distinguish networks.
- We want to apply problem-specific workflows in pre-processing steps, which would enable us to distinguish other organelles (eg nuclei, golgi apparatus).

5. AUTHOR CONTRIBUTIONS

Each author contributed to the planning of the project as well as the final report writing. All codes had feedback from all authors, but each author was chiefly responsible for one module. Parikshit Jain was responsible for the preprocessing of images, Anamika Agrawal for the U-net segmentation algorithm, and Ishan Mehta for the post-processing part. All our code can be found on [this github link](#), with training/test data available upon request.

6. REFERENCES

- [1] Zamponi, N., Zamponi, E., Cannas, S. A., Biloni, O. V., Helguera, P. R., and Chialvo, D. R., “Mitochondrial network complexity emerges from fission/fusion dynamics,” *Scientific reports*, Vol. 8, No. 1, 2018, pp. 1–10, <https://www.nature.com/articles/s41598-017-18351-5.pdf>.
- [2] Viana, M. P., Brown, A. I., Mueller, I. A., Goul, C., Koslover, E. F., and Rafelski, S. M., “Mitochondrial fission and fusion dynamics generate efficient, robust, and evenly distributed network topologies in budding yeast cells,” *Cell Systems*, 2020, <https://www.ncbi.nlm.nih.gov/pubmed/32105618>.
- [3] Brown, A. I., Westrate, L. M., and Koslover, E. F., “Impact of global structure on diffusive exploration of organelle networks,” *Scientific Reports*, Vol. 10, No. 1, 2020, pp. 1–13, <https://arxiv.org/pdf/1905.05320.pdf>.
- [4] Rolls, M. M., Hall, D. H., Victor, M., Stelzer, E. H., and Rapoport, T. A., “Targeting of rough endoplasmic reticulum membrane proteins and ribosomes in invertebrate neurons,” *Molecular biology of the cell*, Vol. 13, No. 5, 2002, pp. 1778–1791.
- [5] Ronneberger, O., Fischer, P., and Brox, T., “U-net: Convolutional networks for biomedical image segmentation,” *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.
- [6] Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F., and Johnson, G. R., “Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy,” *Nature methods*, Vol. 15, No. 11, 2018, pp. 917–920.
- [7] Chen, J., Ding, L., Viana, M. P., Hendershott, M. C., Yang, R., Mueller, I. A., and Rafelski, S. M., “The Allen Cell Structure Segmenter: a new open source toolkit for segmenting 3D intracellular structures in fluorescence microscopy images,” *bioRxiv*, 2018, pp. 491035, <https://www.biorxiv.org/content/10.1101/491035v1.full.pdf>.
- [8] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al., “Fiji: an open-source platform for biological-image analysis,” *Nature methods*, Vol. 9, No. 7, 2012, pp. 676–682.
- [9] Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al., “ilastik: Interactive machine learning for (bio) image analysis,” *Nature Methods*, 2019, pp. 1–7.
- [10] Savin, T. and Doyle, P. S., “Static and Dynamic Errors in Particle Tracking Microrheology,” *Biophysical Journal*, Vol. 88, No. 1, 2005, pp. 623 – 638.
- [11] Sarder, P. and Nehorai, A., “Deconvolution methods for 3-D fluorescence microscopy images,” *IEEE Signal Processing Magazine*, Vol. 23, No. 3, 2006, pp. 32–45.

Response to Critique on : Feature Extraction of Organelle Networks

Anamika Agrawal, Parikshit Jain, Ishan Mehta

June 14, 2020

Group 26

Note: We have not included positive comments, only the critical feedback.

1 Group 32

- *It's kind of hard to get the big picture of the work for someone without a biology background. I could understand why they wanted to use ML and how they used ML, but not the subject they are talking about. But I guess that this is not the point of the course?*

We have added more background and motivation in our report. We address this in the introduction and abstract for our paper. To summarize it here, cell segmentation is important for our project because our goal is to study the effect of network properties like node density, average edge length etc, but these properties can't be obtained unless we have a skeleton of the network after segmentation. We use ML for segmentation as it is the fastest, most accurate and convenient way to do so.

2 Group 21

- *For the preprocessing part of the slides, you use subtitle "Details on feature extraction: Preprocessing", but only give why preprocessing is needed and the result of preprocessing without giving how it is done.*

We address the specific steps taken to do the preprocessing in the paper and also in the demo, however we understand that this may not have been made as clear during the slides.

- *I think the reason that you only use U-net is not strong enough. Although you mentioned U-net has been observed to be the best network for bio-medical segmentation, it would be better if you can compare with other network structures. Even just give a table and list the performance other works achieved by different network would be better for people who are not familiar with this topic.*

We have addressed this concern in the 'Previous Work' section of the paper. U-net is faster for testing on large sets of data once the model is trained for a specific purpose

(in our case, identifying ER-network). We attempted to use the existing alternative on ImageJ's inbuilt WEKA segmentation. However, to get the accuracy that can be achieved with U-net, we needed to manually train several images which took a lot of time.

3 Group 27

- *After watching the code demo, I realized that the background section didn't exactly tell us what the objective of the project was. I believe you wanted to make a segmentation tool to analyze the organelle morphology, making nodes and branches clearer for future researchers who work with fluorescence microscopy; however, the intro slides implied the use of ML to correlate the morphology/edge length to the encounter time, possibly informing us more about cellular function, death or physiology.*

We will try to be clearer about the background section on the paper, but the final conclusion that you made about our ethos was correct. The introduction slide about correlating the edge length with the encounter time was to give an example of how might the network morphology affect cargo spreading times and to make our motivation clearer. We apologize if it was misleading.

- *Some metrics on the model performance in the interactive model slide would have been nice.*

Due to a bug in the code, the loss and accuracy metric during the time of the presentation were not reasonable. The final code has been debugged and accuracy metric has been added to the final report.

- *The presentation could have had more metrics about the models in the slides to give the audience more time to process the many layers of processing. Overall, great work and a useful tool for cellular imaging research. This was shown in every aspect of the presentation and code demo.*

Same as above. We have added a metric showing the accuracy of our segmentation in the final report.

Final note: We got some comments about our transition between slides being abrupt. Zoom was a bit awkward to use for transitioning since we are new to using it, in addition we had some network issues. Hopefully we will transition into being better with that (and with our presentation skills).