UC San Diego
JACOBS SCHOOL OF ENGINEERING

Neuronal Circuit and Synapse Analysis with Deep Neural Networks

Advisors: Dr. Daniela Boassa, Dr. Matthias Haberl

Team Members: Faezeh Ghazi, Jason Kha

Introduction

Biomedical image segmentation has led to numerous breakthroughs in neuroscience research by helping scientists map the brain. Brain mapping could be the key to understanding the progression of neurodegenerative diseases such as Alzheimer's and Parkinson's disease. Currently, the most accurate neural segmentations and synapse detections are annotated manually. However, manual annotation of synapses is unsustainable. The electron microscopy datasets of the brain are very large, on the scale of 75 million neurons for a mouse brain and 86 billion neurons for the human brain. Even a fruit fly brain with only 100,000 neurons requires 100 TB of disk space.

CDeep3M is a deep learning software toolkit that automates the image segmentation of a variety of organelles in 3D microscopy data. For CDeep3M, the segmentations for synapses require the most improvement. There are thousands of small objects in each presynaptic terminal, also known as vesicle clouds, some with sharply delineated membranes and others with much blurrier edges. This makes it a particularly difficult task for human and computer segmentation to identify individual synapses.

Problem Statement

The current CDeep3M neural network model that is trained for synapse segmentations of serial block-face scanning electron microscopy (SBEM) predicts too many false positives. We identify these synapse segmentations as false positives based on biological knowledge.

The objective of this capstone project is to use 3D image processing and deep learning to improve the accuracy of automated synapse segmentations compared to the existing CDeep3M synapse model.

Data Science Pipeline

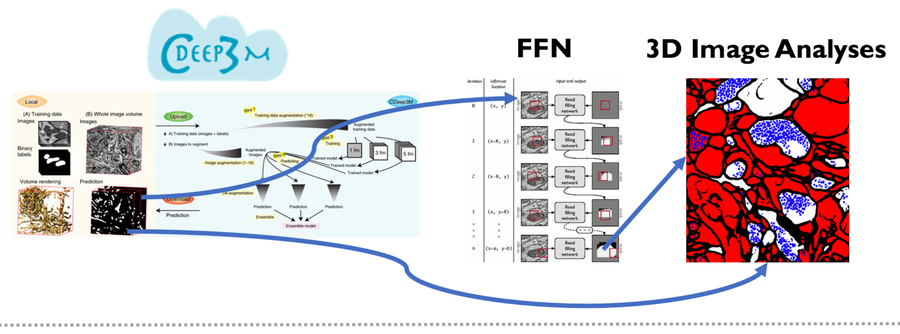
Acquire: The raw training data is acquired from SBEM microscopy data of the *Nucleus accumbens* in C57BL/6NHsd laboratory mice brain tissue. SBEM was performed using a Merlin SEM (Zeiss) with a Gatan 3View system.

Prepare: We use the IMOD software to prepare the *Nucleus accumbens* electron microscopy (EM) data for analysis. First, we convert raw DM4 electron microscopy files to the MRC file format. Next, we crop out a 1024x1024x100 voxel slice from the MRC stack for our validation and test datasets. Finally, we convert the MRC stack to 100 TIFF files, which CDeep3M uses as the input images to segment on.

Analyze: 3D image processing and flood-filling networks (FFN) are the two analytical approaches we implemented to improve the accuracy of synapse segmentations. 3D image processing is used to filter out presynaptic sites that do not meet a depth-wise synaptic vesicle threshold. Combined with applying erosion on the 3D presynaptic objects, this results in a more accurate 3D presynaptic site labeling. FFN is a 3D convolutional neural network that uses a watershed algorithm to produce a cell-body segmentation. We evaluated the accuracy for FFN using the average voxel-wise accuracy among all cell IDs, where a voxel is considered correct if the predicted segmentation and ground truth segmentation values match for that voxel. After analyzing the resulting visualizations and metrics, we refine our models by making incremental changes and determining if it results in better synapse segmentations.

Scale: Our main test for model scalability was doubling the number of images in our *Nucleus accumbens* data to become a 1024x1024x200 voxel image volume. On an AWS g4dn.xlarge EC2 instance with 16 GB of GPU RAM, it takes twice as long for FFN to segment a 200-image stack compared to a 100-image stack. Thus, FFN runs in linear time with respect to the number of voxels. We also ensured our solution CDense3M was scalable by developing automated workflows using AWS CloudFormation and Docker to enable users to deploy any number of CDense3M instances in minutes.

Report: We save the visualizations that our models produce as PNGs and HDF5 stack files so users can visually analyze the segmentations using tools like Fiji ImageJ. Training, validation, and test set metrics for FFN are also published to a Plotly Dash reporting dashboard to provides us real-time feedback on the performance of FFN.

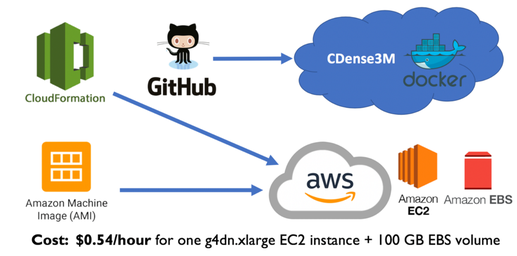


Solution Architecture

We use AWS CloudFormation and the NVIDIA Deep Learning Amazon Machine Image to create CDense3M-capable EC2 instances in less than two minutes. CDense3M requires at least 12 GB of GPU RAM, so we used the least expensive EC2 instance that meets this requirement, which is the g4dn.xlarge instance type with an on-demand price of $0.526/hour. AWS is scalable, both horizontally by creating more EC2 instances with CloudFormation and vertically by changing the EC2 instance type.

In our final solution architecture, CDense3M is deployed as a Docker container. Although we do support installing CDense3M directly on the operating system, we primarily deploy CDense3M inside a Docker container to maintain a consistent and reproducible environment. Docker containers isolate applications from the host machine, similar to a sandbox. Distributing CDense3M as a Docker container image also greatly simplifies the CDense3M installation process because we preinstall all of CDense3M's dependencies in the image via the Dockerfile. All the user needs to do is pull the latest CDense3M image and run the container to start using CDense3M.

CDense3M can also run in Google Colab, which provides up to 12 hours of free GPU access at a time. We created Colab Jupyter notebooks that provide a GUI and CLI for running CDense3M. Output images from CDense3M can be saved directly in Google Drive. However, due to usage limits, we only recommend using Colab for short tasks.



Cost: $0.54/hour for one g4dn.xlarge EC2 instance + 100 GB EBS volume

Key Insights

3D Image Analysis

- Best 3D labeling approach was to assign areas of images that overlap 80% or higher in area the same label and color
- 3D filtering algorithm was more accurate when counting the number of vesicles for each label of each image individually versus across all images
- Error rate after applying 3D vesicle filtering was still too high due to the connection of labels in some images, or the overlapping of the vesicles in one label, which caused the loss of some 3D presynaptic areas
- Applying erosion models on 3D presynaptic objects, along with setting two thresholds of five vesicles and 500 pixels for 3D vesicle filtering, results in the highest accuracy for 3D presynaptic site recognition

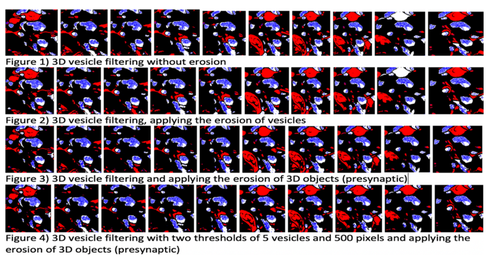


Figure 1) 3D vesicle filtering without erosion

Figure 2) 3D vesicle filtering, applying the erosion of vesicles

Figure 3) 3D vesicle filtering and applying the erosion of 3D objects (presynaptic)

Figure 4) 3D vesicle filtering with two thresholds of 5 vesicles and 500 pixels and applying the erosion of 3D objects (presynaptic)

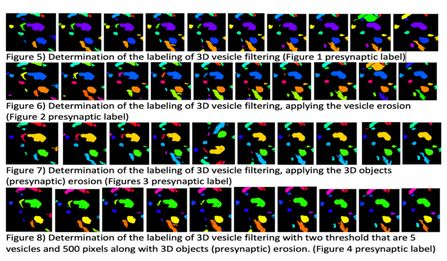


Figure 5) Determination of the labeling of 3D vesicle filtering (Figure 1 presynaptic label)

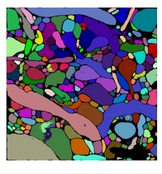
Figure 6) Determination of the labeling of 3D vesicle filtering, applying the vesicle erosion (Figure 2 presynaptic label)

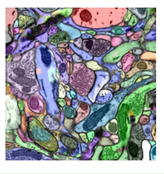
Figure 7) Determination of the labeling of 3D vesicle filtering, applying the 3D objects (presynaptic) erosion (Figures 3 presynaptic label)

Figure 8) Determination of the labeling of 3D vesicle filtering with two threshold that are 5 vesicles and 500 pixels along with 3D objects (presynaptic) erosion. (Figure 4 presynaptic label)

Flood-Filling Networks

- Had to retrain FFN membrane model an additional two million epochs, from six to eight million, to get accurate cell-body segmentations even on the training data
- FFN expects input images to have similar *xy* dimensions as training data
- FFN model trained on CDeep3M membrane segmentations achieves around a 10% increase in the voxel-wise classification accuracy for five segmentation metrics on two 1024x1024x100 voxel volumes of the *Nucleus accumbens* data, compared to an FFN model trained on FIB-25, the raw EM data of the *Drosophila* optic lobe
- FFN membrane model is more generalizable than FIB-25 model





Acknowledgements

We would like to thank our advisors Daniela Boassa and Matthias Haberl for all the support they have given us to help us complete this capstone project. We thank Professor İlkay Altıntaş for guiding us through the capstone design process. We thank all the staff at NCMIR that made this capstone project possible. We thank our families and employers for supporting us throughout this program. Finally, we would like to thank all the DSE MAS faculty, staff, and our classmates for helping us grow as data scientists and making these past two years a rewarding and enjoyable experience.