Semantic segmentation of high-resolution neuron meshes

Nicholas Turner
Princeton University
nturner@princeton.edu

Sven Dorkenwald
Princeton University
svenmd@princeton.edu

Abstract

We address the semantic labeling of neuronal shapes. Contemporary efforts in connectomics generate fragments of neuron shapes which have qualitatively recognizable forms (e.g. axon, dendrite, soma). We develop a machine learning model to predict these forms, and analyze the performance on this model to understand the constraints of its performance. We use PointNet, a deep learning model designed to handle unordered point clouds, and train it to use the mesh vertices of reconstructed neuronal fragments. Despite significant dataset bias, we achieve perfect segment classification performance on a held out test set for single-compartment meshes. We extend this approach to neuron meshes with multiple compartments, and plan future work to fully address this challenge.

1. Introduction

Connectomics is a subfield of neuroscience in which researchers extract neuron shapes and their connections (synapses) from very high-resolution ($\sim 5nm$) electron microscopy (EM) images, which are stacked to make up a 3D volume [7]. In this project, we explore the applicability of a recently published semantic segmentation approach to neuron reconstructions from these data. We see a high potential for this approach in the field of connectomics. Specifically, while the extraction of the connectivity between cells is usually of primary concern, additional information is often needed to draw biologically useful conclusions. For instance, the location of an individual synapse (an informational connection between two neurons) on a receiving cell can reveal insight about its function [4]. Hence, the segmentation of a cell into its biologically functional parts is of great interest.

Qi et al. [12] created PointNet, a neural network which directly operates on point clouds to classify and semantically segment 3D objects. This stands in stark contrast to prior approaches which either rely on dense volumes [10] or 2D views [16]. Compared to these other representations, using a mesh point cloud provides the benefits of a very sparse representation without losing crucial information about an object's shape. This compressed representation makes PointNet significantly faster and allows it to consider a larger context of an object when the object is very large [12].

Here, we apply variants of PointNet to meshes extracted from automatic reconstruction that were acquired by our lab (Seung Lab). Our results indicate that this approach works well for classifying subparts of an object at a time. In addition, we find that taking an extreme strategy to combat dataset bias yields robust performance across artificially difficult versions of the task. However, despite our continued success on segmenting subparts, more analysis needs to be done to investigate the applicability to cells with multiple components.

2. Background

Classically, cells in the neocortex can be partitioned into three parts: axon, dendrite and soma (cell body). Figure 5 shows an example cell from our dataset. In a simplified view, information from other cells is received on dendrites, the accumulated electrical signals travel to the soma, and an output signal propagates through the axon, exciting or inhibiting other cells.

Partitioning a cell into these three compartments is important as it provides information about the flow of signals within the network. Due to the small size of current EM datasets in comparison to the size of the brain, all cells are only partially contained in each dataset and most cells do not have their soma within its bounds. We refer to these cells with cut off processes as orphans.

The sheer number of cellular processes in contemporary connectomics datasets makes manual annotation of their types infeasible. In addition, neurons have extremely long-range connections across the brain [1], which can be much longer than the scale at which current connectomics datasets exist [11]. These observations suggest that we

require an automated method to label features of interest across both orphans and full cells. We see this work as first steps toward this eventual goal.

To a human eye, the three classical compartments are readily distinguishable (Figure 2 and Figure 5). Dendrites are often much thicker than axons, and many feature 'dendritic spines,' which are small protrustions from the main shaft. Axons tend to have simpler shapes, and instead feature bulbous compartments along the main process. Exceptions exist to these simple criteria. For example, inhibitory cells commonly feature dendrites with few or no spines. Nonetheless, these compartments remain qualitatively distinguishable in general.

Furthermore, neurons can be partitioned into cell types based on the shapes of their processes [2]. Distinguishing cells by their type is of such outstanding importance that it was mentioned as the first of seven goals of the NIH Brain Initiative for 2025 [9]. Besides genetics and function, morphology is one of the different ways to classify the type of a cell.

3. Related work

3.1. In connectomics

Biological analysis usually relies on manual classifications of subcelluar parts (e.g.[17, 15]). This has been feasible because studies were limited to a small number of neurons. However, increasing dataset sizes make an automation of this task necessary[3]. A recently published paper showed that much of this information can be classified automatically using Random Forest Classifiers (RFC)[3]. However, this work relied on additional information besides the morphology of the cell, required hand-designed features within a limited field of view, and was computationally expensive.

3.2. Semantic segmentation of point clouds

Prior to PointNet, semantic segmentation of point clouds relied on a transformation of the the point cloud into a feature representation known to be consumable by classifier such as a neural network. These approaches can be partitioned into hand-crafted feature generation and volumetric representations.

Hand-crafted features typically resemble certain statistics of the mesh that are of particular interest to the desired segmentation [14, 8, 3]. While intrinsically limited and not generalizable, this approach showed good performance for specific applications.

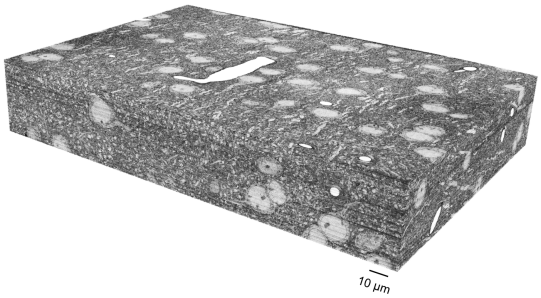

Figure 1. Rendering of the EM image volume. The brighter blobs are cell nuclei that reside in a cells' soma. The white elongated strip on the top is a cut through a blood vessel. The rest of the volume is densely packed with neural wiring (axons and dendrites).

Volumetric approaches either represent the mesh as a dense 3D volume or extract multi-views that can be classified by 2D CNNs. Using dense volumes and 3D CNNs does not sacrifice information but has limited scale [10]. For small enough volumes this approach reaches the best performance. View approaches represent the volume with 2D X-ray like images that are then fed into a multi-channel 2D CNN [16].

PointNet [12] introduced the possibility to use unordered meshes for segmentation and was followed by many similar approaches such as PointNet++ [13]. PointNet++ uses a hierarchical approach which emphasizes the local context of each mesh point.

We decided to use PointNet for this work, as it is the only approach that promises to scale to very large objects and does not require hand-designed features. While it has been shown that the particular task at hand could be solved by classifying small cutouts at a time [3] follow up tasks such as the classification of cell types require large large spatial contexts.

4. Experiments

We created three separate datasets in order to explore the semantic segmentation problem. Two of these datasets contain only orphans (Figure 2), while the other dataset has full cells with axon, dendrite and soma (Figure 5). Each dataset was generated from an automated segmentation of a $196\,\mu\text{m} \times 130\,\mu\text{m} \times 40\,\mu\text{m}$ volume from the primary visual cortex of a mouse (Figure 1). The automated segmentation was designed to be an over-segmentation, but mergers (segments that contain parts of multiple different neurons) still exist.

The first orphan dataset was intended to be small, yet pristine. Axons and dendrites join together at synapses, and

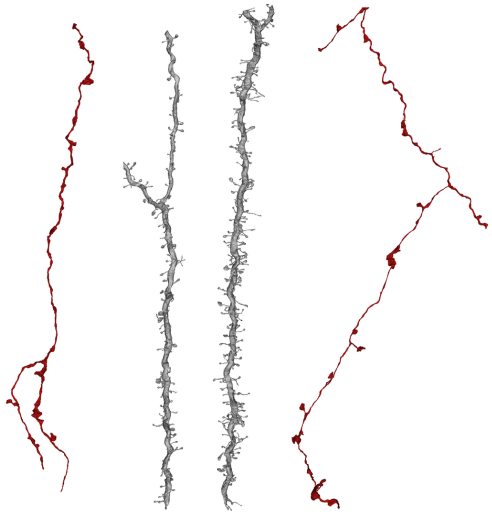

Figure 2. Examples for orphan dendrites (gray) and axons (red) (taken at different scales).

some prior work had been done within the reconstruction volume to locate synapses between many reconstructed segments. So, in order to get a balanced sample of both axons and dendrites, we selected synapses at random from these locations, and checked whether the associated segments had any mergers or obvious errors. After this check, segments at each synapse were added to the dataset if they possessed over 5000 mesh vertices. In total, we collected 182 axons and 200 dendrites. Approximately 50 of each class were randomly selected for both validation and test sets.

The large orphan dataset was generated from the same segmentation as the small one. However, we only applied certain heuristics to remove false examples and did not ensure 100% correctness. This trade-off allowed us to quickly generate 73383 examples for this dataset. It should be noted that only examples with enough mesh points can be used for a given training. For the common case of 5000 points, the dataset shrinks to 56360 examples. We estimate that less than 5% of these are falsely labeled or contain both classes (through mergers or full cells).

The last dataset contained only full cells (those with a cell soma). It was acquired by splitting each cell into its branches and labeling all branches individually ($n_{cells} = 109$). Since some cells lie at the dataset boundary not all of these samples contain an axon. In general, this dataset is highly-unbalanced in terms of mesh points (Axon: 3.4%, Dendrite: 83.1%, Soma:13.5%). Oftentimes, the exact boundary between classes is ambiguous. To circumvent this, we assigned some mesh points to the "no label" group (Figure 5) which are excluded when computing the loss. For training and testing we split the dataset in a 80 / 20 ratio.

Our goal is to identify the subparts of all meshes correctly. Therefore, we choose to formulate the problem of classifying axons and dendrites of the orphans as a semantic segmentation problem although it could be seen as a classification problem. This experiment is meant to be the first step of segmenting the full cells. We hope to be able to leverage the orphan datasets for the segmentation of the full cells.

5. Model architecture

The fundamental challenge in handling a point cloud representation is that point clouds are unordered [12]. Creating a model that learns shape features from point clouds thus requires generating a function which is able to learn based on the context of multiple points while being invariant to a permutation of the input points. PointNet circumvents this challenge by using a symmetric function (e.g. max pooling) to consolidate information across the entire point cloud in a bottleneck layer, and combining this global information with local features extracted from each point (Figure 3). The size of the bottleneck layer b is a tunable parameter.

Prior to the bottleneck layer (blue part in Figure 3), PointNet processes the input points with a Multi-Layer Perceptron (MLP) to extract local point features (n x 64). To align points from different point clouds to a canonical space, Qi et al. use a transformer net [6] that mimics a three dimensional transformation. They include a second transformer network to align the extracted point features to a global feature space with a feature alignment matrix A. This point feature representation is significantly larger than the coordinates handled by the spatial transformer net, making it difficult to optimize. Qi et al. solve this by adding a regularization term to the network loss which penalizes the L2 norm of AA^T to the identity matrix. This encourages A to be close to orthogonal.

Finally, the global features are appended to the local point features (n x 64) leading to a feature vector of size n x $(64 + b)$. The final segmentation is accomplished using another standard MLP with shared weights producing a prediction for each point.

We implemented PointNet in PyTorch along with a datahandler and trainer and added multiple data augmentations such as rotation and scaling. During training a random subsample of the pointcloud is used according to a predefined parameter (5000-20000 in our case). We train in batches of size 4-20 cells per iteration. Our loss is weighted

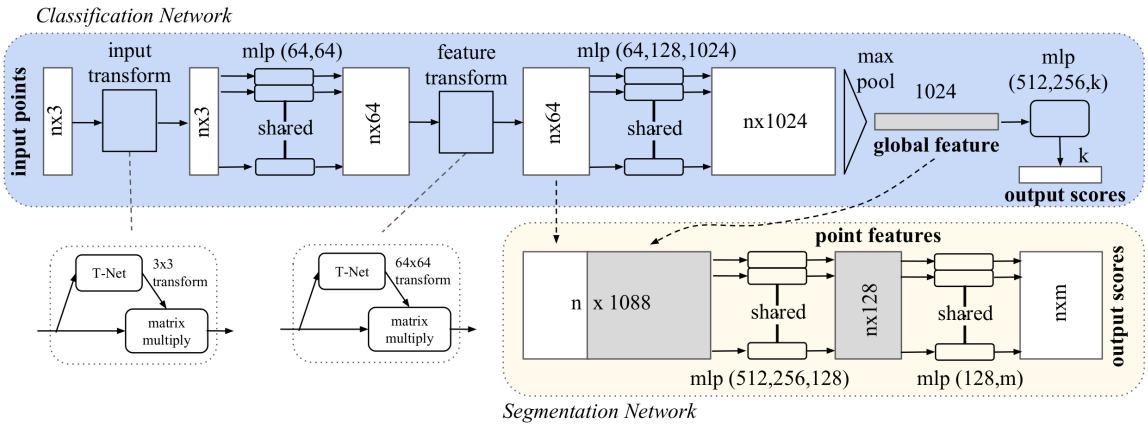

Figure 3. PointNet architecture. Figure taken from [12].

with the inverse frequency of each class to incentivize good performance for each class. The weights are either extracted globally before training or depend on the class frequency within the current batch (batchwise weights). As the larger orphan dataset is highly biased towards dendrites, we also experimented with only sampling batches from an equal number of axons and dendrites for training a network on orphan data (i.e. "even sampling"). In this case, we applied no weighting to the loss function. For the full cells, we also tried gradient masking, where we only considered an even number of points for each class from a larger prediction.

6. Analysis

In addition to accuracy in semantic segmentation or classification, there is significant value in having a model to cluster neuronal shapes into meaningful clusters. This could simply consist of the descriptions we've described (e.g. axon, dendrite), or it could also represent specific types of these processes as well. PointNet features a global feature vector for each mesh that passes through the network (Figure 3), and thus provides an embedding of the shape.

Here we perform a simple analysis of these representations to probe whether these global features represent meaningful patterns. Specifically, we perform K-Means clustering (with 100 restarts) and linear SVM classification on the global feature vectors for each orphan mesh in order to see whether we can recover the axon/dendrite distinction.

There are two prominent parameters we can tune to stress the performance of the PointNet architecture. The first is the bottleneck feature vector dimension, which constrains the complexity of the representation the network uses to describe the global shape of a point cloud. The second is the amount of point we allow the network to see at a time. We vary these attempting to draw some connections between the orphan and full cell segmentation tasks.

7. Challenges

Preliminary results featured significant overfitting to the training set. We took several steps in order to yield good performance. First, we noticed that the open source implementation of PointNet which we'd used for some time was incorrect. Notably, it was missing the second transformer network which operates on point features. We then reimplemented the model from scratch (as well as the PartNet extension) to ensure that all of the desired parts were accounted for.

From the outset of this project, we were also unsure about whether data quality might sabotage our generalization performance. This motivated the creation of separate datasets under different levels of size and cleanliness.

Lastly, we also explored several different dataset augmentation strategies. These strategies included random scaling, rotation, and shifts of the point cloud coordinates, as well as random cropping of contiguous pieces. The current results only include random scaling, where the coordinates are scaled by a factor randomly determined by a clipped normal distribution between 0.5 and 1.5.

The PointNet architecture applies batch normalization to several layers [5, 12]. Batch normalization standardizes each target feature map of the network by its mean and variance across the batch during training, and attempts to measure a global set of these statistics to apply after

training. In our case, this global set of statistics performed very poorly at test time (data not shown). This suggested that the network may base a considerable part of its inference on a segment's feature map values relative to the rest of a batch from the training set. However, abandoning the global statistics introduces some uncertainty into our inference estimates.

Fortunately, incorporating "even sampling" (creating batches from an equal number of axons and dendrites), allowed us to train networks without batch normalization.

We observe a large performance difference between orphan processes and full cells. The magnitude of this difference was unexpected, and lead us to hypothesize that our model wasn't powerful enough to flexibly handle multiple classes within the same shape. Full cells often have many more vertices than orphan processes (10x more on average), and it might be more difficult to consolidate the shape of full cells into a global representation. However, this means that we could explore whether the network wasn't powerful enough by training less powerful models on the orphan processes - in a sense simulating the full cell task by an ablation study. Specifically, we trained models with either (1) a smaller bottleneck feature vector dimension or (2) fewer points shown to the network, varying these variables systemically to see where performance would break down.

8. Results

8.1. Semantic classification and segmentation

The basic PointNet model with no modification achieves perfect accuracy in terms of semantic classification trained on either dataset (Table 1). This encourages further study of other tasks.

Dataset / Accuracy %	Axon	Dend
Small	100	100
Large	100	100

Table 1. Segment semantic classification performance over 20 iterations. Row delineates the dataset used for training.

If we instead inspect performance on a single node basis, we can analyze a small step towards the more general semantic segmentation task on full cells. In addition, these results show more interesting patterns. Specifically, we find that the small and pristine dataset performs worse than an imperfect and much larger training set using the PointNet model with no modification (Table 2). The large training set performance is near perfect, though performance across both datasets for dendrites is slightly worse. Notably, this difference also arises in the small dataset, which has more dendrites than axons. Thus, we expect this difference to arise from dendrites having more complicated shapes, which are more difficult to summarize with a set number of mesh nodes.

However, using even sampling during training and removing batch normalization reached perfect performance. This finding also negates the dataset effect. Specifically, training a model on either dataset yields perfect node-wide classification performance.

Training Procedure / Accuracy %	Axon	Dend	Overall
Orphans (small set)	88.7	86.6	87.6
Orphans (large set)	100	94.3	97.1
Orphans (even)	100	100	100
Full cells (batch weights)	63.5	53.5	10.1
Full cells (gradient masking)	25.2	74.0	17.5

Table 2. Semantic mesh vertex classification performance on the respective test sets over 20 iterations. Row delineates the dataset used for training. All orphan networks are tested on a held out orphan test set. Full cell networks were trained on full cells alone, and tested on a held out set of full cells. even: either training set, even sampling, no batch normalization.

8.2. Global feature vector analysis

Visualizing the global feature vectors for the even sampling model trained on the large training set shows clear separation between axon and dendrite meshes (Figure 4). There is significantly more variation across the representations of dendrites than axons. This spread disrupts a naive clustering of the feature vectors. Using a simple K-Means procedure yields an Adjusted Rand Score of 0.78, despite the clear separation between the two classes. Nonetheless, we're still able to recover the class separation perfectly using a linear SVM. This suggests that the feature vectors are linearly separable by a hyperplane, despite being difficult to cluster by a simple clustering.

8.3. Full cells

We applied PoinNet to the full cell dataset with different weighting approaches and data augmentations (scaling and rotation) to address the over fitting problems that we encountered prior to the milestone report. Using batchwise gradient weights or gradient masking prevented overfitting; the performance on the training set was comparable to the performance on the test set. As a result, the prediction of PointNet is less biased towards the dendrite class (Table 2).

When examining the prediction we found that PointNet is able to learn coarse spatial features but appears to lack the ability to extract subcomponents of a cell. For instance,

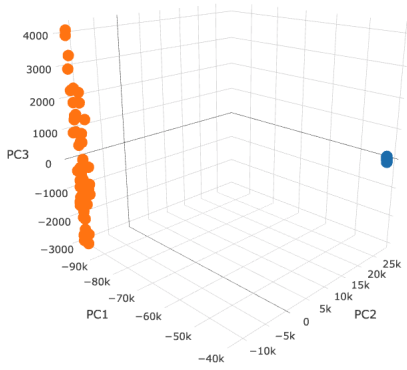

Figure 4. PCA Projection of each bottleneck feature vector of the "Large" dataset network with even sampling and no batch normalization. orange:dendrite, blue:axon

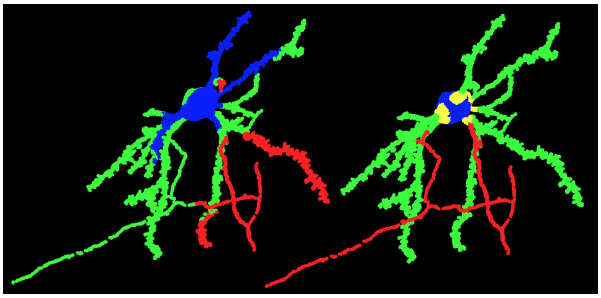

Figure 5. Prediction using batchwise gradient weights (left) and ground truth (right) of a cell from the test set. red: axon, green: dendrite, blue: soma, yellow: no label.

in the example (Figure 5) PointNet correctly identified the axon but also included points belonging to other branches.

We tested different parameters for the bottleneck dimension and the number of points within reasonable ranges but where not able to improve the performance further. We settled with 150000 points and a bottleneck dimension of size 1500.

We tried to use a PointNet trained on orphan cells to classify axon and dendrite parts of the full cells. Interestingly, this net failed by classifying everything as belonging to a single class. This might be due to the classification like overfitting that we encountered on the orphan cells.

8.4. Ablation Studies

The ablation studies revealed a surprising resilience of the orphan segmentation performance to both sparser mesh samples and global feature vector representations. The global feature vector dimension in fact had no effect on the performance on the test set (data not shown). Even training a network with feature vector dimension of 1

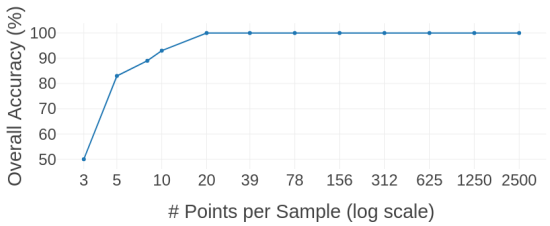

Figure 6. Node-wise semantic segmentation accuracy as a function of number of points used to represent an orphan mesh. Performance on the test set is perfect for any model using more than 20 points.

yielded perfect performance.

Decreasing the number of points used to represent a mesh held stable performance all the way to 20 points (from the original 5000; Figure 6). Performance below this point still remained above chance performance, which suggests that very simple relationships between a small set of points still holds considerable information about the task.

9. Discussion

We've implemented a framework to semantically classify mesh points of neurons using PointNet. We showed that we are able to predict meshes from orphan segments with very good performance. We also show that, in some regimes, an imperfect yet data hungry strategy performs best.

The robust success of even sampling suggests that dataset bias is a core difficulty in addressing this problem. Even sampling can be seen as an extreme form of balancing a dataset, which seems different to approaches which weight the gradient signals by class instead of by sample. We've also extended this balancing to full cells, by only allowing the gradient to see the same number of examples across each class at a time.

The ablation studies showed a high resilience of the orphan segmentation performance, which sheds some light onto it's apparent success. First, dendrites and axons differ even in extremely sparse samples of their shapes (between 0.001 and 0.4%). This suggests that the network can at least partially makes use of simple relationships between a small subset of points. This makes some intuitive sense, as the presence of dendritic spines would likely offset an otherwise smooth curve of points.

In addition, we observe that the bottleneck feature vector dimension has no effect on the orphan segmentation

performance. This observation, paired with the separability of the axon and dendrite representations, strongly suggests that the networks trained on orphan data finds a way to effectively perform classification of the orphan meshes by that point. Upon doing this, the network likely finds a simple strategy to label each vertex with the label corresponding to the identified class.

While we were able to balance PointNets performance on full cells through different gradient weighting schemes, the performance is still poor. This difficulty is non-trivial since, by the results of the ablation study, the full cell network should have enough points to classify each dendrite or axon individually (requiring roughly $10 \times 20 = 200$ vertices). Instead, we suggest that PointNet lacks the ability to identify contiguous subcomponents of a cell. Therefore, we might add a regularization term to the loss function that informs PointNet about components such as branches where all points have the same label.

A few possible avenues remain to be explored as future work. While we suspected that expanding our approach to PointNet++ might have taken more time than the project would allow, a hierarchical approach may work well to extend the single class performance of our model to multiple classes. In addition, we should be able to gather a larger dataset in the coming months.

References

[1] R. D. Allen, J. Metuzals, I. Tasaki, S. Brady, and S. Gilbert. Fast axonal transport in squid giant axon. *Science*, 218(4577):1127–1129, 1982. 1

[2] J. A. Bae, S. Mu, J. S. Kim, N. L. Turner, I. Tartavull, N. Kemnitz, C. S. Jordan, A. D. Norton, W. M. Silversmith, R. Prentki, et al. Digital museum of retinal ganglion cells with dense anatomy and physiology. *bioRxiv*, page 182758, 2018. 2

[3] S. Dorkenwald, P. J. Schubert, M. F. Killinger, G. Urban, S. Mikula, F. Svara, and J. Kornfeld. Automated synaptic connectivity inference for volume electron microscopy. *Nat. Methods*, Feb. 2017. 2

[4] K. M. Harris. Structure, development, and plasticity of dendritic spines. *Current Opinion in Neurobiology*, 9:343, 1999. 1

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4

[6] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015. 3

[7] V. Jain, H. S. Seung, and S. C. Turaga. Machines that learn to segment images: a crucial technology for connectomics. *Current opinion in neurobiology*, 20(5):653–666, 2010. 1

[8] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433 – 449, May 1999. 2

[9] L. A. Jorgenson, W. T. Newsome, D. J. Anderson, C. I. Bargmann, E. N. Brown, K. Deisseroth, J. P. Donoghue, K. L. Hudson, G. S. F. Ling, P. R. Macleish, and et al. The brain initiative: developing technology to catalyse neuroscience discovery. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1668):2014016420140164, 2015. 2

[10] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2015. 1, 2

[11] S. Mikula. Progress towards mammalian whole-brain cellular connectomics. *Frontiers in neuroanatomy*, 10:62, 2016. 1

[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 4

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5105–5114, 2017. 2

[14] R. B. Rusu, N. Blodow, and M. J. Beetz. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009. 2

[15] H. Schmidt, A. Gour, J. Straehle, K. M. Boergens, M. Brecht, and M. Helmstaedter. Axonal synapse sorting in medial entorhinal cortex. *Nature*, 549(7673):469–475, sep 2017. 2

[16] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015. 1, 2

[17] S.-Y. Takemura, A. Bharioke, Z. Lu, A. Nern, S. Vitaladevuni, P. K. Rivlin, W. T. Katz, D. J. Olbris, S. M. Plaza, P. Winston, T. Zhao, J. A. Horne, R. D. Fetter, S. Takemura, K. Blazek, L.-A. Chang, O. Ogundeyi, M. A. Saunders, V. Shapiro, C. Sigmund, G. M. Rubin, L. K. Scheffer, I. A. Meinertzhagen, and D. B. Chklovskii. A visual motion detection circuit suggested by Drosophila connectomics. *Nature*, 500(7461):175–181, Aug. 2013. 2